Институт математики им. С. Л. Соболева СО РАН
Новосибирский национальный исследовательский государственный университет
Математический центр в Академгородке
Российская Ассоциация Искусственного Интеллекта
Российская Инженерная Академия
Institute of Electrical and Electronics Engineers (IEEE)

ЗНАНИЯ - ОНТОЛОГИИ - ТЕОРИИ

Материалы IX Международной конференции
“Знания - Онтологии - Теории” (ЗОНТ-2023)
2 - 6 октября 2023 г., Новосибирск

НОВОСИБИРСК 2023

Программный комитет конференции:
Председатель: академик РИА Пальчунов Д.Е.

Члены Программного комитета:
Абламейко С.В., академик НАН Беларуси (Минск, Беларусь)
Бартуссек В., проф. (Зульцбах, Германия)
Бензе Х., проф. (Дортмунд, Германия)
Болдырев И.А., академик РИА (Новосибирск)
Борисова И.А., доцент (Новосибирск)
Васенин В.А., проф. (Москва)
Васильев С.Н., академик РАН (Москва)
Витяев Е.Е., проф. (Новосибирск)
Вольф К.Э., проф. (Дармштадт, Германия)
Гаврилова Т.А., проф. (С.-Петербург)
Гантер Б., проф. (Дрезден, Германия)
Гончаров С.С., академик РАН (Новосибирск)
Грибова В.В., член-корреспондент РАН (Владивосток)
Гумиров В.Ш. (Новосибирск)
Загорулько Ю.А., доцент (Новосибирск)
Ковалерчук Б., проф. (Сиэтл, США)
Колчанов Н.А., академик РАН (Новосибирск)
Кузнецов С.О., проф. (Москва)
Лукашевич Н.В., проф. (Москва)
Манцивода А.В., проф. (Иркутск)
Марков К., проф. (София, Болгария)
Марчук А.Г., проф. (Новосибирск)
Массель Л.В., проф. (Иркутск)
Окунев А.Г., доцент (Новосибирск)
Пассоу А., проф. (Франкфурт на Майне, Германия)
Рейбольд А.В., проф. (Дармштадт, Германия)
Свириденко Д.И., проф. (Новосибирск)
Смирнов С.В., проф. (Самара)
Трофимов В.К., проф. (Новосибирск)
Хазанкин Г.Р. (Новосибирск)
Херре Г.П., проф. (Лейпциг, Германия)
Целищев В.В., проф. (Новосибирск)
Цельгер Й., проф. (Инсбрук, Австрия)
Черемисина Е.Н., проф. (Дубна)
Шафермайер Р., проф. (Берлин, Германия)
Яхъяева Г.Э., доцент (Новосибирск)

Оргкомитет конференции:
Председатель: Пальчунов Д.Е.
Ученый секретарь: Найданов Ч.А.

Члены Оргкомитета:
Борисова И.А., Зыкова А.А., Мирошниченко Л.А., Хазанкин Г.Р., Яхьяева Г.Э.

В сборнике содержатся тексты докладов IX Международной конференции “**Знания - Онтологии - Теории**” (ЗОНТ-2023), г. Новосибирске, 2-6 октября 2023 года.

Тематика данной Конференции отражает основные стадии процессов познания:

Обнаружение закономерностей и извлечение знаний, скрытых в структурированных и неструктурированных данных.

Систематизация знаний.

Построение теорий предметных областей.

Материалы сборника могут быть полезны специалистам, работающим в области теории познания, искусственного интеллекта, распознавания образов, интеллектуального анализа данных, анализа текстов на естественном языке.

ISSN 0568-661X

© Институт математики им. С. Л. Соболева СО РАН, Новосибирский государственный университет, 2023

Международная конференция “Знания - Онтологии - Теории” (ЗОНТ-2023) продолжает обсуждение проблем, которые рассматривались на Всесоюзных симпозиумах “Методы обнаружения закономерностей” (МОЗ) в период с 1976 по 2002 годы и на Всероссийской конференции с международным участием “Знания - Онтологии - Теории” (ЗОНТ-2007, ЗОНТ-2009, ЗОНТ-2011, ЗОНТ-2013, ЗОНТ-2015, ЗОНТ-2017) с 2007 по 2017 годы, а также на Международной конференции “Знания - Онтологии - Теории” (ЗОНТ-2019, ЗОНТ-2021) в 2019 и 2021 годах. Проблема МОЗ является основной проблемой в тех областях науки, которые связаны с изучением процессов познания. Тематика данной Конференции отражает основные стадии этих процессов:

Обнаружение закономерностей и извлечение знаний, скрытых в структурированных и неструктурированных данных. Машинное обучение. Распознавание образов. Прогнозирование. Индуктивный вывод.

Систематизация знаний. Инженерия знаний. Управление знаниями. Извлечение знаний из текстов на естественном языке. Разработка онтологий предметных областей, технологии создания и применения онтологий.

Построение теорий предметных областей. Разработка семантических и онтологических моделей предметных областей. Анализ формальных понятий. Логическая семантика естественного языка. Нечёткие логики.

Желаем Вам успешного участия в Конференции.

Председатель:
Академик РИА Пальчунов Д.Е.

The International conference “Knowledge - Ontology - Theories” (KONT-2023) continues discussion of problems which where considered on All-Union symposium “Methods of discovering of regularities” (MDR) through 1976 - 2002 years, on the All-Russia conference with the international participation “Knowledge - Ontology - Theories” (KONT-2007, KONT-2009, KONT-2011, KONT-2013, KONT-2015, KONT-2017) through 2007 - 2017 years and on the International conference “Knowledge - Ontology - Theories” (KONT-2019, KONT-2021). MDR problem is the basic problem in those areas of a science, which are connected with studying of knowledge growth processes. Conference topics reflect the main stages of knowledge discovery process:

Discovery of patterns and extraction of knowledge contained in structured and unstructured data. Machine learning. Image recognition. Forecasting. Inductive reasoning.

Systematization of knowledge. Knowledge engineering. Knowledge management. Knowledge extraction from natural language texts. Development of subject domain ontologies, technologies of ontology development and application.

Generation of subject domain theories. Development of semantic and ontological models of subject domain. Formal concept analysis. Logical semantics of natural language. Fuzzy logics.

We wish you a successful participation in the Conference.

Chair:
Full-member of RAE Palchunov D. E.

Содержание

Аникин М.А., Королева М.Н.
Формализация модели предприятия с использованием семантических технологий 10
Артемьева И.Л., Чусова А.Е.
Разработка онтологии предметной области архитектурной акустики 19
Ахмедов Э.Ю.
Исследование подходов к анализу тональности в текстах узбекского языка 24
Балчиков В.И., Ланин В.В.
Разработка программной платформы взаимодействия устройств интернета вещей на основе онтологического инжиниринга 31
Барахнин В.Б., Менглиев Д.Б., Мадиримов Ш.
Разработка алгоритма для замены в текстах на узбекском языке диалектных слов на литературные 39
Бериков В.Б., Кутненко О.А.
Метод решения задачи слабо-контролируемой групповой классификации 45
Вопилова Е.В.
Характеристические функции вычисления значимости терминов в семантической модели представления научных знаний 49
Вопилова Е.В., Крючкова Е.Н.
Динамика изложения информации как ассоциативный идентификатор научного текста 54
Галиева А.Г., Пальчунов Д.Е.
Методы автоматизации разработки смарт-контрактов при помощи моделей ситуаций 60
Гатиатуллин А.Р., Прокопьев Н.А., Сулейманов Д.Ш.
Лингвистические графы знаний для представления моделей тюркских языков 66
Глушкова В.Н., Коровина К.С.
Полиномиально вычислимые регулярные модели систем реального времени 73
Гринберг Э.Я., Кубельский М.В.
Опыт выбора инструмента для визуальной разработки онтологии 83
Губанов Д.А, Кузнецов О.П., Суховеров В.С., Чхартишвили А.Г.
О построении профилей в тематическом пространстве теории управления 89
Давыдов А.В., Ларионов А.А., Нагул Н.В.
Применение исчисления позитивно-образованных формул в задачах управления дискретно-событийными системами 95
Демидов Д.В., Мардер Л.М.
Метод автоматического построения тезауруса технического документа 105
Долгов Н.Ю.
Разработка виртуальной машины для исполнения L*-программ P-полного логического языка в децентрализованных средах 114
Доманов О.А.
Продолжения в семантике естественного языка 120
Зайцев А.О., Пальчунов Д.Е., Трегубов А.С.
Автоматизированные методы проведения собеседований 129

Иванус А.И.
Истинность знаний, эмоции, искусственный интеллект 137
Ильин В.П.
Концепция и архитектура базы знаний систем линейных алгебраических уравнений 145
Кладов Д.Е., Климонтов В.В., Бериков В.Б.
Алгоритм кластеризации временных рядов и его применение для анализа гликемических кривых 154
Козицын А.С., Шачнев Д.А.
К созданию системы определения авторитетности конференций на основе наукометрических данных 162
Колганова А.В., Апанович З.В.
Метод визуализации, упрощающий понимание данных и результатов выравнивания сущностей 169
Кононенко И.С., Загорулько Ю.А., Серый А.С., Сидорова Е.А., Шестаков В.К.
Классификация типовых моделей рассуждения и их применение для исследования аргументации в текстах научной коммуникации 178
Копылова О.А., Хайретдинов М.С.
Анализ и распознавание транспортных шумов с использованием нейросетевого подхода 188
Левичев А.В., Нещадим М.В., Пальянов А.Ю.
О применении переноса Ферми-Уолкера в звёздных наблюдениях: общий подход и солнечная конкретика 195
Лобовиков В.О.
Знания, онтологии и мультимодальные аксиоматические теории (дедуктивное доказательство непротиворечивости формальной теории Ф0, синтезирующей формальную аксиологию, универсальную эпистемологию и философскую онтологию) 202
Мирошниченко Л.А., Бахмутова И.В., Гусев В.Д.
Сравнение традиционных и формальных структурных единиц знаменного распева 212
Павлова А.И., Осипов А.Л.
Применение методов машинного обучения и ГИС для автоматической агроэкологической типизации земель 222
Пахомов И.В.
Метод распознавания эмоциональной окраски текстов, основанный на сочетаниях слов 229
Привезенцев А.И.
Программная архитектура типовых узлов научных данных и вычислений по атмосферным наукам, основанная на их онтологических моделях 235
Резников В.М.
Философские концепции в приложениях анализа данных и искусственного интеллекта 246
Суров И.А.
Процессная онтология и квантование информации 255

Филимонов В.А.
Когнитивные возможности и модели коллективных субъектов 266
Хорошавин А.К.
Уменьшение влияния проблемы холодного старта при формировании туристических рекомендаций 272
Чернявцева С.И.
Разработка интеллектуального помощника для автоматизированного порождения документов кафедры 279
Черпаков К.М.
Разработка блокчейн платформы для осуществления сделок с динамическим дисконтированием 285
Чудаков Д.С., Бериков В.Б.
Прунинг свёрточных нейронных сетей через проецирование свёрток 295
Шелехов В.И., Каблуков И.В., Янбулатов Д.Р.
Разработка и верификация программ обработки графов в предикатном программировании 299
Шишкин А.А., Трегубов А.С.
Применение генеративно-состязательных нейронных сетей для порождения предложений, связанных отношением логической выводимости 307
Щеглов Б.О., Рева Г.В., Щеглова С.Н.
Прототипирование базы знаний для определения цитологических изменений эндоцервикса в гинекологии 313
Khakimova S.S.
Increasing the efficiency of teaching programming through the use of innovative technologies in programming paradigms in higher education 321
Kolonin A.G.
Self-tuning hyper-parameters for unsupervised cross-lingual tokenization 329
Levichev A.V., Palyanov A.Yu.
A brief recollection of the Multi-Level Model, and some of its properties which distinguish it from the Standard Model 337
Masharipov S.A., Rakhimova L.S., Akhmedova X.I.
Evaluating student knowledge levels through fuzzy logic in the "Smart School"information system 348
Matyokubov U.K., Davronbekov D.A., Matqurbanov T.A.
Development of Efficient PON Architecture Models for Small Areas 356
Sadikov M.A., Sobirov B.I.
Application of neural networks in the field of information security 365
Архипова В.И.
Методы извлечения знаний о ситуациях и событиях из текста естественного языка 369
Балашова С.А.
Извлечение эмоциональных оценок из текстов естественного языка 371

Вартазарян Э.А.
Реализация графического инструмента трассировки системных вызовов на основе gVisor 375
Гуртуева О.А.
Разработка методов определения эмоций пользователя на основе анализа текстовых и звуковых данных 376
Демчук М.Е.
Применение опроса для рекомендации косметической продукции с учетом индивидуальных черт пользователя 378
Дробышевский И.А.
Разработка методов создания модели пользователя интеллектуального помощника 379
Дымонт Н.А.
Проверка корректности экспертных оценок, заданных интервалами 382
Дьяченко М.А.
Возможности разработки и применения гибридных подходов для создания децентрализованных метавселенных, доступных в реальном времени 383
Зыкова А.А.
Комплексная интерпретируемость искусственного интеллекта 384
Ковалевская О.А.
Разработка сервиса для автоматизированной рассылки электронных писем 390
Копытков Д.А.
Рассуждения на основе прецедентов: обзор методов и перспектив их применения 393
Курдюков Д.А.
Обучение модели для прогнозирования продовольственных запасов в Российской Федерации 395
Малаева Е.Д.
Анализ согласованности логических аспектов экспертных оценок 396
Мацько А.М.
Разработка системы динамической инструментации системных вызовов на основе gVisor 397
Миронов В.С.
Извлечение эмоций из текстов естественного языка при помощи LogicText 398
Осипов А.А., Болотов К.Ю.
Разработка веб-приложения для управления регистрационными данными домашних животных 401
Панова В.Е.
Анализ качества размещения пользователя в кадре с помощью программной системы 402

Петров С.Е.
Настройка среды для тестирования и развёртывания смарт-контрактов и децентрализованного приложения на основе блокчейна NEO 403
Попова П.В.
Поиск и обработка данных для модели прогнозирования продовольственных запасов Российской Федерации 404
Радеев Н.А.
Анализ эвристических правил объединения признаков в табличных данных 406
Садриев В.Д., Иванов А.И., Пальчунова О.Д., Пашкова А.С.
Мобильное приложение рецептов с голосовым помощником Talky Chef 407
Смоляков П.Е.
Выборочное подключение процессов к VPN посредством перехвата системных вызовов в ОС Linux 413
Сухих А.О.
Аналитик - фасилитатор командной работы 414
Тимофеев А.Т.
Оптимизация рациона питания на основе индивидуальных параметров человека 416
Тихомолов Я.А., Михалёв М.А.
Разработка интеллектуального голосового помощника для заведений общественного питания на базе SberPortal 417
Худяков Д.А.
Выявление аномалий в работе программных систем на основе распределенной трассировки логов 418
Чеболтасова С.К.
Разработка цифрового ассистента для подбора финансовых продуктов 419
Черновская Я.Т.
Разработка архитектуры децентрализованного приложения на основе блокчейна NEO 423
Чурилова Р.С.
Построение онтологии продукта для реализации поисковой системы 424
Шатрова А.И.
Разработка алгоритма замены упражнений в йога-тренировках 429
Шелдяков И.П.
Использование психографики для создания рекомендательной системы компьютерных игр 431
Щербин А.С.
Исследование влияния методов прунинга нейронных сетей на их интерпретируемость 432
Якобсон А.С.
Автоматизированные методы ведения аргументативного диалога с пользователем 433
Янин А.С.
Компиляция объектно-ориентированного языка в WebAssembly 435
Lyutikova V.S., Litovchenko I.N.
Modern seismic visualization tools 439

Формализация модели предприятия с использованием семантических технологий

Аникин М.А.[1], Королева М.Н.[1]

[1]*МГТУ им. Н.Э. Баумана, 2-я Бауманская ул., д. 5, корп. 1, г. Москва, 105005, Россия.*

maxim.anikin1@yandex.ru, maria.svyatkina@gmail.com

Аннотация. *В работе анализируются перспективы использования семантических технологий на предприятиях. Основная идея заключается в использовании существующих инструментов создания единого информационного пространства, в котором возможно хранение и изменение данных организации и отдельных ее частей вплоть до отдельных процессов. Оценка проводится на основе учета фактора уменьшения расходов на сопровождение и создание автоматизированных систем, а также повторного использования частей системы на разных предприятиях. Для формирования единой базы знаний используется технология OSTIS и SC-code, а именно две формы его представления – SCg и SCn. Отмечается гибкость технологии в отношении записи знаний предприятия, а также простота интерпретации используемых форм представления.*

Ключевые слова: семантические сети, онтология, OSTIS, база знаний, графы, интеллектуальная система

1 Введение

В настоящее время подавляющее большинство информации представлено в потоковом виде (текст, изображение, видео и аудио). Данный тип представления является естественным для человека, но требует ручной обработки для интерпретации. Существуют направления, позволяющие анализировать потоковую информацию методами машинного обучения [1-3], но их проблема в том, что они основываются на системах «черного ящика», что накладывает ограничения на оценку правдоподобности получаемых результатов. Также недостатком этих методов является решение и обучение моделей под конкретную узкую задачу.

Параллельно идет развитие и внедрение семантических технологий [4, 5] в различных областях деятельности. Они позволяют представлять и использовать информацию в электронном виде. При этом системы, использующие данные технологии, могут быть однозначно интерпретированы, так как структурирование информации и методы ее использования задаются человеком посредством применения определенных форматов, стандартов и договоренностей.

Использование семантических технологий позволяет не только представлять информацию в электронном виде, но производить над ней автоматические операции, в том числе и логические выводы, если это необходимо. Это открывает возможности для автоматизации процессов в производственной сфере, а также в области информационных технологий.

Актуальной проблемой использования уже существующих решений различных задач является вопрос их взаимодействия и соответственно интеграции в единую систему с учетом дальнейшей ее модернизации. Даже при разработке интерфейсов для взаимодействия между разработанными системами возникает проблема выделения большого количества человеческих и материальных ресурсов для ее поддержания и изменения в течение всего жизненного цикла. Поэтому необходимы технологии, позволяющие реализовывать систему, используя подход модульности, где каждый отдельный модуль является достаточно автономным для независимого, т.е. не влияющего на другие части системы, изменения. Также система должна располагать возможностями для записи, хранения, чтения и изменения данных о ее частях в рамках единой

базы знаний. Эти факторы позволят существенно уменьшить расходы на ее создание и сопровождение.

Для данной работы ставится задача исследования перспектив использования семантических технологий на предприятиях с дальнейшим анализом полученных результатов.

2 Постановка задачи

Предлагается использование технологии OSTIS. Согласно ей [6], предприятие представляется в виде многоагентной системы, внутри которой:

- вся информация объединена в единое информационное пространство;
- все участники процесса (люди, роботизированные системы, различные виды промышленных комплексов и т.д.) рассматриваются в качестве агентов в этой общей базе знаний (БЗ). Они в свою очередь отслеживают интересующие их ситуации в БЗ и реагируют на них соответствующим образом, а затем описывают результаты своей деятельности в БЗ таким образом, чтобы они были доступны другим агентам для анализа;
- БЗ системы иерархически организована, то есть представляет собой иерархию предметных областей и соответствующих им онтологий;
- многоагентная система также организована иерархически, то есть агенты могут формировать иерархии произвольной глубины.

Данный подход имеет ряд преимуществ:

- отсутствие необходимости разработки инструментов для прямого взаимодействия компонентов системы ввиду их взаимодействия через общую память;
- в связи с тем, что все агенты взаимодействуют через общую память, для системы не имеет значения, каким образом устроен физически тот или иной агент. В свою очередь это открывает возможности для модернизации, изменения, добавления и удаления агентов без модификации системы;
- благодаря использованию ассоциативного поиска по БЗ любой участник может иметь доступ ко всей интересующей его информации, а не только к какому-либо заранее заготовленному фрагменту;
- информация, хранящаяся в БЗ, может по-разному отображаться для конкретных категорий пользователей. При этом сама информация остается неизменной, меняется только ее визуализация, что приводит к отсутствию ее дублирования.

Для формирования единой БЗ необходим универсальный язык, позволяющий представлять любые знания. Данному критерию соответствует SC-код. Он сочетает в себе как теорию графов, так и теорию множеств, что позволяет записывать знания любого вида на одном языке. Иными словами, этот код владеет свойствами универсальности и унифицированности представления знаний и информации, что в свою очередь приводит к удобству их хранения и обработки.

Есть несколько форм представления SC-кода:

• SCs – текстовая форма представления;

• SCn – гипертекстовая форма представления;

• SCg – графическая форма представления.

В данной работе основное внимание уделяется SCg и SCn формам представления. Далее представлены основные элементы данных форм представления, используемых в работе.

2.1 Элементы в форме преставления SCg

Константный sc-узел общего вида (рис. 1). Представляет собой либо множество, либо терминальную сущность, то есть сущность, которая не является множеством.

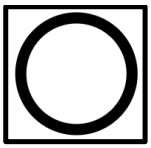


Рис.1. Константный sc-узел общего вида.

Базовая sc-дуга (рис. 2). Данный элемент обозначает принадлежность элемента множеству.

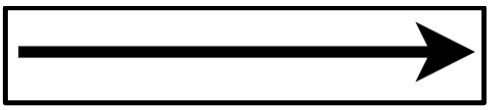


Рис.2. Базовая sc-дуга.

Ролевое (атрибутивное) отношение (рис. 3). Данный элемент обозначает подмножество отношения принадлежности (элементами являются базовые sc-дуги). То есть он указывает роль сущности во множестве. Идентификатор элемента должен иметь в конце символ «'».

Рис.3. Ролевое отношение.

Неролевое отношение (рис. 4). В основном данный элемент используется для уточнения типа связи, задаваемой при помощи sc-души общего вида. Идентификатор элемента должен иметь в конце символ «*».

Рис.4. Неролевое отношение.

Класс (рис. 5). Этот элемент представляет собой множество сущностей, которые однотипны по какому-либо признаку.

Рис.5. Класс.

sc-дуга общего вида (рис. 6). Обозначает какую-либо бинарную связь между элементами множества. Семантика связи может быть уточнена элементами отношений.

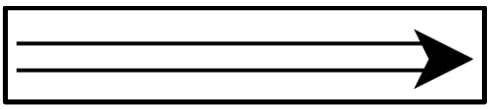


Рис.6. sc-дуга общего вида.

2.2 Элементы в форме преставления SCn

SCn-код является гипертекстовым вариантом визуализации текстов SC-кода, приближенный к форматированному естественному языку. В его рамках используются идентификаторы sc-элементов, а также специальные обозначения sc-коннекторов (табл. 1).

Таблица 1. sc-коннекторы в форме представления SCn.

sc-коннектор	*Описание*
=> <=	Изображение константной sc-дуги общего вида, обозначающей направленную бинарную связь между двумя сущностями.
∈ ∋	Изображение sc-дуги основного вида, обозначающей принадлежность элемента множеству.

3 Формализация модели предприятия

При формировании упрощенной модели рецептурного колбасного производства опираемся на стандарт ISA-88 [7], в котором определяются иерархии для модели предприятия.

3.1 Физическая модель предприятия

Первый уровень модели – уровень предприятия. Рассматривается предприятие с наименованием ОАО «КолбасЗавод». Данный факт формализован на языке SCg (рис. 7) и языке SCn (листинг 1).

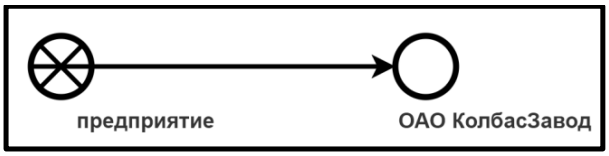


Рис.7. Уровень предприятия на языке SCg.

```
ОАО КолбасЗавод
∈ предприятие
```

Листинг 1. Уровень предприятия на языке SCn.

Следующий уровень – уровень мест производства. Он содержит информацию о производственных мощностях предприятия, разделенных по географическому признаку. Предприятие "КолбасЗавод" имеет производственные мощности в трех местах: Шатуре, Зарайске и Калуге. Соответствующая формализация представлена на рисунке 8 и листинге 2.

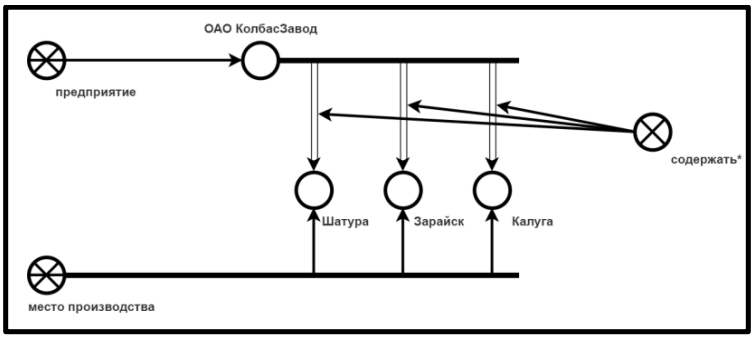


Рис.8. Уровень мест производства на языке SCg.

```
ОАО КолбасЗавод
∈ предприятие
=> содержать*:
    {
    • Шатура
      ∈ место производства
    • Зарайск
      ∈ место производства
    • Калуга
      ∈ место производства
    }
```

Листинг 2. Уровень мест производства на языке SCn.

Уровень областей производства делит место производства на несколько областей, которые отвечают за укрупненные процессы. Место производства в Калуге содержит две области производства, которые имеют непосредственное отношение к процессу рецептурного производства (рис. 9 и листинг 3).

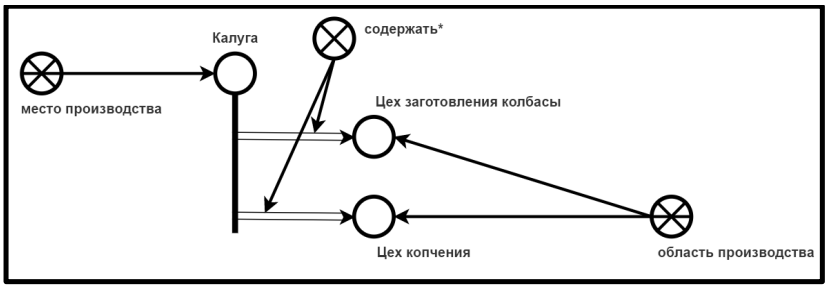


Рис.9. Уровень областей производства на языке SCg.

```
Калуга
∈ место производства
=> содержать*:
   {
   • Цех заготовления колбасы
     ∈ область производства
   • Цех копчения
     ∈ область производства
   }
```

Листинг 3. Уровень областей производства на языке SCn.

Уровень ячейки процесса описывает цех в отношении процессов, происходящих в нем. Цех заготовления колбасы включает в себя два основных процесса: производство мясной массы и формовка с фасовкой мясной массы, что представлено на рисунке 10 и в листинге 4.

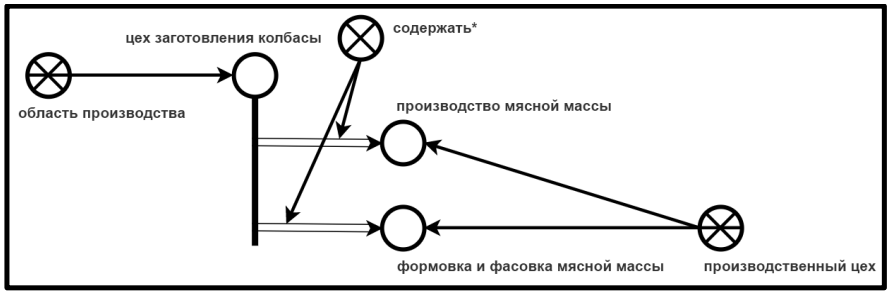


Рис.10. Уровень ячейки процесса на языке SCg.

```
Цех заготовления колбасы
∈ область производства
=> содержать*:
   {
   • производство мясной массы
     ∈ производственный цех
   • формовка и фасовка мясной массы
     ∈ производственный цех
   }
```

Листинг 4. Уровень ячейки процесса на языке SCn.

Оставшиеся уровни физической модели предлагается объединить ввиду их общей направленности – описания оборудования цехов. В производственном цехе производства мясной массы имеется линия приемки мяса, линия приемки специй, промышленная мясорубка, смеситель и экструдер. Данные факты представлены на рисунке 11 и в листинге 5.

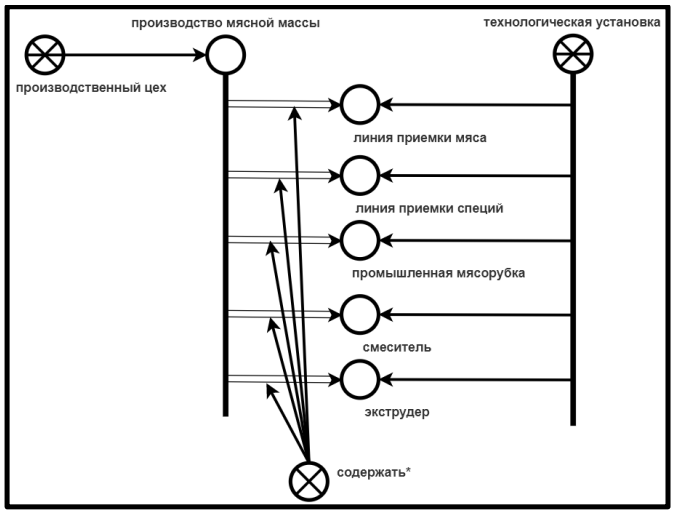


Рис.11. Нижние уровни физической модели на языке SCg.

```
Производство мясной массы
∈ производственный цех
=> содержать*:
  {
  • линия приемки мяса
    ∈ технологическая установка
  • линия приемки специй
    ∈ технологическая установка
  • промышленная мясорубка
    ∈ технологическая установка
  • смеситель
    ∈ технологическая установка
  • экструдер
    ∈ технологическая установка
  }
```

Листинг 5. Нижние уровни физической модели на языке SCn.

Представим пример формализации (рис. 12 и листинг 6) блока оборудования – камеры сушки колбасы. Два датчика, присоединенных к трубе, регистрируют изменения температуры и влажности воздуха в камере сушки. Регулирование характеристик воздуха в камере происходит за счет клапана и теплообменника. Клапан может быть во временном состоянии «Открыт» либо в состоянии «Закрыт».

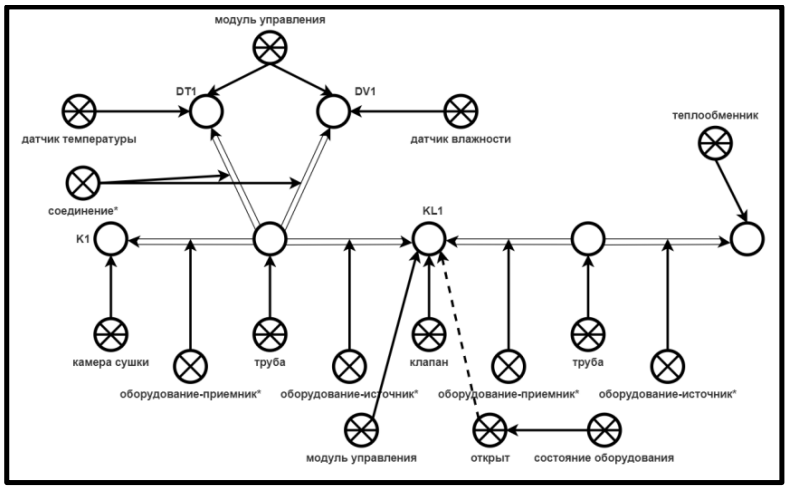


Рис.12. Формализация блока оборудования на языке SCg.

```
K1
∈ камера сушки
<= оборудование-приемник*:
   • T1
     ∈ труба
     => соединение*:
        {
        • DT1
          ∈ датчик температуры
          ∈ модуль управления
        • DV1
          ∈ датчик влажности
          ∈ модуль управления
        }
     => оборудование-источник*:
        • KL1
          ∈ клапан
          ∈ модуль управления
                        ∈ открыт
           ∈ состояние оборудования
     <= оборудование-приемник*:
        • T2
          ∈ труба
            => оборудование-источник*:
               • TO1
                 ∈ теплообменник
```

Листинг 6. Формализация блока оборудования на языке SCn.

3.2 Процедурная модель предприятия

В процедурной модели предприятия объектом исследования является класс процедурных элементов – составных частей процедурной модели. По уровню детализации различаются следующие их разновидности – процедуры производственного цеха, процедуры технологических установок, операции и фазы. Фрагмент процедурной модели, а именно декомпозиция операции смешивания, представлен в формальном виде на рисунке 13 и листинге 7.

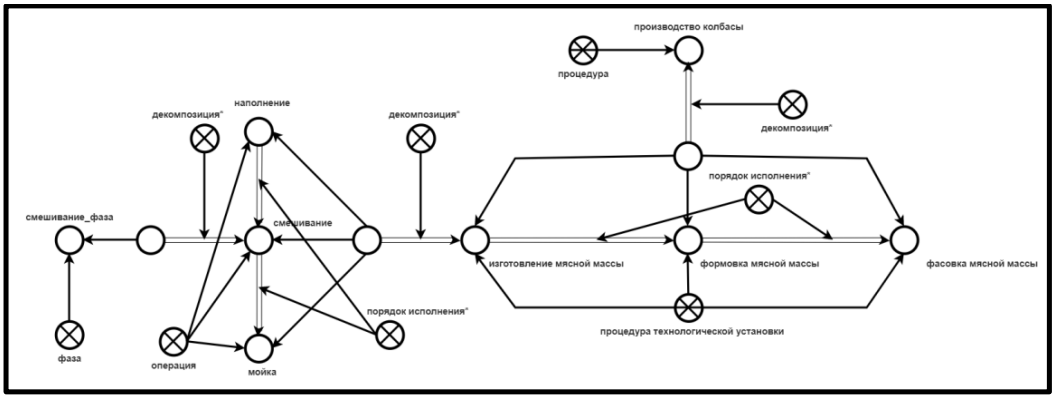


Рис.13. Фрагмент процедурной модели на языке SCg.

```
Производство колбасы
∈ процедура
=> декомпозиция*:
   {
   • изготовление мясной массы
     ∈ процедура технологической установки
     => порядок исполнения*:
        • формовка мясной массы
```

```
    => декомпозиция*:
      {
      • наполнение
        ∈ операция
        => порядок исполнения*:
          • смешивание
        => декомпозиция*:
          • наполнение_фаза
            ∈ фаза
      • смешивание
        ∈ операция
        => порядок исполнения*:
          • мойка
      • мойка
        ∈ операция
      }
  • формовка мясной массы
    ∈ процедура технологической установки
    => порядок исполнения*:
      • фасовка мясной массы
  • фасовка мясной массы
    ∈ процедура технологической установки
  }
```

Листинг 7. Фрагмент процедурной модели на языке SCn.

При помощи отношения «декомпозиция*» есть возможность описать внутреннюю структуру какого-либо элемента. При помощи отношения «порядок исполнения*» задается порядок, в котором следует выполнять элементы процесса. SC-код позволяет описывать в едином информационном пространстве как физическую модель предприятия, так и операции, выполняющиеся на отдельных объектах, что говорит о перспективности его использования.

4 Заключение

Была показана гибкость, присущая базе знаний OSTIS-системы, в качестве универсального хранилища данных о предприятии, начиная с уровня самого предприятия и заканчивая процессами, происходящими на отдельных единицах оборудования. Формы представления SCg и SCn являются относительно хорошо читаемыми для человека, что позволяет довольно легко обучить персонал для создания и сопровождения систем, которые используют данные формы записи. Это в свою очередь снижает затраты на проектирование моделей предприятия и сопровождение полученной в результате работы системы. При использовании единого стандарта формирования иерархий и онтологий наработки, полученные при создании модели одного предприятия, можно использовать на других предприятиях. Технология OSTIS находится в стадии активной разработки и является открытой, что располагает к ее дальнейшему изучению и совместному развитию.

5 Благодарности

Работа выполнена при частичной финансовой поддержке РФФИ, проект № 20-57-04002 «Модели и средства повышения эффективности обучения конечных пользователей и разработчиков интеллектуальных систем на основе открытых семантических и прагматических технологий».

Литература

[1] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A., Kaiser L., Polosukhin I. Attention is all you need [Электронный ресурс] // arXiv.org: an open-access archive. – 2017. – URL: https://arxiv.org/abs/1706.03762v5 (дата обращения: 10.06.2023).

[2] Schuster M., Nakajima K. Japanese and Korean voice search // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) – 2012. – P. 5149–5152. DOI: 10.1109/ICASSP.2012.628907

[3] Аникин М.А., Королева М.Н. Обработка нормативных правовых актов с применением нейросетевых моделей // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов XI Международной научно-практической конференции (ИММВ-2022, Коломна, 16-19 мая 2022 г.). В 2-х томах Т.1. – М.: РАИИ, 2022. – С. 182–188.

[4] Голенков В.В., Гулякина Н.А., Давыденко И.Т., Шункевич Д.В. Семантические технологии проектирования интеллектуальных систем и семантические ассоциативные компьютеры // Доклады БГУИР, 2019. – №3 (121). – С. 42–50.

[5] Шункевич Д.В. Семантические технологии проектирования решателей задач: учеб. пособие. – Минск : БГУИР, 2022. – 219 с.

[6] Taberko V., Ivaniuk D.S., Kasyanik V., Golovko V.A., Gulyakina N.A., Rusetski K.V., Shunkevich D.V., Boriskin A.S., Grakova N.V. Design of batch manufacturing enterprises in the context of Industry 4.0 // Open Semantic Technologies for Intelligent Systems (OSTIS-2018). Материалы международной научно-технической конференции (Минск, 15-17 февраля 2018 года) – Минск : БГУИР, 2018. – С. 307–320.

[7] ANSI/ISA-88.01-1995, Batch Control, Part 1: Models and Terminology, ISBN: 1-55617-562-0

Разработка онтологии предметной области архитектурной акустики

Артемьева И.Л.[1], Чусова А.Е.[2]

[1]*Департамент программной инженерии и искусственного интеллекта ДВФУ, п. Аякс, 10, о. Русский, г. Владивосток, 690922, Россия*

artemieva.il@dvfu.ru, chusova.ae@dvfu.ru

Аннотация. *Статья посвящена разработке онтологии предметной области архитектурной акустики, которая будет использована при создании интеллектуальной системы рекомендации вычислительных методов при решении прикладных задач данной области. Онтология содержит модули с описанием терминологии для задач моделирования звука в различных помещениях, подбора материалов для создания необходимого акустического оформления. Для рекомендации наиболее подходящего способа решения задачи в каждом модуле были выделены наборы свойств, характеризующие методы с точки зрения разных подходов к классификации. Подобный подход может быть использован для решения задач в других областях, а фунциональность системы может быть расширена за счет добавления новых модулей. Приведена архитектура программной системы.*

Ключевые слова: онтология, модель онтологии, интеллектуальная система, архитектурная акустика

1 Введение

Одним из важных факторов, оказывающих существенное влияние на эксплуатационные качества зданий, является учет при их проектировании и строительстве акустических требований. Картина распространения звука зависит от значений множества параметров, начиная от формы помещения и заканчивая характеристиками звука. Таким образом, для учёта всех этих управляющих параметров требуется создание специализированных систем моделирования и вычислительных моделей.

В данный момент разработано множество вычислительных моделей, однако на этапе выбора метода возникают противоречия в рекомендациях по использованию моделей для различных помещений. Причины предпочтения одной модели другой настолько взаимосвязаны и разнообразны, что опираться на один критерий не представляется целесообразным. Поэтому систематизация и описание знаний о методах и моделях решения задач в области архитектурной акустики, а также последующее создание системы поддержки принятия решений является актуальной задачей.

2 Теоретические проблемы

2.1 Архитектура программной системы

Программная система, в основе которой лежит онтология методов и моделей решения задач архитектурной акустики, спроектирована на основе модульного подхода и включает в себя:

- подсистему постановки акустической задачи, в которой пользователь указывает целевое назначение помещение и выбирает тип решаемой задачи;
- графическую подсистему, предоставляющая пользователю инструменты для ввода информации о помещении;

- онтологию предметной области;
- набор модулей для решения акустических задач и модуль для оптимизации и распараллеливания.

Программный комплекс имеет возможность добавления новых модулей для расширения области применимости системы. Помимо этого, описанный подход к проектированию позволяет адаптировать систему под другие предметные области [1].

2.2 Классификации методов

Наличие большого количества способов решения задач обуславливает необходимость в различных классификациях, так как каждый из них может быть применен для нескольких видов акустических задач и обеспечивать разную степень точности результатов в зависимости от конфигурации помещения.

Наиболее распространенным критерием классификации методов является принцип действия [2-3]. Классификация методов моделирования звука, основанная на принципе действия, представлена на Рис.1.

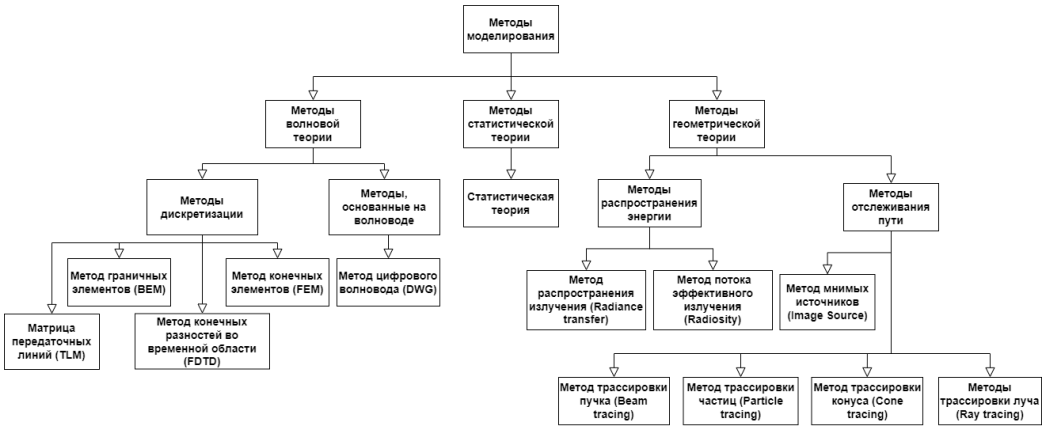


Рис.1. Классификация методов моделирования по принципу действия

Несмотря на естественность такого представления методов, данная классификация не позволяет определить применимость в полной мере, так как в одну и ту же группу могут быть включены методы, которые имеют сравнительно узкий диапазон использования [4]. Наличие дополнительной информации о моделях позволило бы в большей степени учитывать специфику помещения и запросы пользователя. Поэтому возникает необходимость в построении различных классификаций методов, учитывающие следующие факторы:

- сложность объекта моделирования;
- входных и выходных параметров;
- цели моделирования;
- объектов исследования;
- принадлежности модели к иерархическому уровню описания объекта.

Для построения классификаций по вышеперечисленным критериям необходимо выделить характеристики, описывающие методы каждого модуля.

2.3 Характеристики методов

Были выделены универсальные характеристики (см. Рис.2) и специфические, присущие только методам конкретного модуля.

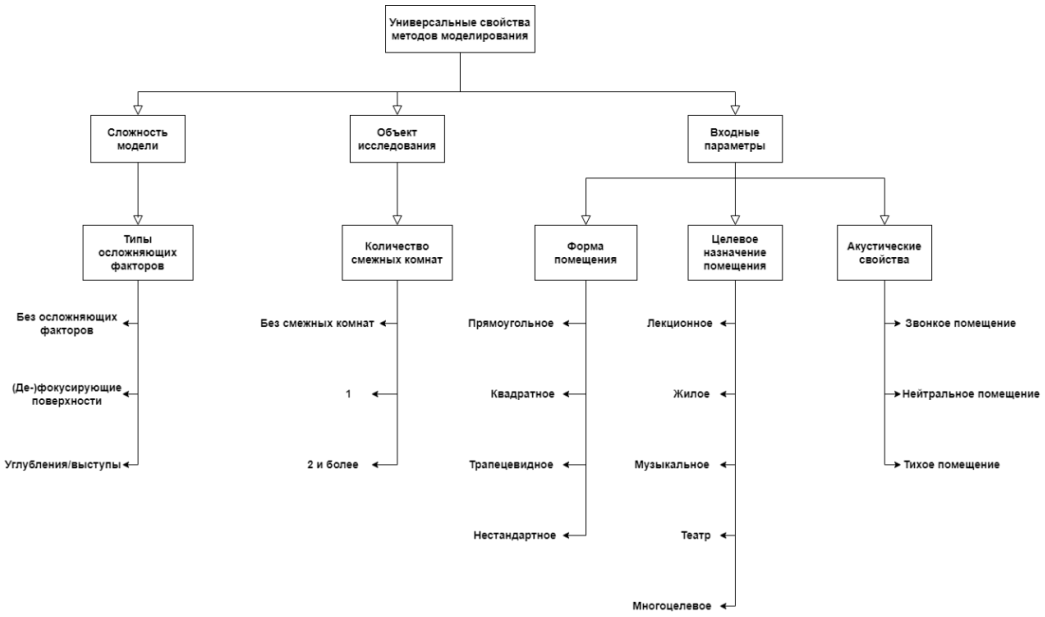


Рис.2. Универсальные свойства методов моделирования.

Под осложняющими факторами подразумеваются элементы помещения, которые существенно влияют на картину распространения звука. В помещениях с осложняющими факторами могут наблюдаться такие волновые явления как интерференция и дифракция, что вызывает потребность в использовании моделей и методов, учитывающих их влияние [6-9]. Учёт смежных комнат также влияет на выбор метода. Несмотря на невозможность учёта волновых явлений, точность результатов при использовании статистической теории в некоторых случаях может быть сравнима с геометрической [2]. Однако основополагающими характеристиками являются целевое назначение помещения и его форма [10-11], так как при проектировании существуют определенные принципы и требования к таким сооружениям.

Теперь рассмотрим специфичные для каждого модуля определен наборы характеристик. Для модуля моделирования жилых помещений следующий набор свойств:

- Моделирование эха
  - есть;
  - нет.
- Целевое назначение комнаты
  - комната отдыха;
  - домашний кинотеатр;
  - домашняя студия.

Лекционные помещения имеют ряд особенностей, например, необходимо обеспечить низкий уровень шумов и отсутствие эха, а также большое количество первых отражений, непосредственно влияющих на звучание. Применение методов моделирования звука в жилом помещении в большинстве случаев не является возможным ввиду отсутствия учёта волновых эффектов и, как следствие, низкой точности результатов. На основе этих особенностей были выделены следующие характеристики методов:

- Учёт волновых эффектов
  - есть;
  - нет.
- Наличие звукоусиления
  - есть;
  - нет.

- Необходимый эффект
  - голос бога;
  - чёткость речи;
  - эффект Хааса;

Учёт волновых эффектов, в частности дифракции, может быть определяющим факторов в выборе метода. В [Романюк, 2011] представлено исследование звукового оформления методом лучевого эскиза. Отмечается, что методы геометрической акустики имеют один значительный недостаток – отсутствие или недостаточный учёт дифракции. Классическими методы не учитывают влияние этого явления, однако существуют модификации, которые позволяют добавить дифракционные компоненты в решение.

Поскольку методы решения акустических задач зачастую имеют высокую вычислительную сложность, возникает потребность в применении оптимизаций. Поэтому для описания алгоритмов были выделены универсальные характеристики для модуля оптимизации и распараллеливания:

- Тип элемента
  - нет
  - плоский
  - объёмный
  - луч
- Зависимость от соседних элементов
  - да
  - нет
- Наличие источников
  - есть
  - нет
- Количество элементов на одно направление
  - 0
  - 1
  - 2 и более
- Матрица элементов
  - есть
  - нет

3 Заключение

В статье рассмотрена онтологии проблемной области «Решение задач архитектурной акустики», которая является основным компонентом интеллектуальной системы рекомендации методов и моделей решения задач архитектурной акустики. Онтология включает в себя модули, отвечающие за решение задач моделирования звука в различных помещениях, подбора материалов для создания необходимого акустического оформления, а также универсальный модуль оптимизации и распараллеливания алгоритмов. Для рекомендации наиболее подходящего под запросы способа решения задачи в каждом модуле были выделены наборы свойств, характеризующие каждый метод с точки зрения разных подходов к классификации. Подобный подход может быть использован для решения задач в других областях, а функциональность системы может быть расширена за счет добавления модулей определенной структуры. Описаны универсальные характеристики и специфичные для каждого модуля, приведена архитектура программной системы.

В дальнейшем планируется расширение набора характеристик для каждого модуля, а также создание инструмента для редактирования характеристик и задания различных классификаций.

Литература

[1] Найданов Ч. А. Разработка ядра онтологической модели, настраиваемой под предметную область // Вестник НГУ. Серия: Информационные технологии. 2019. №1. URL: https://cyberleninka.ru/article/n/razrabotka-yadra-ontologicheskoy-modeli-nastraivaemoy-pod-predmetnuyu-oblast.

[2] Summers, Jason & Torres, Rendell & Shimizu, Yasushi. (2004). Statistical-acoustics models of energy decay in systems of coupled rooms and their relation to geometrical acoustics. The Journal of the Acoustical Society of America. 116. 958-69. 10.1121/1.1763974.

[3] Savioja, L., & Svensson, P. (2015). Overview of geometrical room acoustic modeling techniques. Journal of the Acoustical Society of America, 138(2), 708-730. URL: https://doi.org/10.1121/1.4926438

[4] Щиржецкий Х.А., Сухов В.Н., Щиржецкий А.Х., Алёшкин В.М. К ПРОБЛЕМЕ АКУСТИЧЕСКОГО ПРОЕКТИРОВАНИЯ СОВРЕМЕННЫХ ЗАЛОВ МНОГОЦЕЛЕВОГО НАЗНАЧЕНИЯ // Жилищное строительство. 2019. №7. URL: https://cyberleninka.ru/article/n/k-probleme-akusticheskogo-proektirovaniya-sovremennyh-zalov-mnogotselevogo-naznacheniya

[5] Романюк Максим Алексеевич, Старченко Ирина Борисовна Теоретические исследования звукового поля в помещении и разработка интерактивной системы аудиоуправления // ИВД. 2011. №1. URL: https://cyberleninka.ru/article/n/teoreticheskie-issledovaniya-zvukovogo-polya-v-pomeschenii-i-razrabotka-interaktivnoy-sistemy-audioupravleniya.

[6] Z. Tang, L. Chen, B. Wu, D. Yu and D. Manocha, "Improving Reverberant Speech Training Using Diffuse Acoustic Simulation," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 6969-6973, doi: 10.1109/ICASSP40776.2020.9052932.

[7] Кузнецова А.И. Методы моделирования акустического пространства оперных театров // Вестник Академии русского балета им. А. Я. Вагановой. 2018. №1 (54). URL: https://cyberleninka.ru/article/n/metody-modelirovaniya-akusticheskogo-prostranstva-opernyh-teatrov

[8] Siltanen, Samuel. “EFFICIENT PHYSICS-BASED ROOM-ACOUSTICS MODELING AND AURALIZATION.” (2010).

[9] Nowoświat, Artur & Marcelina, Olechowska. (2016). Investigation Studies on the Application of Reverberation Time. Archives of Acoustics. 41. 10.1515/aoa-2016-0002.

[10] Сергеев М.В. Акустические свойства прямоугольных помещений различных пропорций / М.В. Сергеев // Акустический журнал. – 1979. – Т. 25, № 4. – С. 591-598.

[11] Канев, Н.Г. Реверберация в трапециевидном помещении / Н.Г. Канев // Акустический журнал. – 2013. – Т. 59, № 5. – С. 607-612.

Исследование подходов к анализу тональности в текстах узбекского языка

Э.Ю. Ахмедов

Новосибирский государственный университет, ул. Пирогова, д. 1, г. Новосибирск, 630090, Россия

e.akhmedov@g.nsu.ru

Аннотация. *Область исследования являются мнения пользователей о продуктах и сервисах, написанных в виде неструктурированных текстов на узбекском языке и доступны через социальные сети. Предметом исследования являются задачи извлечения информации о мнениях содержащий текст о продуктах и сервисов. В статье проанализированы существующие методы анализ тональности и исследованы современные методы обработка естественного языка для определения полярности текста узбекского языка и определены более эффективные с целью качественную классификации. Исследованы методы обработки естественного языка, основанные на правилах, словарях и модели, основанные на комплексе методов машинного обучения.*

Ключевые слова: тональность текста, алгоритмы на основе наивного Байеса, алгоритм на основе словарей, машинное обучение, нейронные сети, глубокое обучение, сегментация, лемматизация, полярность текста.

1 Введение

Анализ тональности текста может применяться во многих областях науки и производства.

Как определить эффективность мысли, отраженной в содержании текста, с помощью компьютера? Для этого отражает следующие свойство мнение людей:

- эмоция (чувство) - реакция людей на те или иные внутренние и внешние воздействия: гордость, гнев, радость, страх, печаль;
- настроение - отношение людей к действительности в результате определенного воздействия: депрессия, радость;
- социальные отношения - социальные отношения между людьми в конкретной ситуации: дружба, теплота, холодность;
- эмоциональные отношения - отношения между людьми, длящиеся в течение определенного периода времени: любовь, ненависть, желание;
- личное поведение - характерное для человека поведение людей: ревность, нервозность, ссоры.

При сентимент анализе создается лингвистическая база слов этого типа, тексты автоматически анализируются в зависимости от типа позитива и негатива. Сентимент анализ используется в различных сферах жизни общества:

- в области спорта (для определения рейтинга спортсменов)
- в коммерческом секторе (для оценки продукции)
- в сфере образования (для определения рейтингов вузов)
- в сфере туризма (для оценки показателей гостиниц) и другие.

2 Методы и алгоритмы анализа тональности

Целью исследования является изучение методов и подходов анализа тональности текстов на узбекском языке. Для достижения этой цели необходимо выполнить следующие задачи:

1. Обзор литературы: Изучить существующие исследования и методы анализа тональности текстов, включая как классические подходы, так и новейшие технологии, применяемые в данной области.
2. Идентификация основных компонентов: Разобраться в основных компонентах анализа тональности текстов, таких как создание словаря, определение правил и разработка алгоритмов анализа.
3. Изучение подходов: Изучить различные подходы, используемые для анализа тональности текстов, включая методы на основе правил, базовые методы машинного обучения и нейронные сети.

В большинстве проанализированных исследований применялись подходы на основе правил и базовые подходы к машинному обучению, в нескольких исследованиях использовались нейронные сети.

Для анализа тональности текста существует несколько алгоритмов, включая:
1. Алгоритм на основе словарей: этот алгоритм использует словари с положительными и отрицательными словами, чтобы определить тональность текста. Он считает количество положительных и отрицательных слов в тексте и присваивает ему соответствующую тональность.
2. Машинное обучение: этот алгоритм использует методы машинного обучения, чтобы определить тональность текста. Он обучается на большом количестве размеченных данных, чтобы научиться распознавать тональность слов и фраз.
3. Нейронные сети: этот алгоритм использует нейронные сети для анализа тональности текста. Он обучается на большом количестве данных и может учитывать контекст и семантику слов.
4. Глубокое обучение: этот алгоритм использует глубокие нейронные сети для анализа тональности текста. Он может обрабатывать большие объемы данных и учитывать сложные зависимости между словами и фразами.

Это только некоторые из алгоритмов, используемых для анализа тональности текста. Каждый из них имеет свои преимущества и недостатки, и выбор конкретного алгоритма зависит от конкретной задачи и требований.

Для анализа тональности на казахском языке ведутся серьезные исследования. В исследованиях Ергеш Бану для создания тональный словарь казахского языка определены следующие классы:

1. Прилагательные слова и словосочетания (значение в пределах: [-2; 2]);
2. Существительные [-1;1];
3. Слова и фразы, связанные с глаголом [-1;1];
4. Наречия (усиливающие).

Словарь создан вручную и определен по 5-бальной шкале. Тональность измеряется с значениями -2 – крайне негативный, -1 – негативный, 0 – нейтральный, 1 – позитивный, 2 – крайне позитивный и определяется среднеарифметическая величин измерения.

Влияют текстовую полярность на казахском языке части речи существительные, глагол, прилагательное и наречие. Результаты показывают, что существительное является объектом обсуждения, а прилагательные в основном определяют полярность текста.

С помощью созданного словаря и правил был разработан алгоритм определение тональности текстов по казакхскому языку. Этот алгоритм имеет следующую структуру:

- сбор и подготовка текстовых данных;
- морфологический анализ;
- оценка тональности по словарю;
- оценка цепочек по правилам;
- результат анализа.

Тональных словарей возможно создать как вручную, так и автоматически. Автоматический легко и быстро создается [5].

В корпусе русского языка каждая слова имеет свой тег. Ev:posit => «гений», «малыш», Ev:neg=> «обжора», «прилипала». Исходя из этого нейтральная слова на русском лицо – юз на узбекском языке можно определить по значениям тональности следующим образом:

Таблица 1. Определение слова «юз-лицо» по степени тональности

Очень плохой	**Плохой**	**Нейтральный**	**Хороший**	**Лучший**
-2	-1	0	1	2
Башара	Афт	юз	чехра	Хушсурьат

Как правило, комментарии, полученные из социальных сетей, не соблюдают правил грамматики, различных знаков, аббревиатур и т. д. Поэтому в таком случае предварительная обработка данных позволяет добиться хороших результатов. С появлением технологий глубокого обучения и их применением в обработке естественного языка появилась возможность повысить точность этих методов в двух основных направлениях: предобработка данных и использование обученных и необученных нейронных сетей в обучении кластеризаторов и классификаторов.

В трудах Кадырбек Н. К. [6] были исследованы текстовые документы с обучением нейронной сети, для которых была использована выборка из 1000 мнений пользователей. Эти данные проходят следующие этапы предварительной обработки:

1. Удаление лишних символов: оставление только буквенных символов.
2. Сегментация: каждое мнение разбивается на предложения, а предложения на токены.
3. Лемматизация: процесс нормализации токенов (их инициализации).

Например, лемма для токена «телефоны» - это «телефон». Основное различие между обычными нейронными сетями и рекуррентными сетями заключается в том, что рекуррентные сети зависят от времени. В рекуррентных сетях, каждый временной шаг (time step) обрабатывает не отдельные слова, а последовательность слов. То есть, входной последовательностью является не каждое отдельное слово, а целая последовательность слов. Каждый элемент входной последовательности (слово) связывается с определенным временным шагом, который указывает на порядок следования слов в последовательности. На практике количество временных шагов в цепочке равно ее максимальной длине.

Каждый временной шаг h_t связан с новым компонентом, называемым скрытым вектором состояния. Следовательно, x_t - это вектор, содержащий всю информацию, относящуюся к конкретному слову, а h_t - вектор, накапливающий информацию с предыдущих временных шагов. Скрытое состояние является функцией вектора текущего слова, а также вектора скрытого состояния из предыдущего временного шага.

$$h^t = \sigma(W^H h_{t-1} + W^X x_t) \qquad (1)$$

W являются весовыми матрицами. Входной вектор умножается на матрицу весов W^X, а вектор скрытого состояния на предыдущем временном шаге умножается на рекуррентную матрицу весов W^H. W^H - матрица, которая остается неизменной на всех временных шагах, а матрица измерений W^X различна для каждого входного сигнала.

Эти весовые матрицы влияют либо на текущее, либо на предыдущее скрытое состояние вектора скрытого состояния. Блоки долговременной памяти - это модули, которые можно размещать внутри рекуррентных нейронных сетей. Долговременная и кратковременная память (Long Short-Term Memory, LSTM) - это особый тип модуля, который используется внутри рекуррентных нейронных сетей для эффективной работы с долгосрочными зависимостями в последовательных данных, таких как тексты, речь и временные ряды. В LSTM блоках, в отличие от обычных рекуррентных сетей, вводятся специальные компоненты, называемые "воротами" (gates), которые позволяют контролировать поток информации и сохранять или забывать определенные аспекты данных. Они состоят из трех основных элементов: входного ворота (input gate), забывающего ворота (forget gate) и выходного ворота (output gate). Долговременная память в LSTM блоках представляет собой состояние, которое может сохранять информацию на протяжении длительного времени. Забывающий ворот контролирует, какая информация должна быть забыта, а входной ворот определяет, какая новая информация должна быть добавлена к текущей долговременной памяти. Таким образом, долговременная память позволяет модели сохранять и использовать информацию о важных зависимостях на протяжении длительных периодов времени. Кратковременная память в LSTM блоках представляет собой текущее скрытое состояние, которое может передаваться между временными шагами в последовательности. Она используется для учета текущего контекста и генерации соответствующего выхода или передачи

информации следующим временным шагам. Таким образом, долговременная и кратковременная память в LSTM блоках работают вместе, чтобы модель могла эффективно улавливать и использовать как краткосрочные, так и долгосрочные зависимости в данных.

На высоком уровне они контролируют, способен ли скрытый вектор состояния h инкапсулировать информацию о долгосрочных зависимостях в тексте.

Приведенная выше формулировка RNN подход не может объединять информацию, разбитую на несколько временных шагов. С точки зрения единиц LSTM, единицы принимают текущий вектор слов x_t и выводят скрытый вектор состояния h_t.

С использованием сетей LSTM анализе тональности текстов на казахском языке для обучения нейронной сети предварительная обработка текстов способствует повышению качества модели [6].

Для создания модели сентимент анализа текстов на казахском языке используются результаты морфологического и синтаксического анализа. Для определения тональности используются следующие словосочетания:

1. [N] · [V]
2. [N] · [V] · [Negation]
3. [ADJ] · [N]
4. [ADJ] · [Negation] · [N]
5. [ADJ] · [V]
6. [ADJ] · [V] · [Negation]
7. [ADV] · [ADJ]
8. [ADV] · [N];

Здесь, ADJ - это прилагательное, N - существительное, Negation - отрицательное слово (например, «эмес» или «жоқ»), V - глагол, а ADV – наречие [7].

Таблица 2. Обозначения для моделирования сентимент анализа

Обозначение	**Назначение**
α,β,γ,…ζ,ξ,…	Множество слов языка-переменные
ω	$\omega = \zeta \cdot \alpha \cdot \beta \cdot \xi$-лексические единицы (непустое слово или словосочетание)
L	Множество предложений языка
N	Множество имен существительных
Adj	Множество прилагательных
Pron	Множество местоимений
V_Post	Множество положительных форм глаголов
V_Negt	Множество отрицательных форм глаголов
AdvIntens	Множество наречий для усиления
Sent	Установление сентимента-предикат
@	Слова отрицания эмас, йўқ – константы
ד	Превращения в отрицательную форму-операция
+	конкатенация-операция

Для анализа тональности текста на узбекском языке можно использовать различные подходы. Один из них - это использование алгоритмов машинного обучения, таких как метод опорных векторов (SVM) или наивный Байесовский классификатор. Для этого необходимо иметь набор размеченных данных, где каждый текст помечен как позитивный, негативный или нейтральный. Затем можно обучить модель на этих данных и использовать ее для анализа тональности новых текстов на узбекском языке.

Метод опорных векторов (SVM) - это алгоритм машинного обучения, который используется для классификации и регрессии. Он основан на идее поиска оптимальной гиперплоскости, которая разделяет данные разных классов в максимально возможном расстоянии. SVM может использоваться для решения задач бинарной классификации, многоклассовой классификации и регрессии.

Основная идея метода SVM заключается в том, чтобы найти оптимальную гиперплоскость, которая разделяет данные разных классов. Гиперплоскость выбирается таким образом, чтобы максимизировать расстояние между ней и ближайшими точками обучающей

выборки, называемыми опорными векторами. Это позволяет SVM быть устойчивым к выбросам и обобщать на новые данные.

Для решения задачи классификации SVM использует функцию решающего правила, которая принимает входные данные и возвращает предсказанный класс. Функция решающего правила определяется весами и смещением, которые оптимизируются в процессе обучения. Обучение SVM включает в себя нахождение оптимальных значений весов и смещения путем решения оптимизационной задачи.

SVM имеет несколько вариантов, включая линейный SVM, SVM с ядровыми функциями и SVM с мягким зазором. Линейный SVM использует линейную гиперплоскость для разделения данных, в то время как SVM с ядровыми функциями позволяет проецировать данные в пространство более высокой размерности, где они могут быть линейно разделимыми. SVM с мягким зазором позволяет некоторым точкам попадать внутрь разделяющей гиперплоскости, чтобы учесть возможные шумы или неоднозначности в данных.

Метод SVM широко применяется в различных областях, включая компьютерное зрение, биоинформатику, финансовую аналитику и многие другие. Он обладает хорошей обобщающей способностью и может быть эффективным в задачах с большими объемами данных.

Наивный Байесовский классификатор - это алгоритм машинного обучения, который используется для классификации текстовых данных. Он основан на принципе условной независимости признаков, что означает, что каждый признак считается независимым от других признаков при условии заданного класса. Для анализа тональности текста на узбекском языке с использованием наивного Байесовского классификатора, необходимо предварительно подготовить обучающую выборку, состоящую из текстов с известной тональностью. Затем, на основе этой выборки, строится модель классификатора, которая будет использоваться для классификации новых текстовых данных на положительную или отрицательную тональность. Однако, для более точного анализа тональности текста на узбекском языке, может потребоваться дополнительная предобработка данных, такая как удаление стоп-слов, лемматизация и т.д.

Рекуррентные нейронные сети (RNN) - это тип нейронных сетей, который широко используется для анализа последовательностей данных, таких как текст. Они могут быть применены для анализа тональности узбекского языка. RNN обладают способностью учитывать контекст и последовательность слов в тексте, что позволяет им более точно определить тональность.

3 Анализ тональности текстов на узбекском языке

В трудах Сметанина С.И. [1] исследованы этапы анализа тональности текстов. Они включают создание словаря, определение правил для вычисления тональности предложений и разработку алгоритма анализа. Исследованиях применялись подходы на основе правил, базовые методы машинного обучения и нейронные сети для анализа тональности текстов.

По результатам исследования получены следующие выводы:

Применение предварительно обученных языковых моделей с использованием трансферного обучения дает эффективные результаты в задаче классификации тональности.

Для проведения эксперимента доступны несколько языковых моделей, подходящих для анализа тональности текстов на узбекском языке: M-BERT, RuBERT и M-USE. M-BERT уже получила признание среди исследователей, занимающихся анализом неанглоязычных языков.

Для проведения анализа тональности текстов на узбекском языке можно использовать выбранную языковую модель (например, M-BERT, RuBERT или M-USE) и применить трансферное обучение с использованием размеченных данных на узбекском языке. Это позволит модели изучить особенности узбекского языка и классифицировать тексты по тональности.

Важно отметить, что для достижения оптимальных результатов в анализе тональности текстов на узбекском языке может потребоваться больше размеченных данных и дополнительные настройки модели.

В целом, применение предварительно обученных языковых моделей с использованием трансферного обучения представляется перспективным подходом для анализа тональности текстов на узбекском языке. Однако для получения более точных результатов рекомендуется проводить дополнительные эксперименты, а также учитывать специфику узбекского языка и его лексические особенности при разработке алгоритмов анализа тональности.

В трудах [2 – 4] узбекских ученых дают развернутый обзор ясно, что на данный момент проведение анализа тональности текстов на узбекском языке и создание тезауруса для этого

языка не были осуществлены. Однако, существуют некоторые ресурсы и инструменты в области обработки естественного языка (Natural Language Processing, NLP) для узбекского языка, хотя требуются дальнейшие улучшения.

По результатам исследование получены следующие выводы:

В отношении лексикона эмоций, упомянутый лексикон ассоциации словесных эмоций NRC содержит положительные и отрицательные слова, которые могут быть использованы в контексте анализа тональности текстов на узбекском языке. Однако, такой лексикон может быть недостаточно полным и точным для узбекского языка, учитывая его уникальные лексические и семантические особенности.

Также упоминается NRC Hashtag Sentiment Lexicon, созданный на основе положительных и отрицательных настроений, выраженных в хэштегах Twitter. Этот лексикон может быть полезным для анализа тональности текстов, содержащих хэштеги на узбекском языке, но его применимость может быть ограничена.

Однако, необходимо отметить, что для более точного и надежного анализа тональности текстов на узбекском языке требуется создание специализированных ресурсов и моделей NLP, адаптированных к этому языку. Это включает в себя составление корпуса узбекских текстов с разметкой тональности, создание словаря с узбекскими словами и их соответствующей тональностью, а также разработку алгоритмов, учитывающих специфические особенности узбекского языка.

Для достижения этой цели можно рассмотреть следующие подходы:

1. Проведение исследования и сбор размеченного корпуса узбекских текстов различной тональности, что позволит создать обучающую выборку для моделей анализа тональности.
2. Разработка узбекского лексикона с учетом семантических особенностей узбекского языка и его выражения тональности.
3. Использование методов машинного обучения и моделей NLP, например, нейронных сетей, для обучения моделей анализа тональности на узбекском языке с использованием собранного корпуса данных и разработанного лексикона.

В целом, разработка ресурсов и моделей NLP для анализа тональности текстов на узбекском языке является перспективным направлением и требует дальнейших исследований и усилий. Применение существующих языковых моделей, таких как M-BERT, RuBERT или M-USE, может быть полезным для узбекского языка.

В трудах [5–7] казахских ученых по анализу тональности на казахском языке проводились с использованием созданного вручную тонального словаря, который включал различные части речи (прилагательные, существительные, глаголы и наречия) и определял их тональность на 5-балльной шкале. Разработанный алгоритм определения тональности текстов на казахском языке включал сбор и подготовку данных, морфологический анализ, оценку тональности по словарю, оценку цепочек по правилам и вывод результатов анализа.

Также было упомянуто использование технологий глубокого обучения и предварительной обработки данных для повышения точности методов анализа тональности текстов на казахском языке.

В целом, исследования по анализу тональности на казахском языке показывают, что создание тональных словарей и использование алгоритмов на основе правил и машинного обучения, включая нейронные сети, могут быть эффективными методами для определения тональности текстов на этом языке.

По результатам исследование получены следующие выводы:

Для анализа тональности текстов на узбекском языке можно рассмотреть следующие подходы:

1. Сбор и разметка корпуса узбекских текстов с различной тональностью. Это позволит создать обучающую выборку для моделей анализа тональности.
2. Создание узбекского лексикона, включающего различные части речи и их связь с тональностью. Это может быть выполнено вручную или автоматически с использованием методов машинного обучения.
3. Применение методов машинного обучения, включая нейронные сети, для обучения моделей анализа тональности на узбекском языке с использованием размеченного корпуса данных и созданного лексикона.

4. Использование предварительной обработки данных, включая удаление лишних символов, сегментацию текста, для повышения точности методов анализа тональности на узбекском языке.

4 Заключение

В данной статье рассматривается исследование, проведенное по анализу тональности текста. Анализы проведённых работ, описанные в данной работе, станут полезным ресурсом для специалистов, занимающихся компьютерной обработкой узбекского языка для решения прикладных задач анализа мнений, классификации текстовых документов, извлечения информации, кластеризации и др. Область применения анализ тональности текста очень обширна и может быть использована при оценке, мониторинге, анализе продукции и рекомендательных системах государственных и негосударственных предприятий в области образования, науки, искусства, здравоохранения и других социальных сфер.

Важно отметить, что разработка ресурсов и моделей NLP для анализа тональности на узбекском языке является активной областью исследований, и реализация этих подходов требует дальнейших исследований и усилий. Применение существующих языковых ресурсов и инструментов, разработанных для других языков, также может потребовать адаптации и расширения для работы с узбекским языком.

Литература

[1] Сметанин С.И. Анализ тональности текстов из социальных сетей на основе методов машинного обучения для мониторинга общественных настроений: дис. … канд. техн. наук: 1.2.2. / С. И. Сметанин – М., 2022. - 162 с.

[2] Абдурахманова, Н. З. Теоретические вопросы сентимент анализа на узбекском языке [Текст] / Н. З. Абдурахманова // Международный журнал Искусство слова. - 2023. - № 1. - С. 37-40.

[3] Kuriyozov E., Salaev U., Matlatipov S., Matlatipov G. Text classification dataset and analysis for Uzbek language / E. Kuriyozov, U. Salaev, S. Matlatipov, G. Matlatipov [Текст] // Proceedings of the 10th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. - Poznan, Poland: Wydawnictwo nauka i innowacje, 2023.

[4] Matlatipov G., Rahimboeva H., Rajabov J., Kuriyozov E. Uzbek Sentiment Analysis based on local Restaurant Reviews / G. Matlatipov, H. Rahimboeva, J. Rajabov, E. Kuriyozov [Текст] // CEUR Workshop proceedings. - Koper, Slovenia:Sun SITE Central Europe, 2022. - С. 126-136.

[5] Ергеш Б. Ж. Определение тональности текстов на казахском языке на основе эмоционального словаря / Б. Ж. Ергеш [Текст] // Пятая Международная конференция по компьютерной обработке тюркских языков «TurkLang 2017». - Казань:Академии наук Республики Татарстан, 2017. - С. 63-66.

[6] Кадырбек, Н. К. Использование сетей LSTM в анализе тональности документов на казахском языке [Текст] / Н. К. Кадырбек // Вестник КазНПУ им. Абая, серия Физико-математические науки. - 2020. - № 1. - С. 366-370.

[7] Ергеш, Б. Ж., Шарипбай, А. А., Бекманова, Г. Т. Модели и методы сентимент анализа текстов на казахском языке / Б. Ж. Ергеш, А. А. Шарипбай, Г. Т. Бекманова [Текст] // Вычислительная обработка казахского языка. - Алматы:Қазақ университеті, 2020. - С. 95-96.

Разработка программной платформы взаимодействия устройств интернета вещей на основе онтологического инжиниринга

Балчиков Василий Игоревич, Ланин Вячеслав Владимирович

Пермский государственный национальный исследовательский университет, ул. Букирева, д. 15, г. Пермь, 614990, Россия.

gitbleidd@gmail.com, vlanin@live.com

Аннотация. *В связи с активным развитием Интернета вещей (IoT) количество всевозможных устройств стремительно возрастает. Несмотря на то, что существует понимание необходимости унификации способов взаимодействия устройств, они все еще остаются недостаточно разработанными. В большинстве случаев устройства способны взаимодействовать только с программным обеспечением, предоставляемым конкретным поставщиком. Это в свою очередь приводит к зависимости от конкретных систем и не позволяет объединять существующие устройства из различных систем. В данной работе предлагается решение проблемы функциональной совместимости устройств на основе методов онтологического инжиниринга в виде программной платформы. Архитектура платформы рассматривает каждый компонент IoT как независимые единицы и позволяет использовать общепринятые стандарты для их взаимодействия. В результате разработанная платформа, позволяет взаимодействовать с устройствами IoT на более высоком уровне, используя формализованное описание возможностей и функций устройств в виде онтологии, а также повторно использовать существующие модули для обеспечения взаимодействия и добавлять новые при необходимости.*

Ключевые слова: функциональная совместимость устройств, интероперабельность устройств, интернет вещей, онтологический инжиниринг, онтология oneM2M, программная платформа, MQTT

1 Введение

В настоящее время количество устройств в интернете вещей стремительно возрастает и несмотря на множество различных проблем в IoT, связанных с энергоэффективностью, автоматической реконфигурацией архитектуры, безопасностью и конфиденциальностью устройств [1], одной из основных нерешенных проблем остается функциональная совместимость устройств между собой [2].

На данный момент нет единого стандарта общения устройств IoT. Сеть интернета вещей состоит из огромного количества разнородных устройств, которые имеют различные возможности, взаимодействуют между собой на основе различных протоколов [3]. Многие поставщики устройств интернета вещей, такие как Amazon, Google или Apple на сегодняшний день предоставляют собственные вертикально масштабируемые решения. В качестве таких решений обычно выступают системы, состоящие из устройств, шлюза для взаимодействия устройствами и облачных сервисов для управления устройствами через шлюз. Очевидным преимуществом в данном случае является то, что в системе конкретного поставщика, устройства могут взаимодействовать между собой без каких-либо проблем. Однако у данного подхода, есть большой недостаток. Устройств различных поставщиков зачастую являются несовместимыми между собой, в результате чего потребители начинают зависеть от конкретных систем, поставщиков, не имея возможности подключить сторонние устройства и оставаясь

ограниченными в возможностях конфигурирования. Вследствие этого горизонтально масштабируемые решения являются более предпочтительными. Они позволяют рассматривать каждый компонент – устройство, шлюз, сервис, как отдельную независимую единицу и использовать общепринятые стандарты для взаимодействия устройств. В результате разработчики могут сконцентрировать на создания самих устройств, а не поиске решений в области совместимости устройств между собой.

Для решения данной проблемы возможно прибегнуть к методам онтологического инжиниринга [4], на основе которых построить формализованное описание возможностей, функций устройства в виде онтологии, с целью дальнейшего использования в качестве некоторой универсальной модели для построения взаимодействия с данным устройством.

Используя семантические описания устройств, появляется возможность взаимодействовать с ними на более высоком уровне, чем на уровне какого-то конкретного прикладного протокола. Платформа, построенная на основе методов онтологического инжиниринга, позволит переводить сообщения устройств различных систем между собой, наладив таким образом коммуникацию между ними.

2 Задача взаимодействия устройств IoT

На текущий момент большинство существующих работ сосредоточены на семантическом аннотировании данных устройств для решения проблемы функциональной совместимости. Такой подход решает проблему только на уровне данных, то есть необходимо, чтобы устройства могли предоставлять описание о себе, доступные функции, формат данных и т. д. Подход с аннотирование данных не может быть использовать для огромного числа уже существующих устройств – необходимо обновление устройств, часть из которых вовсе могут не иметь такой возможности, например, проприетарные устройства. В работе рассматривается подход, который объединяет сразу несколько различных аспектов совместимости устройств IoT. Подобные решения, базирующиеся на нескольких уровнях совместимости, являются недостаточно изученными в современной литературе. Разрабатываемая платформа является попыткой совместить подходы и предоставить средство для более построения более гибких и эффективных систем интернета вещей.

Взаимодействие устройств интернета вещей можно рассмотреть на трех различных уровнях [5]. Первый уровень – уровень среды описывает с помощью каких протоколов физического, канального, сетевого уровней устройства взаимодействуют между собой. На данный момент наиболее распространёнными протоколами с низким энергопотреблением являются ZigBee, ZWave, Bluetooth [6]. Также на данном уровне используются такие протоколы, как Ethernet, WiFi и, кроме того, возможно проводное соединение устройств с использованием других протоколов. Платформа предполагает возможность разрешения совместимости устройств на данном уровне, строя взаимодействия используя их формализованное описание в виде онтологий.

Вторым уровнем для взаимодействия является прикладным. Он работает поверх уровня среды и ориентирован на обмен сообщениями между устройствами. На данном уровне наиболее распространёнными протоколами на данный момент являются MQTT (Message Queue Telemetry Transport), CoAP (Constrained Application Protocol), AMQP (Advanced Message Queuing Protocol) [7]. Каждый протокол обладает уникальным характеристиками, структурой сообщений и соответственно предназначен для различных типов систем интернета вещей. В следствии этого невозможно избежать применения различных прикладных протоколов. Однако при использовании онтологического подхода должна требоваться лишь разработка компоненты для работы с конкретным прикладным протоколом. После этого можно использовать описание возможностей устройств, другими словами метаинформацию, и на основе нее реализовать взаимодействие с другими устройствами. Используя описание устройств, можно конвертировать сообщения между отправителями и получателями, а обращаясь к соответствующей компоненте передавать сообщения между устройства.

Кроме того, стоит отметить не менее важный уровень взаимодействия – уровень данных. В большинстве случаев датчики предоставляют сырые данные, то есть без какой-либо дополнительной информации. В следствии этого, для обработки этих данных необходимо знать, что именно данные содержат и в каком формате передаются сообщения. Более сложные устройства могут содержать метаданные в сообщение, однако в связи с тем, что нет единого стандарта аннотации передаваемых данных, в каждом таком решении используется свой способ описания данных, из-за чего также требуется дополнительные усилия при обработке такого рода данных. Онтология устройства включает непосредственно описание функций, аргументов в них,

возвращаемых значений, поэтому используемые подход уже включает решение описанной проблемы.

2.1 Обзор существующих решений

В современных работах, основывающихся на онтологиях, предлагаются различные модели и архитектуры для решения проблемы совместимости устройств. Как было отмечено ранее большинство работ сосредоточены на семантическом аннотировании данных датчиков, но также есть небольшое число решений, которые все же используют онтологии в качестве средства описания возможностей устройств для построения дальнейшего взаимодействия с ними, коротко рассмотрим их для того, чтобы можно выделить недостатки, а также сильные стороны.

Платформа NETCONF-MQTT [8] представляет собой одно из решений, использующих описанный подход. В основе данной платформы лежит мост NETCONF-MQTT, который принимает от устройств (использующих протокол MQTT) семантическое описание их возможностей и позволяет конфигурировать и управлять устройствами используя протокол NETCONF. Мост генерирует YANG модели функций устройств, используя которые становится возможным генерировать программный интерфейс для работы с устройством. Мост конвертирует соответствующие запросы NETCONF в MQTT запросы для устройств. Данная платформа ориентирована только на MQTT устройства, и не предполагает возможности добавления других протоколов. Устройства в данной системе должны хранить описание и предоставлять его. Из этого вытекает проблема, заключающаяся в том, что не все устройства располагают большим количеством памяти на борту и могут реализовать возможность его предоставления.

В работе Д. Пратиккумара предлагается архитектура Semantic Gateway as Service (SGS) [5] для построения платформы. Она основана на использовании шлюзов, которые являются посредниками между устройствами IoT и облачными сервисами [9]. В SGS используются семантические веб-технологии и стандарты для аннотации данных и обеспечения совместимости функциональных возможностей систем IoT на уровне служб. SGS не учитывает совместимость устройствами IoT на основе различных протоколов и стандартов.

Концептуальная модель Device and Service Discovery (DSD) в работе Чинденга Э. и др. [10] предлагает модель, основанную на сервис-ориентированной архитектуре, которая обеспечивает связь между данными устройств и поддерживает динамическое изменение среды, обнаружение и добавление новых устройств. Ограничением данной модели выступает необходимость аннотирования передаваемых данных самим устройством с помощью собственной онтологии, которая не соответствует не одному существующему стандарту в этой области.

Фреймворк Task Computing предлагает использование промежуточного семантического уровня, а также сервисного уровня для решения проблемы совместимости устройств IoT [11]. Оба уровня позволяют обеспечить взаимодействие с вычислительной средой на уровне его семантического описания и создавать абстрактные сервисы для выполнения задач, скрывая детали взаимодействия с различными стандартами взаимодействия. Однако это решение также ставит требование аннотирования информации к устройствам.

Таким образом предлагаемое решение учитывает ограничения, содержащиеся в перечисленных работах, базируясь на все тех же методах онтологического инжиниринга, но подходя к проблему с другой стороны. Платформа, основанная на модульной архитектуре, работает с устройствами используя их онтологии, компоненты взаимодействия и конвертеры с помощью которых реализует цепочку преобразований сообщений отправителей во всевозможные форматы получателей и последующую доставку адресату.

2.2 Сравнение онтологий, связанных с IoT

Для описания устройств необходима онтология, от которой можно было бы оттолкнуться. В ходе обзора были выделены наиболее распространенные онтологии – SSN, oneM2M Base Ontology, SAREF, IoT-Lite. Для сравнения онтологий в качестве параметров были выбраны ключевые понятия, отраженные в онтологиях. На основе полученной информации была построена сравнительная таблица 1.

Каждая из онтологий, представленных в таблице в целом, по-своему моделирует область интернета вещей, но они имеют общие пересечения. Онтология SSN [12] (Semantic Sensor Network) построена вокруг датчиков и данных наблюдения. SSN фокусируется на измерениях и наблюдениях, игнорируя аппаратную информацию устройства и описывает датчики с точки

зрения их возможностей, производительности, наблюдений (измерений), процессов измерения и развертывания. Для большинства практических целей онтология требует расширения.

Онтология oneM2M [13] является относительно новой и используется для выделения ресурсов в сети описания выполнения операций чтения, записи из ресурсов и в данные ресурсы. oneM2M описывает устройства в достаточно широком смысле, тем самым позволяя определять свойства сети, функциональные возможности устройств, функционирование и службы. Идея данного подхода заключается в том, чтобы сделать возможным обнаружение семантически разграниченных ресурсов с использованием минимального набора понятий.

Таблица 1. Сравнение IoT онтологий.

	SSN	***oneM2M***	***SAREF***	***IoT-Lite***
Вещь (Thing)	+	+	+	+
Устройство	+	+	+	+
Возможности устройства	+	-	-	-
Развертывание устройства	+	неявно	-	+
Функции и сервисы	-	+	+	только сервисы
Датчики	+	-	неявно	неявно
Активные датчики (Actuators)	-	-	+	+

SAREF (The Smart Applications REFerence) [14] фокусируется на описании интеллектуальных устройств во всевозможных местах. Устройства в SAREF преимущественно характеризуются функциями, которые они выполняют, а также командами, которые они могут принимать и состояниями, в которых могут находиться. Данные три категории служат основой семантического описания. Элементы каждой категории можно комбинировать для создания более сложных описаний многофункциональных устройств.

IoT-Lite [15] является прямым расширением онтологии SSN, при этом сохраняя ориентир на описание датчиков, их измерений и высокую модульность и расширяемость. Основная цель данной онтологии заключается в создании легковесной семантической модели, которая бы могла быть менее требовательной к устройствам, которые ее используют, чем другие подобные модели. IoT-Lite расширяет SSN за счет добавления понятия активных датчиков, дополняющих обычные датчики и свойств охвата области действия датчиков. Для последнего дополнения IoT-Lite использует понятия из специальной онтологии геолокации.

Подводя итог можно сделать вывод, что для описание устройств в соответствии с выбранным подходом наиболее оптимальной онтологией будет oneM2M. Онтологии SSN и IoT-Lite не могут быть применены, так они сосредоточены на описании данных датчиков и их функций, при этом не предоставляют описания того, как устройства это делают. Онтология SAREF не отражает аспект, связанный с развертыванием устройств. В свою очередь понятия, предоставляемые онтологией oneM2M является хорошей отправной точкой для описания возможностей устройств.

3 Архитектура платформы взаимодействия IoT-устройств

Разрабатываемая платформа получила название Fusion. Она опирается на методы онтологического инжиниринга для решения проблемы функциональной совместимости устройств. Благодаря онтологии устройства появляется возможно получить формализованное описание сервисов, функций, предоставляемых устройством. Платформа в свою очередь, имея онтологическое описание устройства, может извлекать данные о том, как собственно устройство работает, какие методы взаимодействия использует и какие функции предоставляет. И благодаря этому предоставлять семантический уровень взаимодействия с устройством, отличный от привычного уровня какого-то конкретного прикладного протокола.

3.1 Обзор основных компонентов Fusion

Различные группы устройств использует различные механизмы коммуникации и протоколы, поэтому для работы с ними требуются различные компоненты связи. С точки зрения архитектуры платформы данные компоненты можно выделить в коммуникационные модули.

Коммуникационный модуль предоставляет работу с конкретным протоколом, реализуя при этом общий интерфейс для того, чтобы платформа могла работать со всеми модулями единым образом. Имея коммуникационные модули, появляется возможность работы с конкретными группами устройств, но для того, чтобы обеспечить связь между различными группами необходимы две другие составляющие – это брокер сообщений, а также семантический модуль.

Устройства используют различные форматы передаваемых данных и для того, чтобы можно было передать данный от одного устройства другому, во-первых, необходимо онтологическое описание устройств, где установлено соответствие между передаваемыми данными, во-вторых, компонента платформы, которая будет выполнять преобразование данных с помощью онтологии. В качестве такой компоненты, обеспечивающей совместимость между устройствами на уровне данных, будет выступать модуль – конвертер. Так как для различных устройств предполагается использование различных конвертеров, то данные модули в свою очередь должны соблюдать контракт интерфейса конвертеров, позволяя одинаково работать с ними.

Последним компонентом является брокер сообщений, отвечающий за связь между устройствами различных групп. Основной задачей данного модуля является – прием данных от устройства издателя, поиск подписчиков данного устройства и использование конвертеров для преобразование данных в формат соответствующих подписчиков. Для определения подписчиков устройства используется онтология устройства, в которой должно быть определено соответствие между устройствами и их функциями, которые генерируют и получают данные.

Процесс взаимодействия компоненты платформы можно описать следующим образом. При инициализации платформа считывает описания устройств и инициализирует конвертеры, коммуникационные модули. Далее платформа находится в ожидании получения сообщений. После того, как какой-либо из коммуникационных модулей получил сообщения оно передается брокеру сообщений. Брокер сообщений готовит список устройств-подписчиков, а также используя конвертер устройства-издателя и конвертеры для соответствующих устройств переводит исходные данные в формат онтологии издателя, а затем в формат устройств подписчиков. После этого выполняется отправка сообщений всем устройствам.

На Рис.1 представлена схема, содержащая архитектуру платформы в действии.

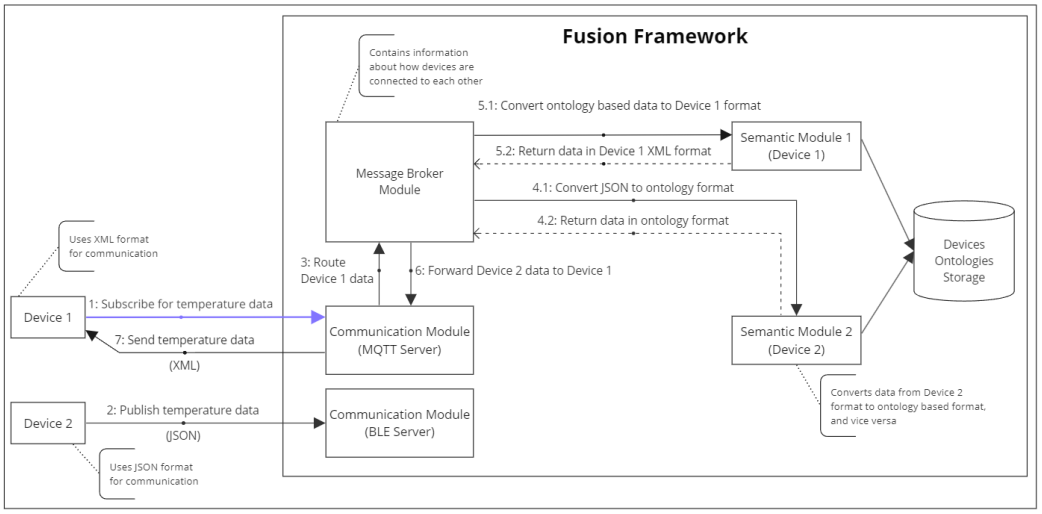


Рис.1. Архитектура Fusion с MQTT и BLE модулями в действии.

На данной схеме представлено всего два коммуникационных модуля, два устройства и семантические модули (конвертеры) для них. Устройства передают сообщения используя некоторые XML и JSON схемы. Данная схема позволяет наглядно продемонстрировать компоненты платформы и их взаимодействие между собой и то, как платформа обеспечивает решение проблемы функциональной совместимости устройств. Таким образом архитектура Fusion обеспечивает совместимость устройств на уровне среды, а также данных, позволяя устройствам с различными механизмами коммуникации и форматами данных взаимодействовать между собой. Таким образом, представленная архитектура предоставляет возможность с помощью онтологии переводить и передавать сообщения различных форматов от одних устройств другим, в результате, наладив коммуникацию между ними и решив проблему функциональной совместимости.

3.2 Подключение устройств к платформе

Предварительно для того, чтобы начать работу с конкретным устройством необходимо подготовить онтологическое описание или использовать уже существующее. Вместе с этим требуется разработка конвертера, который позволяет конвертировать данные из формата устройств в формат онтологии и наоборот. В случае коммуникационных модулей, если среди них не имеется ни одного для работы с определенной категорией устройств, то требуется выполнение программной реализации данного модуля. Однако большинство устройств интернета вещей используют лишь конечный набор механизмов взаимодействия, в числе которых есть наиболее часто используемое подмножество. Следовательно, можно сделать вывод, что при расширении возможностей платформы по работе с различными категориями устройств потребуется реализовать лишь ограниченный набор коммуникационных модулей.

Рассмотрим в качестве подключаемых устройств два микроконтроллера на базе ESP – NodeMCU ESP8266 и NodeMCU ESP32. Первый микроконтроллер представляет устройство, которое генерирует данные, а второй микроконтроллер устройство, который может не иметь встроенного функционала первого устройства, но имеет возможность принимать соответствующие данные в заданном формате из вне, причем форматы передаваемых данных устройствами отличаются. Для соответствующих устройств подготовлены онтологии, которые представлены на Рис. 2.

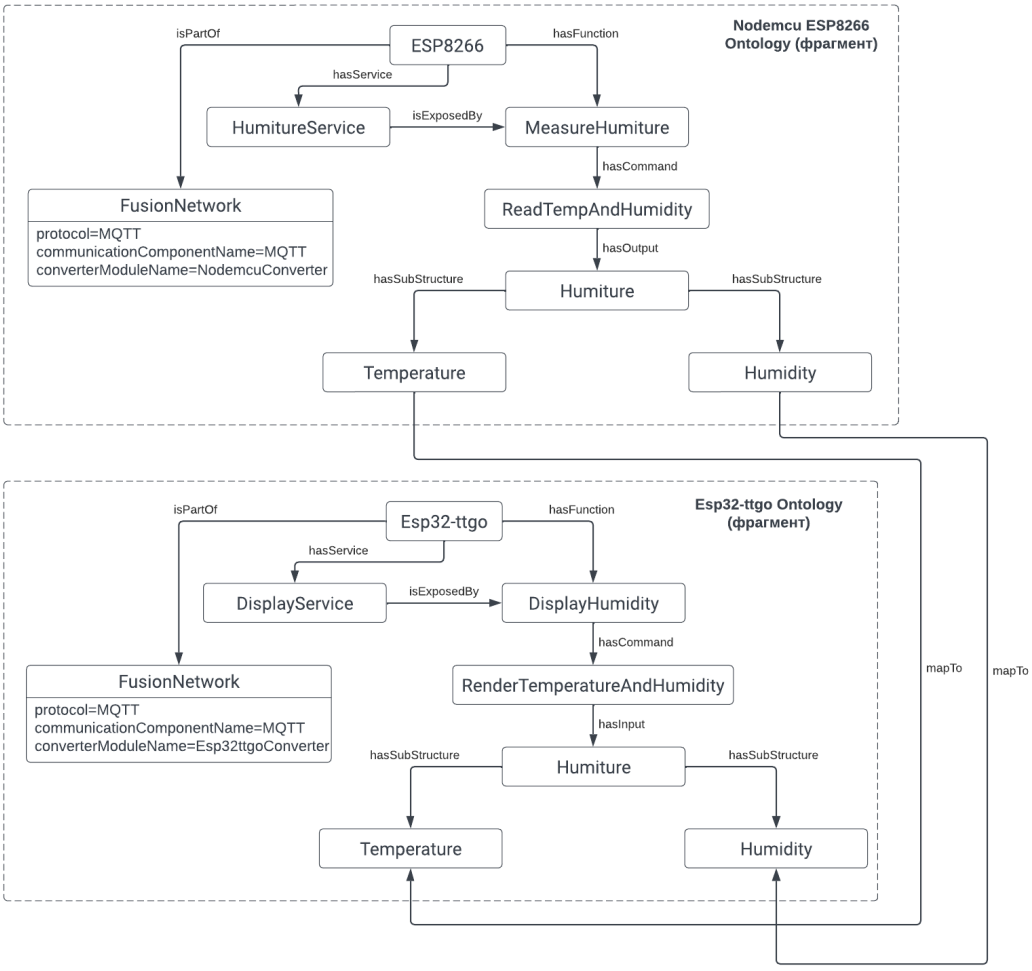


Рис.2. Онтологии устройств ESP8266 и ESP32 с отображением передаваемых данных.

Данные описания устройств построены на базе онтологии oneM2M, которая была выбрана в ходе сравнительного анализа. Описания содержат экземпляры классов, которые предоставляют информацию об устройство, его доступных сервисах, функциях и их получаемых и возвращаемых типах данных. Также для данных устройств построено отображение между передаваемыми данными.

Функция чтения влажности и температуры ESP8266, возвращает сложный тип данных, состоящий из двух простых типов. Непосредственно тип данных и его значение определяется в свойстве класса простого типа данных. Для влажности и температуры значение хранится в свойстве типа double – число с плавающей запятой с двойной точностью. Для того, чтобы платформа могла распознать какому устройству и как должны передаваться данные установлено одностороннее соответствие для данных температуры и влажности из функции ReadTempAndHumidity и RenderTemperatureAndHumidity для соответствующих устройств при помощи связи mapTo.

Также была построена схема для фрагмента онтологии ESP8266, представленная на Рис. 3, которая демонстрирует классы онтологии oneM2M использованные при описании устройства.

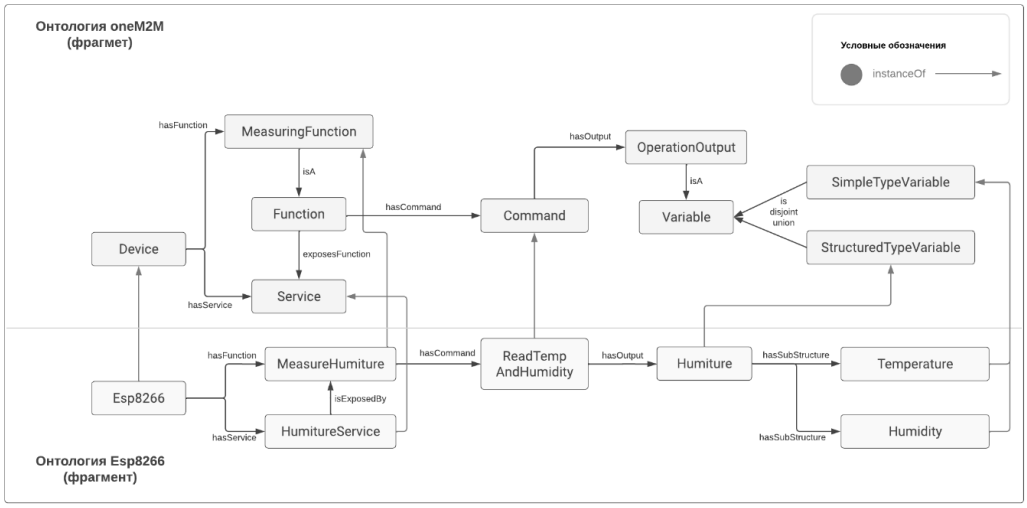


Рис.3. Фрагмент онтологии ESP8266 с классами oneM2M base ontology.

Рассмотрим работу платформы с микроконтроллерами. В начале инициализируются компоненты платформы, считываются онтологии устройств, и далее платформа ожидает подключений. Платформа обрабатывает широковещательное сообщение от устройств и сохраняет их адреса. Затем оба устройства подключаются к брокеру модуля MQTT. После этого коммуникационный модуль принимает сообщение и передает его брокеру сообщений. Брокер сообщений обращается к семантическому модуля и переводит сообщение из формата ESP8266 в формат устройства ESP32. Далее брокер просматривает подписчиков ESP8266, находит ESP32 и получает описание данного устройства, благодаря которому получает его коммуникационный модуль и вызывает метод передачи сообщений. ESP32 принимает входящие сообщения, после чего обрабатывает их и выводит на экран устройства.

4 Заключение

Рассмотренная архитектура платформы, в основе которой лежит использования онтологий устройств и модульная структура, позволяет решить одну из основных проблем в области устройств интернета вещей – функциональной совместимости устройств. Использованный подход показал реальную значимость позволив наладить коммуникацию между двумя несвязанными устройствами интернета вещей.

На данный момент платформы содержит необходимую базу для того, чтобы можно было перейти к этапу расширения числа ее компонентов и последующего их тестирования на реальных устройствах. Кроме того, платформа имеет возможность улучшения со стороны хранения онтологический описаний устройств, а именно переход к графовой базе данных. Еще одной стороной развития может стать разработка системы, представляющей полноценное приложение с пользовательским интерфейсом, базирующейся на реализованной платформе и добавление дополнительных параметров для описания устройств с целью повышения гибкости настройки взаимодействия с ними.

Литература

[1] Medagliani P., Leguay J., Duda A., Rousseau F., Duquennoy S., Raza S., Ferrari G., Gonizzi P., Cirani S., Veltri L., Montón M., Domingo M., Prieto M., Dohler M., Villajosana I., Dupont O. Internet of Things Applications - From Research and Innovation to Market Deployment. 1st ed. River Publishers, 2014.

[2] Androcec D., Vrcek N. Thing as a Service Interoperability: Review and Framework Proposal // 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud). Vienna: Institute of Electrical and Electronics Engineers (IEEE), 2016. P. 309-316. https://doi.org/doi:10.1109/FiCloud.2016.51

[3] Terminology for Constrained-Node Networks. Request for Comments: 7228 – RFC Editor United States, 2014. https://doi.org/doi:10.17487/RFC7228

[4] Jabbar S., Ullah F., Khalid S., Khan M., Han K. Semantic interoperability in heterogeneous IoT infrastructure for healthcare // Wireless Communications and Mobile Computing. Hindawi, 2017.

[5] Desai P., Sheth A., Anantharam P. Semantic Gateway as a Service Architecture for IoT Interoperability // 2015 IEEE International Conference on Mobile Services. New York: Institute of Electrical and Electronics Engineers (IEEE), 2015. P. 313-319. https://doi.org/10.1109/MobServ.2015.51

[6] Danbatta S. J., Varol A. Comparison of Zigbee, Z-Wave, Wi-Fi, and Bluetooth Wireless Technologies Used in Home Automation // 2019 7th International Symposium on Digital Forensics and Security (ISDFS). Barcelos: Institute of Electrical and Electronics Engineers (IEEE), 2019. P. 1-5. https://doi.org/10.1109/ISDFS.2019.8757472

[7] Naik N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP // 2017 IEEE International Systems Engineering Symposium (ISSE). Vienna: Institute of Electrical and Electronics Engineers (IEEE), 2017. P. 1-7. https://doi.org/doi:10.1109/SysEng.2017.8088251

[8] Sahlmann K., Scheffler T., Schnor B. Ontology-driven Device Descriptions for IoT Network Management. // 2018 Global Internet of Things Summit (GIoTS). Bilbao: Institute of Electrical and Electronics Engineers (IEEE), 2018. P. 1-6. https://doi:10.1109/GIOTS.2018.8534569

[9] Henson C., Sheth A., Thirunarayan K. Semantic perception: Converting sensory observations to abstractions // IEEE Internet Computing. Institute of Electrical and Electronics Engineers (IEEE), 2012. Vol. 16, Number 2. P. 26–34. https://doi.org/10.1109/MIC.2012.20

[10] Chindenga E., Gurajena C., Thinyane M. Towards an adaptive ontology based model for interoperability in internet of things (IoT) // 2016 IST-Africa Week Conference. Durban: Institute of Electrical and Electronics Engineers (IEEE), 2016. P. 1-8. https://doi.org/10.1109/ISTAFRICA.2016.7530599

[11] Song Z., Cárdenas A.A., Masuoka R. Semantic middleware for the Internet of Things // 2010 Internet of Things (IOT). Tokyo: Institute of Electrical and Electronics Engineers (IEEE), 2010. P. 1-8. https://doi.org/doi:10.1109/IOT.2010.5678448

[12] Semantic Sensor Network Ontology [Электронный ресурс] URL: https://www.w3.org/TR/vocab-ssn/ (дата обращения: 01.12.22).

[13] oneM2M Base Ontology [Электронный ресурс] URL: https://www.onem2m.org/technical/onem2m-ontologies (дата обращения: 24.10.22)

[14] Official ETSI portal for SAREF [Электронный ресурс] URL: https://saref.etsi.org/ (дата обращения: 01.12.22).

[15] IoT-Lite Ontology [Электронный ресурс] URL: https://www.w3.org/Submission/iot-lite/ (дата обращения: 01.12.22).

Разработка алгоритма для замены в текстах на узбекском языке диалектных слов на литературные

В.Б.Барахнин[1,2], Д.Менглиев[2,3], Ш.Мадиримов[2]

[1]Федеральный исследовательский центр информационных и вычислительных технологий, проспект Лаврентьева, 6, г. Новосибирск
[2]Новосибирский государственный университет, ул. Пирогова 1, г. Новосибирск
[3]Ургенчский филиал Ташкентского университета информационных технологий имени Мухаммада ал-Хоразмий, ул. аль-Хорезми 110, г.Ургенч (Узбекистан)

bar@ict.nsc.ru, shogunuz@gmail.com, s.madirimov@g.nsu.ru

Аннотация.

Статья посвящена разработке и реализации алгоритма обработки диалектных вариаций узбекского языка. Целью исследования является автоматизация процесса преобразования диалектных форм слов к их литературным аналогам. В работе обсуждается создание двух баз данных, представляющих южнохорезмский и североогузский диалекты узбекского языка, в которых литературные слова, включая словоформы, образованные всевозможными комбинациями аффиксов, связаны с их диалектными аналогами. Это позволяет алгоритму быстро и эффективно находить соответствующую литературную форму для входного диалектного слова.

Результаты тестирования демонстрируют высокую точность работы алгоритма на словах, которые заранее предполагалось удачно распознать и трансформировать в литературную форму. Хотя точность уменьшается при работе со словоформами, которые отсутствуют в базах данных, однако данный алгоритм имеет значительный потенциал и может быть улучшен в будущем. Он может оказаться полезным во многих областях, связанных с обработкой естественного языка, включая машинный перевод, информационный поиск, лексическую и семантическую обработку текстов, а также в областях, связанных с изучением и сохранением языкового разнообразия.

Ключевые слова: алгоритм морфологического анализа, узбекский язык, диалекты, база данных, литературные слова, аффиксы, преобразование словоформ, стемматизация,

1 Введение

Узбекский язык, богатый и разнообразный по своей природе, обладает широкой диалектной системой [1], которая отражает географическое разнообразие и культурное богатство страны. Однако эта диалектное разнообразие является преградой для эффективного общения [2], поэтому необходима замена диалектных слов и выражений на их литературные аналоги. Данная статья посвящена разработке алгоритма, способного автоматизировать этот процесс замены применительно к электронным документам на узбекском языке.

2 Обзор существующих алгоритмов и методологий

Автоматическая замена диалектных слов на литературные в текстах на любом языке − это сложная задача, требующая глубокого понимания лингвистики соответствующего языка. Она становится еще сложнее, когда речь идет о языках с богатой диалектной системой, таких как узбекский. На протяжении последних десятилетий было предложено несколько алгоритмов и методологий для решения этой проблемы, но все они имеют не только сильные, но и слабые стороны.

Один из основных подходов к автоматической замене диалектных слов − это использование традиционных методов обработки естественного языка, таких как морфологический анализ и семантическое моделирование. В рамках этого подхода текст разбивается на отдельные слова или фразы, которые затем анализируются и заменяются их

литературными аналогами на основе словарей и грамматических правил. Этот подход обеспечивает относительно высокую точность замены, но он может быть ресурсоемким и не всегда справляется с исключениями и неправильностями в диалектах (необходимо продумать все возможные исключительные случаи).

Второй основной подход — это использование методов машинного обучения и глубокого обучения, таких как нейронные сети и модели преобразователей. Эти методы обучаются на больших объемах данных и могут обрабатывать сложные зависимости между словами и их диалектными и литературными формами. Они также могут обрабатывать различные типы диалектов и обеспечивают высокую скорость обработки. Однако эти методы требуют больших объемов обучающих данных и могут быть сложными в настройке и интерпретации. Узбекский язык все еще является низко-ресурсным языком [3], поэтому большие объемы обучающих данных для него отсутствуют.

В работе [4] исследователи успешно создали новые двуязычные словари для пяти тюркских языков: турецкого, узбекского, азербайджанского, казахского и киргизского, − применяя API Google Translate. Дополнительно они смогли сгенерировать улучшенные векторные представления слов для узбекского языка. Для обучения использовались корпусы тюркских языков [5] с применением fastText [6], что дало значительное преимущество по сравнению с ранее доступными векторными представлениями для узбекского языка.

Исследователи также провели анализ перекрестных векторных представлений слов для всех пяти тюркских языков. С этой целью они использовали современные методы выравнивания, включая MUSE [7], VecMap [8] и Meemi [9]. Используя индукцию двуязычного словаря для проверки своих выравниваний, исследователи представили результаты для всех пяти языков с использованием различных словарей и векторных представлений слов, которые применялись для выравнивания векторных пространств.

Эта работа позволила исследователям проанализировать качество не только существующих и недавно созданных двуязычных словарей, но и векторных представлений слов, используемых для выравнивания встраиваний. Было замечено улучшение качества в одноязычном векторном пространстве после перекрестного выравнивания, что подтверждается дополнительной задачей: сентимент-анализ текстов на узбекском языке, использующий наборы данных и методы из [10].

Важность этой работы подчеркивается тем, что она, по мнению авторов, является первой такого рода для тюркских языков. Однако, несмотря на обнадеживающие результаты, авторы признают, что есть место для дальнейшего улучшения.

Предложены направления для будущих исследований, включая улучшение основных ресурсов, таких как одноязычные векторные представления слов и двуязычные словари, для языков с ограниченными ресурсами. Это поможет улучшить результаты в задачах перекрестной обработки языков. Вторым направлением является расширение языковой сферы, включая больше языков тюркской группы, что будет полезно для решения задачи машинного перевода между тюркскими языками. И, наконец, возможно пересмотреть выбор английского как языка-ссылки, и вместо этого выбрать родственный язык, вероятно, с большими ресурсами (например турецкий для тюркской группы). Это может привести к улучшению результатов.

В статье [11] авторы представляют разработанный алгоритм для синтеза словоформ казахского языка. Они начинают с разбиеня существительных казахского языка на флективные классы, что служит основой для представленного алгоритма. Алгоритм выделяется своей ясностью и относительной простотой воспроизведения. Эти характеристики делают его особенно полезным для генерации словоформ других изменяемых частей речи в казахском языке, прежде всего, глаголов. Авторы также разработали веб-приложение для генерации словоформ существительных казахского языка, доступное для широкой аудитории через Интернет. В ходе тестирования была продемонстрирована корректность работы данного приложения. Этот алгоритм представляет собой важный вклад в морфологический анализ казахского языка, обеспечивая практический инструментарий для генерации и анализа словоформ.

Статья [12] рассматривает процесс создания интеллектуального веб-приложения для морфологического анализа слов в узбекском языке. Узбекский язык является вторым по числу носителей среди языков тюркской группы, при этом в последние годы активно развивается узбекский сектор Интернета. В свете растущей важности узбекского языка как средства коммуникации в Республике Узбекистан, включая научные исследования, статья подчеркивает важность создания электронных ресурсов на узбекском языке, надежных систем перевода, а также других сфер применения компьютерных технологий для узбекского языка, прежде всего, необходимости надежных алгоритмов анализа содержания слабоструктурированных документов.

В контексте исследования была реализована система для генерации и стемминга (отсечения окончаний слов для приведения к базовой форме) словоформ узбекского языка. Некоторые использованные алгоритмы аналогичны ранее разработанным алгоритмам для других тюркских языков из-за морфологических особенностей слов.

В статье [13] авторы описывают разработку и внедрение системы для морфологического анализа словоформ тюркских языков. Одним из ключевых компонентов системы является алгоритм стемминга, основанный на усечении аффиксов до достижения корня слова. Однако, как подчеркивают авторы, при неправильном применении стемминг может привести к ошибочным результатам, таким как излишнее усечение и получение набора букв вместо корня слова.

Чтобы преодолеть указанные проблемы, авторы вводят дополнительные методы обработки словоформ. Эта система была реализована на языке программиоования Python и может быть использована как отдельная библиотека для встраивания в более крупные проекты или для морфологического анализа словоформ тюркских языков. Преимуществом разработанного решения является его простота, а также поддержка многоязычности.

В заключении авторы подчеркивают, что, несмотря на успешную реализацию и тестирование системы, есть некоторые ограничения, которые они планируют устранить в будущих версиях инструмента. В частности, пока что инструмент анализирует слова, используя только словарь корневых слов, и может воспринимать слово, отсутствующее в наборе данных (словаре), как недействительное слово.

3 Описание разработанного алгоритма

3.1 Постановка задачи

Цель представленной работы − создание алгоритма, который бы мог автоматически заменять диалектные слова на литературные в текстах на узбекском языке. Современные технологии обработки естественного языка предлагают новые и эффективные подходы к этой задаче.

Важность этой работы трудно переоценить. Она способствует не только улучшению междиалектного общения, но и укреплению узбекской литературной нормы, обеспечивая более консистентное использование литературной формы языка. Кроме того, разработанный алгоритм может стать ценным инструментом для научных исследований в области узбекской лингвистики, обеспечивая автоматическую обработку и анализ диалектных слов.

Ожидается, что результаты этого исследования будут способствовать улучшению междиалектного общения на узбекском языке и будут ценным вкладом в область исследований обработки естественного языка и машинного обучения.

В рамках данной работы будет разработан алгоритм по конвертации словоформ на диалектах, представленных в [14], где представлена классификация Е.Д.Поливанова по огузскому наречию, включающая 2 типа:

1-й тип. Южнохорезмский диалект (говоры Хивы, Ургенча, Шавата, Газавата, Хазараспа, Янгиарыка, Ханки). В качестве отдельного подтипа в данном диалекте выделен говор Шурахана;

2-й тип. Североогузский диалект (Иканско-Карабулакские говоры). В составе данного диалекта в качестве отдельного подтипа выделен говор поселка Багдад в Фарышском районе Джизакской области.

Кроме того, авторами включены слова некоторых подтипов южнохорезмского диалекта, применяемые в таких районах как Тупракалла и Янгибозор.

3.2 Предлагаемое решение

В данной работе представляется алгоритм для морфологического анализа и конвертации диалектных форм словоформ узбекского языка в их литературные аналоги. В качестве основы этого подхода были использованы предшествующие исследования [12], в результате которых были созданы две базы данных − для южнохорезмского и североогузского диалектов [14]. В этих базах литературные слова, включая словоформы, образованные всевозможными комбинациями аффиксов, были сопоставлены с их диалектными аналогами.

Основу этих баз данных составило корпус, содержащий более 80 тысяч корней слов, которые затем были преобразованы вручную в диалектные формы. После этого базы данных были дополнены сгенерированными словоформами, полученными с использованием скрипта, который осуществляет конкатенацию корней слов со всеми возможными вариантами аффиксов в соответствии с правилами словообразования узбекского языка.

Одним из ключевых моментов представленного подхода было обнаружение, что большинство аффиксов, используемых для создания литературных слов, также подходят для формирования диалектных аналогов. Это облегчило генерацию различных вариаций словоформ и в диалектах.

В результате, когда пользователь вводит слово на диалектном узбекском языке, алгоритм быстро ищет его в базе данных, используя хэш-таблицы, и возвращает литературный вариант этого слова в качестве результата. В случае, если алгоритм не выявляет совпадения из баз, начинается стемминг исходного слова, после каждой итерации стемматизации полученный результат сверяется с базами. При успешном результате выводится результат, а в ином случае стемминг продолжается, но не более трех итераций в целом. Для стемминга имеется база, состоящая из 120 аффиксов. Ниже прилагается фрагмент алгоритма, составленный на псевдо-коде.

```
// Создаем функцию для чтения данных из Excel файла
Функция ReadExcelFile(имя_файла):
  // Загружаем файл Excel
  workbook = открыть_файл(имя_файла)
  worksheet = получить_активный_лист(workbook)
  возврат worksheet
// Создаем функцию для создания словаря из пар слов
Функция CreateWordDict(worksheet):
  word_dict = пустой_словарь
  для каждой строки i в диапазоне(1, максимальное_число_строк(worksheet) + 1):
    word1 = прочитать_ячейку(worksheet, i, 1) // 1 - номер колонки с первым вариантом слова
    word2 = прочитать_ячейку(worksheet, i, 2) // 2 - номер колонки со вторым вариантом слова
    word_dict[word1] = word2
  возврат word_dict
// Создаем функцию для поиска второго варианта по первому варианту слова
Функция FindWord(word_dict, word):
  если word в word_dict:
    возврат word_dict[word]
  иначе:
    возврат 'Первый вариант слова не найден'
// Основной блок кода
worksheet = ReadExcelFile('words.xlsx')
word_dict = CreateWordDict(worksheet)
печать(FindWord(word_dict, 'искомое_слово'))
```

Алгоритм изображен на рисунке 1 в виде блок-схем.

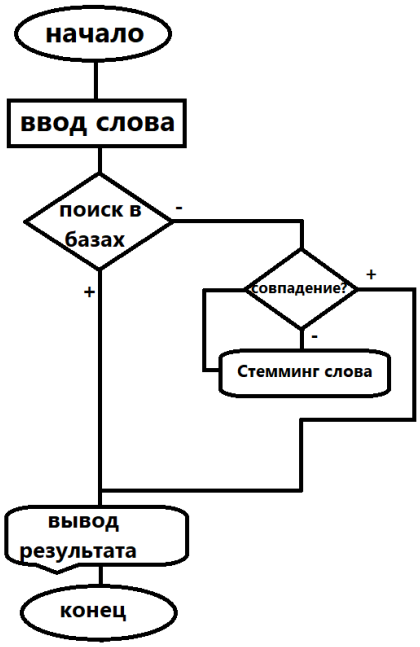


Рис.1. Схема работы алгоритма

3.3 Результаты тестирования алгоритма

В ходе экспериментальной проверки алгоритма была использована выборка из 2000 словоформ, каждый из двух диалектов представлен 1000 слов. С целью более глубокого анализа правильности функционирования алгоритма слова каждого диалекта были разделены на две подвыборки. Первая подвыборка включала 900 словоформ, которые, исходя из теоретических предположений, должны успешно обрабатываться алгоритмом и конвертироваться в литературный вариант (в связи с тем, что они уже имеются в базе). Вторая подвыборка состояла из 100 слов, которые отсутствуют в обеих рассматриваемых базах. Это могут быть либо полностью отсутствующие слова, либо преднамеренно сформированные с ошибками (например, вместо uyimda – uydimda). Результаты тестирования представлены в таблице 1.

Таблица 1. Результаты тестирования

	Первая подвыборка (900 слов)		***Вторая подвыборка (100 слов)***	
	Кол-во успешно замененных слов	***% точности***	***Кол-во успешно замененных слов***	***% точности***
Количество диалектных слов (южнохорезмский)	900	100	28	28
Количество диалектных слов (североогузский)	900	100	45	45

Исследование результатов работы алгоритма показало, что точность его функционирования для первой подвыборки достигает 100%. Это подтверждает предположение о том, что слова, содержащиеся в базах данных, будут успешно обработаны и преобразованы в литературный вариант. Однако при анализе работы алгоритма со второй подвыборкой наблюдается значительное снижение показателей точности. Это связано с тем, что в отсутствие прямого соответствия в базах данных, алгоритм проводит итерационную стемматизацию исходной словоформы с целью поиска совпадения.

4 Заключение

Таким образом, в данной работе представлен алгоритм, являющийся универсальным инструментом для анализа и трансформации диалектных форм узбекского языка в их литературные аналоги.

На основе предшествующих исследованияй было произведено вручную формирование баз данных, включающих литературные и диалектные словоформы южнохорезмского и североогузского диалектов. Данный процесс подразумевал множественные комбинации аффиксов, что, в свою очередь, повышает точность анализа и конвертации словоформ.

Специфика агглютинативных языков, к которым относится узбекский, предполагает возможность изменения значения слова с помощью добавления одного аффикса. Большинство аффиксов, используемых для формирования литературных слов, также применимы для создания диалектных аналогов, что существенно облегчило процесс генерации различных вариаций словоформ в диалектах.

Важной особенностью разработанного алгоритма является его простота и быстродействие. Когда пользователь вводит слово на диалектном узбекском языке, алгоритм оперативно ищет его в базе данных с использованием хэш-таблиц и возвращает литературный аналог этого слова.

Тем не менее, необходимо учитывать, что разработка находится в стадии усовершенствования: в дальнейшем планируется расширение функционала и возможностей алгоритма. Это позволит обеспечить повышение точности при анализе и конвертации словоформ, а также повысит степень универсальности алгоритма, что сделает его применимым не только для узбекского языка, но и для других языков тюркской группы.

Применение данного алгоритма может оказаться полезным во многих областях, связанных с обработкой естественного языка, включая машинный перевод, информационный поиск, лексическую и семантическую обработку текстов, а также в областях, связанных с изучением и сохранением языкового разнообразия.

Литература

[1] Мусаева Ф.Т. Диалектная лексика как объект изучения лингвокультурологии. Тюркское языкознание xxi века: лексикология и лексикография, Материалы международной научной конференции, посвященной 80-летию создания Института языка, литературы и искусства им. Г. Ибрагимова Академии наук Республики Татарстан. 2019

[2] Рахматова С.А., Кузибаева М.М. Общность и специфика диалектики и ее отражение в морфологии узбекского языка // Экономика и социум. 2021. №9 (88). URL: https://cyberleninka.ru/article/n/obschnost-i-spetsifika-dialektiki-i-ee-otrazhenie-v-morfologii-uzbekskogo-yazyka (дата обращения: 13.06.2023).

[3] Aharoni R., Johnson M., Firat O. Massively multilingual neural machine translation //arXiv preprint arXiv:1903.00089. – 2019.

[4] Kuriyozov, Elmurod & Doval, Yerai & Gómez-Rodríguez, Carlos. (2020). Cross-Lingual Word Embeddings for Turkic Languages.

[5] Baisa, V., Suchomel, V., et al. (2012). Large corpora for Turkic languages and unsupervised morphological analysis. In Proceedings of the Eighth conference on International Language Resources and Evaluation (LREC'12), Istanbul, Turkey. European Language Resources Association (ELRA).

[6] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching Word Vectors with Subword Information. Transactions of the Association of Computational Linguistics, 5(1):135–146

[7] Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jegou, H. (2018). Word translation without parallel data. ´ In Proc. of the 6th Int. Conf. on Learning Representations, ICLR 2018.

[8] Artetxe, M., Labaka, G., and Agirre, E. (2018). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, pages 5012–5019.

[9] Doval, Y., Camacho-Collados, J., Espinosa-Anke, L., and Schockaert, S. (2019a). Meemi: A simple method for post-processing cross-lingual word embeddings.

[10] Kuriyozov, E., Matlatipov, S., Alonso, M. A., and Gomez- ´ Rodr´ıguez, C. (2019). Deep learning vs. classic models on a new Uzbek sentiment analysis dataset. In Human Language Technologies as a Challenge for Computer Science and Linguistics – 2019, pages 258–262. Wydawnictwo Nauka i Innowacje.

[11] Барахнин В. Б., Лукпанова Л. Х., Соловьев А. А. Алгоритм построения словоформ с использованием флективных классов для систем морфологического анализа казахского языка // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2014. Т. 12, вып. 2. С. 25–32.

[12] Mengliev, D.; Barakhnin, V.; Abdurakhmonova, N. Development of Intellectual Web System for Morph Analyzing of Uzbek Words. Appl. Sci. 2021, 11, 9117. https://doi.org/10.3390/app11199117

[13] N. Z. Abdurakhmonova, A. S. Ismailov and D. Mengliev, "Developing NLP Tool for Linguistic Analysis of Turkic Languages," 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Yekaterinburg, Russian Federation, 2022, pp. 1790-1793, doi: 10.1109/SIBIRCON56155.2022.10017049.

[14] Решетов В.В., Шоабдураҳмонов Ш. Ўзбек диалектологияси.. — Ташкент: Ўқитувчи. — С. 234.

Метод Решения Задачи Слабо-контролируемой Групповой Классификации

Бериков В.Б.[1], Кутненко О.А.[1]

[1] *Институт Математики им. С.Л. Соболева СО РАН, проспект академика Коптюга 4, г. Новосибирск, 630090, Россия*

berikov@math.nsc.ru, olga@math.nsc.ru

Аннотация. *В работе рассматривается задача слабо-контролируемой групповой классификации. Разработан метод ее решения, основанный на выборе информативных признаков и цензурировании обучающей выборки. Приводятся результаты тестирования разработанного метода на задаче анализа снимков компьютерной томографии.*

Ключевые слова: слабо-контролируемое обучение, групповая классификация, информативные признаки, цензурирование объектов, компьютерная томография

1 Введение

В задаче слабо-контролируемого обучения учитывается возможная неопределенность или нечеткость разметки, см. обзор [1]. В данной работе формулируется постановка задачи слабо-контролируемой групповой классификации, когда каждый объект может включать набор подобъектов, относящихся к различным классам. Предлагаемый метод решения задачи основан на выборе информативного признакового пространства и построении информативной (по заданному критерию) обучающей выборки. Рассматривается случай задачи бинарной классификации.

Разработанный метод протестирован на задаче прогнозирования степени поражения участков головного мозга при ишемическом инсульте с использованием снимков компьютерной томографии. В данной задаче неточность разметки характерна для областей изображений, находящихся на границах контура очага поражения. Приведены результаты применения разработанного алгоритма и его сравнения с рядом известных алгоритмов.

2 Основные понятия

Пусть имеется совокупность объектов, описываемых множеством признаков X, $|\mathrm{X}| = d$, где d – размерность пространства признаков. Обозначим через $\vec{x} = \mathrm{X}(a)$ $(\vec{x} \in \mathbf{R}^d)$ вектор наблюдений признаков для объекта a. Дана метрика r, позволяющая находить расстояние $r(a,b)$ между любыми объектами a и b как в пространстве признаков X, так и в произвольном его подпространстве. Имеются два образа, которые обозначим A («положительный») и B («отрицательный»). Для объекта a задана величина $y_A(a)$ – степень принадлежности к образу A. Будем полагать, что $y_A(a) \in [0,1]$.

Для произвольного наблюдения $\vec{x}$ требуется прогнозировать его принадлежность $f(\vec{x})$ к образу A. При этом необходимо, чтобы оптимальная решающая функция давала бы минимально возможное значение ожидаемых потерь. Для построения решающей функции используется случайная обучающая выборка $S = \{\vec{x}_i, y_i\}_{i=\overline{1,N}}$, где $\vec{x}_i = \mathrm{X}(a_i)$, $y_i = y_A(a_i)$ – степень принадлежности объекта a_i к образу A, $i = 1,\ldots,N$. Оценка качества построенной

решающей функции $f(\vec{x})$ может быть определена как оценка риска по тестовой выборке S_t, сформированной независимо от обучающей выборки S. В случае абсолютной функции потерь соответствующий критерий выглядит как

$$R(f) = \frac{1}{N_t} \sum_{\vec{x}, y: \atop \vec{x} \in \mathbf{X}_t, y \in Y_t} \left| f(\vec{x}) - y \right|, \tag{1}$$

где $N_t = |S_t|$, $\mathbf{X}_t$ – набор наблюдений объектов S_t, Y_t – набор слабых разметок для объектов из S_t.

3 Метрики качества решения задачи групповой классификации

Далее будем рассматривать вариант постановки задачи слабо-контролируемого группового обучения, когда каждый объект может включать набор подобъектов как образа A, так и образа B. Определим метрики качества бинарной классификации. Предположим, что для произвольного объекта a получен прогноз степени принадлежности к классу A. Тогда для подобъектов объекта a можно определить следующие характеристики: TP – количество истинно положительных результатов, TN – число истинно отрицательных меток, FP – количество ложноположительных прогнозов, FN – число ложноотрицательных случаев.

Для оценки качества решающей функции на тестовой выборке вычисляются следующие характеристики:
точность

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%;$$

чувствительность (Sensitivity) – доля истинно положительных прогнозов:

$$Sens = \frac{TP}{TP + FN} \times 100\%;$$

специфичность (Specifisity) – доля истинно отрицательных меток:

$$Spec = \frac{TN}{TN + FP} \times 100\%;$$

а также сбалансированная точность:

$$BA = \frac{Sens + Spec}{2}.$$

Последние три показателя часто применяются, когда данные несбалансированы, то есть частоты встречаемости разных классов существенно различаются.

4 Метод решения

В задачах машинного обучения часто возникает необходимость решать две дополнительных задачи: выбор информативного признакового пространства и формирование информативной (с точки зрения заданного критерия) и представительной обучающей выборки. Стратегия формирования обучающей выборки заключается в удалении объектов, плохо описываемых моделью, используемой для классификации. Как выбор информативных признаков, так и удаление «шумовых» объектов осуществляется на основе анализа локального окружения объектов. При решении будем использовать метод k ближайших соседей (kNN).

Пусть $\Omega(a)$ – множество, состоящее из k ближайших соседей по метрике r для объекта a. Определим решающую функцию для прогнозирования целевого признака $\hat{y}_A(a)$ по наблюдению $\vec{x} = \mathrm{X}(a)$:

$$f(\vec{x}) = \frac{1}{C} \sum_{z \in \Omega(a)} n_A(z) e^{-\gamma r(z,a)}, \tag{2}$$

где $C = \sum_{z \in \Omega(a)} n(a) e^{-\gamma r(z,a)}$ – нормализующая константа. Здесь $\gamma > 0$ – заданный параметр, $n(a)$ – количество подобъектов a, $n_A(z) = y_A(z) n(z)$. Соответственно, $\hat{y}_A(a) = f(\vec{x})$.

При решении задачи выбора информативного пространства предлагается использовать алгоритм AdDel [2]. Добавлять и исключать признаки можно как по одному, так и группами (гранулами), состоящими из нескольких признаков. В используемом при построении информативного признакового пространства алгоритме GRAD [3] алгоритм AdDel работает на множестве наиболее информативных гранул, состоящих из заданного количества признаков.

В случае рассмотрения целевого признака $y_A(a)$ как характеризующего степень принадлежности объекта a к образу A, «выброс» – это такой объект, что $l = \left|\hat{y}_A(a) - y_A(a)\right| > l^*$, где l^* – пороговое значение ошибки. Исключение (цензурирование) подобных объектов позволяет повысить информативность обучающей выборки и уменьшить ошибку прогнозирования контрольных объектов. Для получения количественной оценки информативности обучающей выборки предлагается использовать описанный выше критерий (1). Когда количество исключенных объектов увеличивается, возникает проблема переобучения модели, и ее прогнозы становятся менее надежными. Чтобы учесть этот эффект, применяется нормирующий (штрафной) коэффициент $\left(N / N^*\right)^q$, где N – исходное число объектов в выборке, а N^* – число объектов, оставшихся после удаления объектов-выбросов; $q > 0$ – параметр. Итоговая оценка информативности обучающей выборки задается как

$$I = \left(\frac{N}{N^*}\right)^q R(f)\,. \tag{3}$$

Конкретная стратегия процедуры цензурирования и критериев ее остановки зависит от решаемой прикладной задачи.

5 Экспериментальное исследование

Оценка эффективности предлагаемого метода решения задачи слабо-контролируемого обучения (в задаче групповой классификации) проводилось на реальных медицинских данных, полученных в Международном Томографическом Центре СО РАН.

Данные представляют собой цифровые снимки бесконтрастной компьютерной томографии (КТ) головного мозга у пациентов с ишемическим инсультом. Зоны поражения на снимках были аннотированы специалистами-рентгенологами. Примеры изображений срезов головного мозга с соответствующей разметкой представлены на Рис. 1.

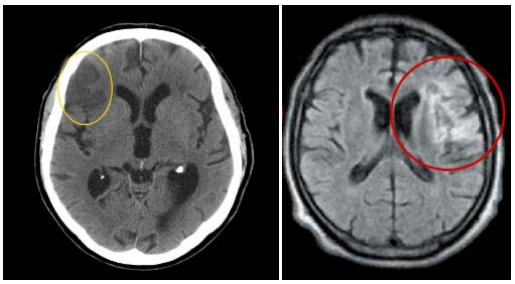


Рис. 1. Примеры КТ-изображений (обведены зоны поражения).

В данной работе были использованы анонимизированные данные 24 пациентов. Анализируемая выборка представлена 8043 объектами – участками КТ-снимков головного мозга со степенью поражения от 0% до 100%, описанными в 30-мерном признаковом пространстве. Подробное описание извлекаемых из изображения признаков приведено в [4]. Целевой признак – степень поражения области, оцененная по уровню пересечения с сегментационной маской.

Предварительно данные были нормализованы к диапазону [0,1]; для вычисления расстояний между объектами использовалась метрика Евклида. Оценка качества метода проводилась с помощью процедуры скользящего экзамена по пациентам. Лучшие результаты были получены при $k = 7$. В (2) выбрано $\gamma = 40$. Для выбора информативного признакового пространства применялся алгоритм GRAD. Параметры алгоритма фильтрации: $l^* = 0.5$, в (3) $q = 5$. Удаление нетипичных объектов проводилось последовательно: на j-ом шаге удалялась группа объектов с $l > 1 - j \times 0.1$, $l > l^*$.

Для оценки качества предложенного метода вычислялись указанные в разделе 3 метрики качества и ошибка определения степени поражения. Среднее по скользящему экзамену значение критерия (2) было равно 0.185, стандартное отклонение – 0.006. В Таблице 1 для предложенного алгоритма и ряда других алгоритмов приведены усредненные метрики качества решений, полученные с помощью процедуры скользящего экзамена. Сравнение проведено с алгоритмами регуляризации многообразия [4], случайного леса (Random Forest), опорных векторов (SVM) и *k*NN. Указаны наилучшие результаты, полученные для различных параметров работы алгоритмов.

Таблица 1. Усредненные метрики качества алгоритмов

Название алгоритма	*Значения метрик*		
	Sens, %	*Spec*, %	*BA*, %
Предложенный алгоритм	86.8	73.4	80.1
Алгоритм [4]	82.7	77.3	80.0
Random Forest	77.0	77.4	77.2
SVM (histogram intersection kernel)	68.8	34.3	51.6
*k*NN ($k = 4$)	42.8	61.1	52.0

По результатам сравнения видно, что качество решений предложенного алгоритма оказалось максимальным по чувствительности, и сравнимым с другими лучшими алгоритмами по сбалансированной точности. Разработанный алгоритм позволил дополнительно указать набор информативных признаков.

6 Заключение

В работе предложен метод решения задачи слабо-контролируемой групповой классификации с использованием выбора информативного признакового пространства и цензурирования обучающей выборки. Приведены результаты тестирования метода на задаче прогнозирования степени поражения участков головного мозга при инсульте по снимкам компьютерной томографии. Результаты сравнения с рядом известных алгоритмов машинного обучения подтвердили достаточно высокую эффективность разработанного метода. В отличие от других аналогичных алгоритмов, метод позволяет выбрать наиболее информативные признаки, что важно для повышения качества и интерпретируемости решений. Планируются дальнейшие исследования для повышения надежности и устойчивости слабо-контролируемого распознавания, учитывающие число подобъектов анализируемых объектов.

7 Благодарности

Работа выполнена при финансовой поддержке РНФ (грант № 22-21-00261).

Литература

[1] Zhou Z-H. A brief introduction to weakly supervised learning National science review. 2018; 5(1):44-53.

[2] Zagoruiko N.G., Kutnenko O.A. Recognition methods based on the AdDel algorithm. Siberian Journal of Industrial Mathematics. 2004; 7(1(17)):39-47.

[3] Загоруйко Н.Г. Когнитивный анализ данных. Новосибирск: Академическое издательство ГЕО; 2013 С. 186.

[4] Berikov V., Litvinenko A., Pestunov I., Sinyavskiy Y. On a weakly supervised classification problem. Lecture Notes in Computer Science. Springer. 2022; 13217:315-329. DOI:10.1007/978-3-031-16500-9_26.

Характеристические функции вычисления значимости терминов в семантической модели представления научных знаний

Вопилова Е.В.

Алтайский государственный технический университет им. И.И. Ползунова, пр. Ленина, д. 46, г. Барнаул, 656038, Россия.
Polzunov Altai State Technical University, Lenina ave., 46, Barnaul, 656038, Russia

vopilova.elena@gmail.com

Аннотация. *В докладе описывается формирование модели представления научных знаний предметной области, извлеченных из тематической энциклопедии. Предлагаемая модель реализована в виде взвешенного ориентированного графа, вершины которого содержат термины домена, связанные семантическими связями различного типа. Предложены варианты расчета характеристической функции значимости семантических связей между терминами домена, а также вариант расчета количественной оценки качества связей вершины графа. Проведены эксперименты по расчету значимости связи между терминами в соответствии с вариантами расчета, произведено отсечение незначащих связей и узлов семантического графа, проведен анализ результатов предложенных вариантов расчета и структуры полученного графа.*

Ключевые слова: семантический граф, автоматическая обработка неструктурированных текстов, средства обработки естественного языка

1 Введение

Рост объемов информации и развитие сети Интернет стимулируют развитие методов автоматической обработки научных публикаций с целью анализа трендов и новых направлений исследований в научных статьях, сравнения предлагаемых методов для тех или иных задач, поиска нужной информации. Как следствие, разработка инструментов для автоматической обработки текстов научной тематики сегодня является одной из актуальных задач. Такие тексты отличаются не только особой лексикой, но и определёнными синтаксическими и семантическими структурами, состоят из блоков, следующих в определенном порядке. Более того, ключевые слова и заголовки разделов позволяют автоматически отнести весь текст или его часть к определенной тематике. Все это позволяет рассматривать научные тексты как частично структурированные. Особенно это касается тематических энциклопедий.

2 Семантическая модель представления научных знаний

2.1 Выбор обучающих данных

Модель представления знаний о предметной области должна содержать не только специфическую терминологию домена, но и семантические отношения между базовыми понятиями домена [1]. Источником базовых доменных понятий может служить энциклопедия соответствующей научной области, в настоящее время энциклопедические издания, как правило, представлены в электронном варианте. Текст энциклопедии имеет определенную структуру: описание каждого термина выделено в отдельный блок, который содержит термин в качестве заголовка, список синонимов, краткое определение и основной разъясняющий текст [2]. В данной работе предлагается метод построения взвешенных семантических отношений между

извлекаемыми из научных текстов терминами и устойчивыми словосочетаниями. Результаты вычислительных экспериментов приведены на примере обработки математической энциклопедии [3].

2.2 Алгоритм построения семантических связей

Семантическая модель предметной области представлена в виде взвешенного ориентированного графа $G=(V, U)$, где вершины V соответствуют терминам домена, а дуги U представляют семантические отношения между ними. При построении семантического графа G необходимо выделить из текста энциклопедии термины данной научной области, а также семантические связи между ними.

Выделены 3 типа семантических связей между вершинами графа:

1. *связь-определение* устанавливается между базовым термином из заголовка статьи энциклопедии и всеми n-граммами из определения данного термина;
2. *родовидовая связь* устанавливается для термина, представляющего собой n-грамму длиной более одного слова, и всеми его подмножествами;
3. *ассоциативная связь* устанавливается от базового термина ко всем n-граммам из текста статьи данного термина.

Как показал дальнейший опыт применения построенного семантического графа для анализа тематической принадлежности научной статьи, наибольший вклад в расчет семантической близости как статей в целом, так и их отдельных фрагментов вносит ассоциативная связь, а минимальный – связь определения. Поэтому главной проблемой при построении семантического графа является вычисление весов дуг на основе функции значимости одного термина для другого. Для определения значимости семантических связей между терминами используются статистические подходы компьютерной лингвистики, основанные на данных о совместной встречаемости слов [4, 5].

Введем следующие обозначения:
w_i – отдельное слово из множества слов энциклопедии;
$X=\{w_1,w_2,...w_n\}$ – n-грамма из слов множества тематической энциклопедии;
$l(X)$ – количество слов в n-грамме X;
T – базовый термин предметной области, соответствует заголовку статьи энциклопедии;
O_T – часть статьи энциклопедии, содержащая определение термина T;
$Z(X, T)$ – значимость n-граммы X для термина T;
$TF(w_i, T)$ – частота слова w_i в статье энциклопедии, описывающей термин T;
$IDF(w_i)$ – инверсия частоты, с которой слово w_i встречается в статьях энциклопедии [6].

Вес связи между базовым термином T и n-граммой X, выделенной из текста соответствующей данному термину статьи энциклопедии, является некоторой функцией, зависящей от частотности отдельных слов n-граммы. Для различных типов семантических связей используются различные функции вычисления весов дуг графа. Вес связи-определения оценивается следующим образом:

$$Z(X,T) = \frac{l(X)}{l(O_T)}. \quad (1)$$

Вес родовидовой связи определяется как

$$Z(X,T) = \frac{l(X)}{l(T)} \quad (2)$$

Последующие эксперименты были проведены с использованием трех функций вычисления весов дуг, соответствующих ассоциативным связям семантического графа.

1. **Оценка по сумме вкладов.** Предполагая, что вес многословных терминов выше, чем вес термина, представляющего собой одно слово, получим

$$Z(X,T) = \sum_{i=1}^{n} TF(w_i,T) \cdot IDF(w_i). \quad (3)$$

2. **Вероятностная оценка.** Значимость n-граммы X для термина T оценивается как вероятность появления X в статье энциклопедии, описывающей термин T.

$$Z(X,T) = \prod_{i=1}^{n} TF(w_i,T) \cdot \sum_{i=1}^{n} IDF(w_i). \quad (4)$$

3. **Оценка по вкладу в содержание.** Значимость *n*-граммы *X* определяется как количество информации, которое несет *X* для статьи термина *T*.

$$Z(X,T) = \log_2 N \cdot \sum_{i=1}^{n} TF(w_i,T) \cdot IDF(w_i). \quad (5)$$

После расчета весов всех дуг графа проводится нормализация их значений в диапазоне [0; 1].

Для количественной оценки качества связей вершины T введем некоторую числовую характеристику $h(T)$, вычисляемую как

$$h(T) = \frac{\sum_{i=1}^{k} Z(X_i,T)}{k \cdot \log_2(k+1)}, \quad (6)$$

где X_i – вершины графа, являющиеся непосредственными соседями вершины T,
k – количество соседних вершин для вершины T.

2.3 Результаты экспериментов

При извлечении терминов-кандидатов в семантический граф были добавлены только те *n*-граммы, которые при проверке синтаксическим анализатором [7] распознаны как корректные: около 2% выделенных из текста энциклопедии *n*-грамм были отброшены как синтаксически некорректные. В процессе эксперимента было выделено:

- 6211 базовых терминов, длина которых составляет от одного до 6 отдельных слов;
- 1105341 терминов-кандидатов;
- 2267949 семантические связи между терминами, из них 2137304 связей-ассоциаций, 116778 связей-определений, 13867 родовидовых связей.

Был произведен анализ предложенных вариантов расчета: для каждого значения веса связи определено количество связей такого веса. Результаты представлены на рис. 1.

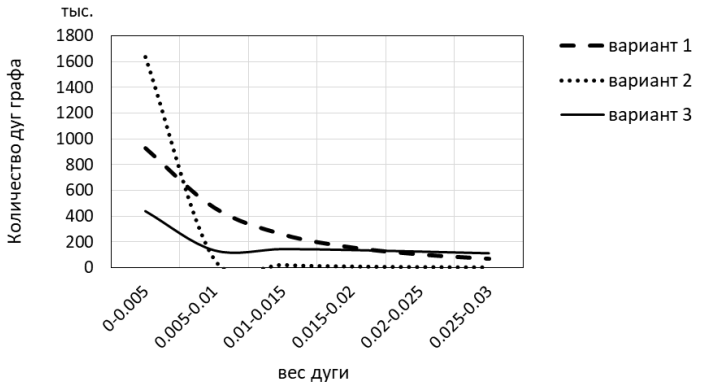


Рис.1. Распределение количества связей по весу при различных вариантах расчета веса дуги графа

Построенный семантический граф предметной области планируется использовать для анализа научных статей, с целью сокращения времени обработки дуги графа, вес которых менее некоторого фиксированного порога, должны быть отброшены как незначащие. После удаления дуг графа вершины, не имеющие входящих и исходящих дуг, также должны быть отброшены.

В ходе экспериментов выбрано значение фиксированного порога для отсечения 0.005 и вариант расчета 1 веса дуги. Полученный граф после удаления незначащих связей имеет следующую структуру:

- 817523 термина, базовые термины и термины-кандидаты (удалено 26% начального количества вершин графа);
- 1338269 семантических связей между терминами (удален 41% первоначального количества дуг графа).

В полученном графе набор связей каждой вершины был оценен с помощью предложенной характеристики качества $h(T)$. Выбор варианта расчета $Z(X, T)$ позволяет управлять структурой графа: для дальнейшего использования оптимально сохранить вершины с небольшим количество связей, но с высокими значениями весов. Граф был проанализирован с точки зрения распределения количества связей по значению характеристики качества $h(T)$ при различных вариантах расчета $Z(X, T)$. Соответствующие гистограммы распределения представлены на рис. 2.

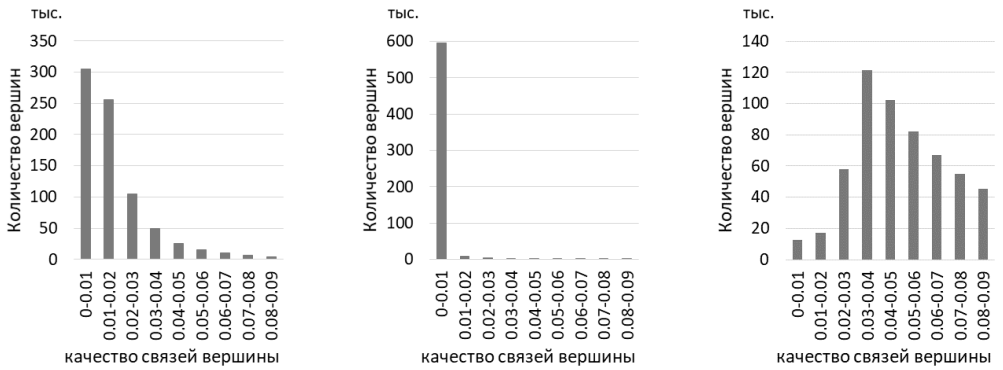


Рис. 2. Распределение количества вершин по значению характеристики качества h при различных вариантах расчета веса дуги графа

Предполагая, что значимость n-грамм длиной более одного слова выше, чем значимость отдельных слов, семантические связи с составными терминами являются более важными. Вариант расчета 3 повышает значение веса таких связей, что позволяет повысить качество дальнейшей обработки научных текстов с использованием построенной семантической модели.

3 Заключение

В качестве модели представления научных знаний предметной области предлагается взвешенный ориентированный граф, вершинами которого являются термины данной предметной области, связанные семантическим отношениями различного типа. Источником обучающих данных является тематическая энциклопедия соответствующей научной области. Предложены различные варианты расчета веса семантических связей между терминами домена, предложен вариант количественной оценки качества связей графа после удаления незначащих связей. Проведен анализ результатов экспериментов, структура полученного графа оценена с помощью предложенной количественной характеристики качества связей вершин.

Литература

[1] Korney A., Kryuchkova E., Savchenko V. (2020) Information Retrieval Approach Using Semiotic Models Based on Multi-layered Semantic Graphs // In: Jordan V., Filimonov N., Tarasov I., Faerman V. (eds) High-Performance Computing Systems and Technologies in Scientific Research, Automation of Control and Production. HPCST 2020. Communications in Computer and Information Science. Springer, Cham. - 2020. - Vol. 1304. - P. 162-177.

[2] Бачище О. И., Крючкова Е. Н., Шушаков Д. С. Проблемы автоматической обработки научных текстов на основе извлечения информации из энциклопедий соответствующих

доменных областей // Программная инженерия. 2023. Том 14, № 1. С. 42-50. DOI: 10.7587/prin.14.42-50.
[3] Виноградов И.М. (ред.) Математическая энциклопедия в 5 томах. М.: Сов. Энциклопедия, 1977.
[4] Yuan Y., Gao J., Zhang Y. Supervised learning for robust term extraction // International Conference on Asian Language Processing. 2017. URL: https://www.researchgate.net/publication/321748439_Supervised_Learning_for_Robust_Term_Extraction
[5] Hossari M., Dev S., Kelleher J. D. TEST: A Terminology Extraction System for Technology Related Terms. In: Proceedings of the 2019 11th International Conference on Computer and Automation Engineering, 2019, p. 78-81.
[6] Qaiser S., Ramsha A. Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents // International Journal of Computer Applications. 2018. Vol. 181. No. 1. P. 25-29.
[7] Natasha. Tools for Russian NLP: segmentation, embeddings, morphology, lemmatization, syntax, NER, fact extraction. URL: https://github.com/natasha/natasha (дата обращения 14.03.2023)

Динамика изложения информации как ассоциативный идентификатор научного текста

Вопилова Е.В., Крючкова Е.Н.

Алтайский государственный технический университет им. И.И. Ползунова, пр. Ленина, д. 46, г. Барнаул, 656038, Россия.
Polzunov Altai State Technical University, Lenina ave., 46, Barnaul, 656038, Russia

vopilova.elena@gmail.com, kruchkova_elena@mail.ru

Аннотация. *В работе предлагается метод определения динамики аспектного содержания научных текстов. Обсуждаются проблемы, возникающие при автоматической обработке научных текстов, приведены результаты работы по созданию комбинированного метода аспектно-ориентированного анализа научных текстов в области математики. В качестве источника специальных терминов использована тематическая энциклопедия. Предложена структура динамического шаблона изложения материала, построенного на основе извлеченных множеств профессиональных терминов, предлагается алгоритм формирования семантических зависимостей между фрагментами текста, приводятся примеры полученных в результате экспериментов статистических зависимостей.*

Ключевые слова: аспектно-ориентированный анализ, научный лексикон, семантический граф, классификация научных текстов, автоматическая обработка неструктурированных текстов

1 Введение

Анализ научных публикаций имеет свои особенности, связанные не только с составом научной лексики в конкретной прикладной области [1, 2] и спецификой значений некоторых слов и выражений, но и другими свойствами, которые необходимо учитывать при автоматической обработке текста [4]. Текст научной публикации имеет, как правило, сложную синтаксическую и семантическую структуру, соответствующую общепринятому порядку изложения материала, начиная от аннотации и заканчивая итоговыми выводами по сделанной работе. При этом научный текст – это не бессвязный набор тематической информации, а логически связанная последовательность положений, данных, фактов, выводов. Отдельные текстовые блоки публикации могут относиться к разным разделам и иметь неравномерный вклад в общую семантику работы. При этом важна не только общая семантика текста, но, в первую очередь, логика перехода от одного фрагмента к другому, определяющая базовый смысл научного текста.

Ключевой смысл текстового фрагмента в некотором приближении можно считать зависимым от списка профессиональных терминов, содержащихся в этом фрагменте. Обработка научной терминологии полностью определяется научной областью и имеет особую сложность, обусловленную как сложностью самой терминологии, так и отсутствием полномасштабных корпусов терминов. И если для англоязычной терминологии имеются некоторые корпуса научных терминов [4, 5], то для русского языка такие наборы практически отсутствуют. По этой причине для создания базового набора терминов научной области предлагается использовать научные энциклопедии [6], на основе которых формируется базовый семантический граф предметной области. В данной работе тематический анализ научных текстов проведен с использованием терминологии, автоматически извлеченной из пятитомной математической энциклопедии [7]. Следует отметить, что полученный список терминов достаточно хорошо

покрывает область математической терминологии. Попытка использовать в качестве частично структурированных текстов, используемых для пополнения базового терминологического набора, тексты статей из научных журналов не привела к существенному пополнению списка стандартных *n*-грамм и тем более к модификации структуры семантического графа. Видимо, существенное влияние на конструкцию графа может оказать только целенаправленный подбор текстов по новым разделам науки.

Все приведенные в данной работе эксперименты проведены на основе семантической информации, извлеченной исключительно из математической энциклопедии.

2 Аспектный анализ научного текста

2.1 Базовые термины и устойчивые словосочетания

Проведенный в рамках данной работы статистический анализ научных текстов показал, что большая часть таких текстов состоит из *n*-грамм двух типов.

- Специальная терминология научной области с весьма ограниченным набором уникальных словоформ. В качестве таких терминов в данной работе мы рассматриваем только такие, которые либо получены из списка ключевых слов научных публикаций, либо являются заголовками статей энциклопедии.
- Устойчивые словосочетания, уникальные как для научной терминологии в целом, так и специфичные для конкретных научных разделов. Некоторые из таких *n*-грамм, состоящих из слов общеупотребительной лексики, имеют смысл только в научном тексте и даже являются специальными терминами, не представленными в энциклопедии в виде отдельной статьи (например, «собственный идеал кольца», «поле частных кольца», «пучок колец» и т.п.).

Таким образом, в качестве множества вершин V семантического графа $G_{domain} = (V, U)$ будем рассматривать как термины, извлеченные из наименования статей энциклопедии и списка ключевых слов публикаций, так и устойчивые словосочетания (*n*-граммы). На программном уровне эти объекты представлены неупорядоченными множествами, содержащими нормализованные словоформы входящих в *n*-грамму слов. Полученный после отсечения малозначимых словосочетаний состав представлен на рисунке 1.

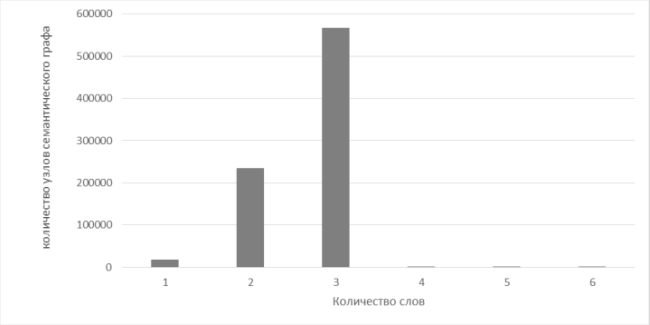


Рис. 1. Состав множества математических терминов и устойчивых словосочетаний по количеству слов

Семантические графы хорошо представляют зависимости между терминами научных текстов. Например, в [8] представлен корпус, сформированный из аннотаций научных статей, в которых были размечены понятия и семантические отношения между ними. Обработка энциклопедии не требует предварительной разметки, поскольку наименования статей уже служат в качестве частичной разметки. Однако установить отношения между выделенными сущностями только на основе заголовков статей энциклопедии невозможно. Устойчивые сочетания слов, обороты и идиомы, которые чаще всего используются в домене, могут служить основой для метода выявления сильной семантической связи между соответствующими словами. При этом важно не только определить тип семантической связи [9], но и вычислить ее значимость. Вершины графа G_{domain} связаны ориентированными дугами, веса которых соответствуют значимости одного объекта для другого. Расчет весовых коэффициентов – это отдельная задача и здесь не рассматривается. В данной работе важным является лишь то, что дуги множества U

семантического графа имеют вес, соответствующий значимости связи между вершинами. Веса связей нормированы и находятся в диапазоне [0;1].

2.2 Первичный семантический анализ текста научной публикации

Задача семантического анализа – выявить тематику анализируемого текста. Зная частоту вхождения *n*-грамм в текст, можно выявить некоторую информацию не только о лексических, но и семантических особенностях текстов. Любой научный текст относится к определенной области и, следовательно, его семантика должна быть основана на знаниях именно этой области. Это означает, что семантический граф данного текста является подграфом семантического графа G_{domain} всего домена. И чем больше в тексте публикации используются конкретная терминология, тем больше оснований считать, что публикация затрагивает именно данные конкретные тематические моменты.

В тексте публикации термины не всегда явно используются, зачастую составные термины представлены в усеченной формулировке. Но при чтении они могут появляться в нашем сознании в результате установившихся там ассоциативных связей в соответствии с предметной областью. Ключевым принципом является повторяемость сигнальных данных, возможно различающихся по наименованию, но принадлежащих одной области.

Для эффективного проведения аспектного анализа необходимо рассмотреть проблему объединения в едином семантическом ядре как общей и полной информации о рассматриваемой науке, так и узкоспециализированных знаний, связанных с темой публикации. Будем использовать частотность появления отдельных элементов множества V для выделения кластеров использованных предметных терминов. Веса дуг базового графа G_{domain} представляют ассоциативные зависимости между значимыми словосочетаниями предметной области, поэтому появление некоторого термина $t \in V$ в тексте должно повлиять на наше представление о возможном использовании терминов из некоторой окрестности термина t.

Предварительно установим веса всех вершин графа G_{domain} в нулевые значения. Алгоритм распространения весов вершин предназначен для выявления значимой терминологии в тексте с учетом затухания по расстоянию от исходной вершины. Пусть Δs – коэффициент пространственного затухания, $H(t)$ – частотность появления термина t в анализируемом тексте, тогда связанная с t вершина t_i в результате работы алгоритма распространения весов получит добавочный вес

$$H(t_i) = p(t, t_i) \cdot \Delta s \cdot l(t, t_i), \qquad (1)$$

где $l(t, t_i)$ – число дуг на этом пути,

$p(t, t_i)$ – произведение весов дуг на кратчайшем пути от t к t_i.

Использование $p(t, t_i)$ в (1) отмечает тот несомненный факт, что чем меньше значимость связи между терминами, тем меньшее влияние должна оказывать частотность $H(t)$. И наоборот, при высоком уровне веса связи между вершинами частотность $H(t)$ оказывает большее влияние.

После обработки всего текста публикации из графа G_{domain} получим его подграф G_{text}, вершинам которого присвоены веса прямой или распространенной частотности. Выделим кластеры предметных терминов в графе G_{text}. Коэффициент затухания Δs управляет размерами контекста, на который распространяется влияние частотности использованного термина, и при его уменьшении окрестность выделенных из текста статьи терминов в семантическом графе резко сжимается.

В построенном графе G_{text} выделены вершины с высоким весом. Именно такие вершины являются претендентами на центры кластеров. Будем выбирать центры кластеров как вершины с превышающим некоторый порог значением функции $f_focus(v)$, где $v \in V_{text}$. В зависимости от вида функции $f_focus(v)$ центрами кластеров могут стать как вершины с максимальным значением весов (при выборе функции $f_focus(v) = H(v)$, где $H(v)$ – вес вершины v), так и при выборе соответствующих функций – центры сгустков вершин с высоким весом. Например, в качестве функции $f_focus(v)$, можно использовать сумму $H(v_i)$ непосредственных соседей вершины v. Попавшие в топ по значению функции f_focus (v) вершины будем считать первоначальными центрами кластеров, а множество окружающих их вершин на расстоянии не более заданного радиуса кластера – аспектными терминами этого кластера.

Для последующей кластеризации терминов графа использован алгоритм k-medoids, являющийся модификацией классического алгоритма кластеризации k-means. Данный алгоритм

предназначен для решения задач, когда в качестве центра кластера может выступать не любая точка, а только точка из набора обозначенных, что делает этот алгоритм пригодным для использования на графах [10].

Эксперименты показали хорошее качество построения весов при использовании формулы (1) вычисления добавочного веса. Результаты проведенного анализа статьи «Островной генетический алгоритм с динамическим распределением вероятностей выбора генетических операторов» [11] приведены в таблице 1.

Таблица 1.

№	Центр кластера	Вес термина	Суммарный вес кластера
1	Приближение функции	0,007	334
2	Вариационное исчисление	0,022	617
3	Аналитическая функция	0,036	8
4	Обыкновенное дифференциальное уравнение	0,151	288
5	Теория вероятности	1,012	533
6	Топологическое пространство	0,222	160
7	Спектральная теория	0,001	324
8	Аналитическая теория чисел	0,000	179

2.3 Анализ динамики изложения материала

В построенном графе образовались сгустки вершин с высоким весом, рассматриваемые нами как претенденты на центры семантических кластеров, соответствующие обсуждаемым в публикации тематическим категориям. Процесс чтения текста человеком происходит на протяжении некоторого временного интервала, очередной входной фрагмент переводит понимание содержимого текста на некоторый более высокий уровень. По мере чтения текста происходит комбинация осознанного смысла, извлеченного из уже прочитанных фрагментов с новыми данными. Промежуточные знания модифицируются и постепенно затухают во времени, причем обработка новых знаний производится более приоритетно. Это означает, что со временем происходит затухание накопленной информации, которая в тексте больше не упоминается.

Пусть Δt – коэффициент темпорального затухания. Будем теперь наряду с весом, вычисляемым по формуле (1), на каждом шаге использовать темпоральное затухание в каждой вершине:

$$Z(t_i) = H(t_i) \cdot \Delta t, \qquad (2)$$

реализуя тем самым уменьшение значимости давно обработанной информации.

Выделенное на шаге предварительного семантического анализа множество кластеров $\{K_1, K_2, \ldots, K_k\}$ с центрами $\{C_1, C_2, \ldots, C_k\}$ представляют совокупность содержащихся в тексте аспектов. Вычисляя вес каждого кластера с использованием темпоральных весов (2), можно оценить текущий вес каждого из K_i. Удалая из них малозначимые мы динамически получаем список контекстно-зависимых аспектов. На рисунке 2 приведен пример, иллюстрирующий динамику обработки статьи «Островной генетический алгоритм с динамическим распределением вероятностей выбора генетических операторов», выделенные аспекты перечислены в таблице 1. Анализ был проведен по тексту публикации, который был разделен на 6 фрагментов, нумерация кластеров соответствует нумерации в таблице 1.

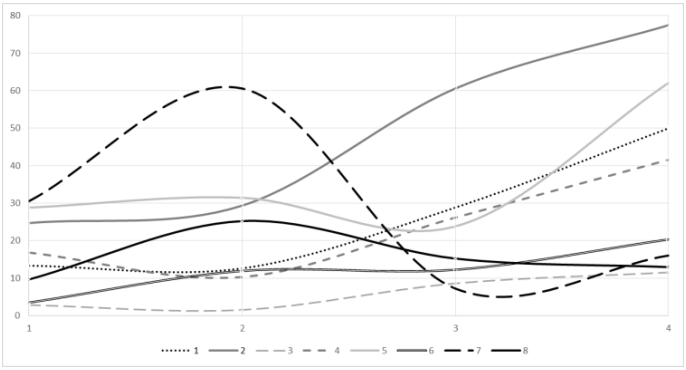


Рис.2. Динамика аспектного анализа публикации

Схема динамики изложения может быть представлена в виде бинарного дерева, левые потомки представляют элементы одного фрагмента текста, упорядоченные по убыванию весов кластеров. Правый потомок существует только у первого элемента каждой цепочки и указывает на первый элемент цепочки следующего фрагмента. Фрагмент бинарного дерева публикации представлен на рисунке 3.

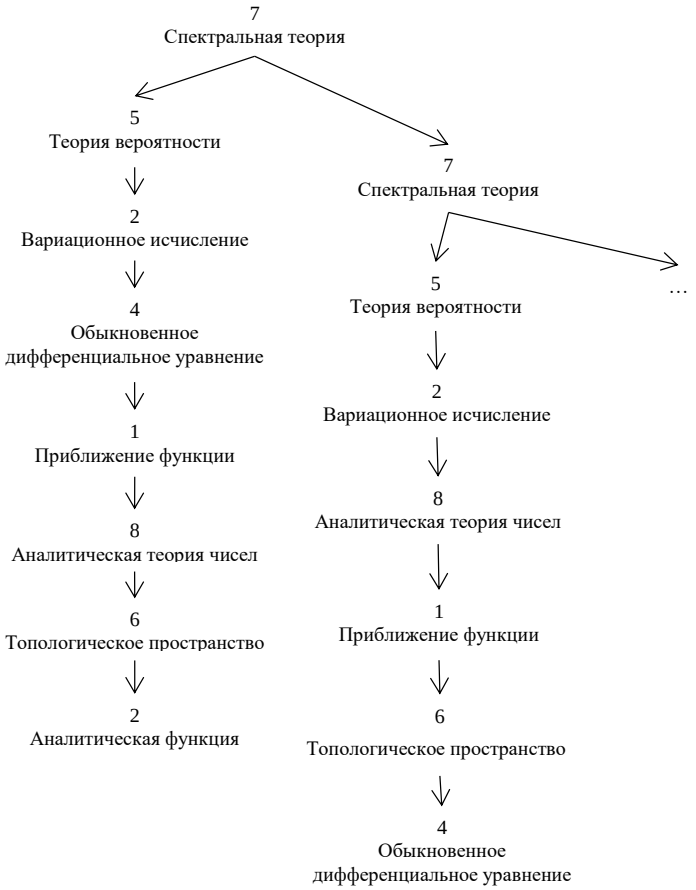


Рис.3. Бинарное дерево аспектного анализа публикации

Такое дерево можно изобразить в линейном виде и представить строкой, в которой последовательно перечислены цепочки соседей, а каждая цепочка заканчивается специальным маркером. Например, фрагменту бинарного дерева на рисунке 3 соответствует строка номеров тем:

7;5;2;4;1;8;6;2#7;5;2;8;1;6;4#...

Такая строка может служить ассоциативным идентификатором текста научной публикации. Полученные идентификаторы можно сравнивать, реализуя тем самым расширенные способы сравнения публикаций как по тематике, так и по способу изложения материала. Следует отметить, что предложенная структура ассоциативного идентификатора текста позволяет использовать различные алгоритмы обработки бинарных деревьев.

3 Заключение

Предложенный метод динамического анализа научных текстов, основанный на кластеризации семантических графов, позволяет определить последовательность изложения материала. Построенная в процессе схема динамической смены аспектов может служить виртуальным идентификатором публикации и использоваться для сравнения публикаций.

Литература

[1] Bruches E.P., Pauls A.E., Batura T.V., Isachenko V.V. Study of Methods for Entity Recognition and Relation Extraction in Scientific Texts. Science and Artificial Intelligence conference (SAIC-2020). 2020. p. 41-45. DOI: 10.1109/S.A.I.ence50533.2020.9303196 http://alt.qcri.org/semeval2016/task5/

[2] Sysoev A., Andrianov I. Named Entity Recognition in Russian: the Power of Wiki-Based Approach. Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference Dialogue, pp. 746-755, 2016.

[3] Агеев М.С., Добров Б.В., Лукашевич Н.В. Автоматическая рубрикация текстов: методы и проблемы // Ученые записки Казанского государственного университета. Серия Физико-математические науки. 2008. Том 150. книга 4. С. 25-40.

[4] QasemiZadeh B., Schumann A. The ACL RD-TEC 2.0: A Language Resource for Evaluating Term Extraction and Entity Recognition Methods. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). European Language Resources Association (ELRA), Portoroz, Slovenia, pp.1862-1868. 2016.

[5] Rosario B., Hearst M. Classifying Semantic Relations in Bioscience Texts. Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04). Barcelona, Spain, pp. 430-473. 2004. DOI: 10.3115/1218955.1219010.

[6] Бачище О. И., Крючкова Е. Н., Шушаков Д. С. Проблемы автоматической обработки научных текстов на основе извлечения информации из энциклопедий соответствующих доменных областей // Программная инженерия. 2023. Том 14, № 1. С. 42-50. DOI: 10.7587/prin.14.42-50.

[7] [3]Виноградов И.М. (ред.) Математическая энциклопедия в 5 томах. М.: Сов. Энциклопедия, 1977.

[8] Gabor K., Buscaldi D., Schumann A., QasemiZadeh B., Zargayouna H., Charnois T.. SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers. Proceedings of The 12th International Workshop on Semantic Evaluation. Association for Computational Linguistics, New Orleans, Louisiana, pp. 679-688. 2018. DOI: 10.18653/v1/S18-1111.

[9] Luan Y., He L., Ostendorf M., Hajishirzi H. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. Association for Computational Linguistics, Brussels, Belgium, pp. 3219-3232. 2018. DOI: 10.18653/v1/D18-1360.

[10]Дмитриев И. Н. Быстрый алгоритм кластерного анализа k-medoids. ПДМ, 2018, № 39, С.116-127. DOI: https://doi.org/10.17223/20710410/39/11

[11]Мясников А.С. Островной генетический алгоритм с динамическим распределением вероятностей выбора генетических операторов // Наука и образование: научное издание мгту им. Н.Э. Баумана, 2010.

Методы автоматизации разработки смарт-контрактов при помощи моделей ситуаций

Галиева А.Г.[1], Пальчунов Д.Е.[2]

[1] *Институт Математики им. С.Л. Соболева СО РАН, пр. Академика Коптюга, д. 4, г. Новосибирск, 630090, Россия*
[2] *Новосибирский Государственный Университет, ул. Пирогова, д. 1, г. Новосибирск, 630090, Россия*

a.galieva@g.nsu.ru, palch@math.nsc.ru

Аннотация. *В статье проведён анализ существующих подходов к автоматизации бизнес-процессов. Методы, предложенные в данной работе, позволяют элиминировать непосредственное участие разработчиков в создании и управлении смарт-контрактами. Описан методологический подход к автоматизации разработки смарт-контрактов с использованием моделей ситуаций. Представлена схема работы программной системы автоматизации бизнес-процессов при помощи конструирования моделей ситуаций. Описано использование онтологически гомоморфных моделей для создания шаблонов смарт-контрактов. Предложены методы генерации смарт-контрактов на языке sDSL на основе использования частичных моделей. Приведены примеры трансляции представления бизнес-процессов через ситуационные модели в исполняемые смарт-контракты на языке sDSL. Описано взаимодействие нескольких смарт-контрактов между собой. Рассмотрены перспективы интеграции нейронных сетей при оценке рисков бизнес-процессов.*

Ключевые слова: смарт-контракт, модель ситуации, автоматизация бизнес-процессов, семантическое моделирование, частичная модель, онтологический гомоморфизм, аудит смарт-контрактов

1 Введение

Современные бизнес-процессы и модели работы как отдельных предприятий, так и целых отраслей, все больше усложняются, расширяются и объединяются. Это делает их автоматизацию более сложной. Существующие подходы к моделированию ситуаций, возникающих в различных бизнес-процессах, ограничены замкнутостью описываемых систем: не допускается недетерминированность сигналов, поступающих извне, в описываемый бизнес-процесс.

Однако, для управления рисками необходимо обеспечить моделирование недетерминированных бизнес-процессов. Для этого предлагается использование ветвящихся смарт-контрактов, при этом их ветвление на базовом уровне не будет относиться к моделированию рисков, а будет лишь моделировать их поведение в зависимости от определенной ситуации. С другой стороны, ветвление на метауровне будет относиться к обработке рисков, поэтому необходимо обеспечить корректное исполнение смарт-контрактов при возникновении ситуаций, в которых возможно различное поведение отдельных участников бизнес-процесса.

2 Автоматизация создания смарт-контрактов при помощи моделей ситуаций

Проблема аудита смарт-контрактов, который на сегодняшний день остается довольно дорогой, но необходимой процедурой для автоматизации бизнес-процессов с их использованием, актуальна, что подчеркивается во многих современных публикациях и источниках [1-5].

Существует множество методов формализации описания бизнес-процессов, основанных на семантическом моделировании [6-8]. Для автоматизации бизнес-процессов, формализованных таким способом, необходимо решить проблему упрощения аудита смарт-контрактов с помощью частичного исключения участия специалистов в области их разработки. Эту задачу можно решить, предоставив прямое управление требованиями и условиями смарт-контракта экспертам предметной области для соответствующих бизнес-процессов.

Для формализации бизнес-процессов предлагается использование моделей ситуаций, которые организованы в ориентированный граф, в вершинах которого расположены все возможные в данном бизнес-процессе ситуации (состояния), а ребрами являются условия перехода в эти состояния. В результате переходов между состояниями, в зависимости от поступающих в модель сигналов управления, будет выделяться актуальная цепочка ситуаций, формализованных частичными моделями [9]. Таким образом, при исполнении смарт-контракта будет выполняться та или иная цепочка ситуаций, в зависимости от дополнительных параметров, поступающих в систему извне и описывающих последовательность недетерминированных событий бизнес-процесса.

Для автоматизации вывода данных цепочек исполнения предлагается использование аргументации на основе прецедентов [10-14], а также частичных моделей [12, 15] возможных ситуаций в данном бизнес-процессе. При формальном описании бизнес-процесса для его последующей автоматизации будет построен граф, в вершинах которого будут находиться частичные модели, описывающие текущее состояние системы, а также множество выводимых предложений (непротиворечивых условий перехода в следующие состояния).

Для моделирования смарт-контрактов с использованием моделей ситуаций необходимо реализовать инструмент, способный на основе описаний частичных моделей строить все теоретически возможные цепочки исполнения смарт-контракта. Таким образом смарт-контракт будет симулировать путь в графе возможных переходов между частичными моделями.

Такое моделирование бизнес-процесса позволит также использовать онтологические гомоморфизмы частичных моделей при построении цепочек исполнения смарт-контрактов, благодаря чему станет возможным переиспользование уже существующих смарт-контрактов в качестве шаблонов для подобных подпроцессов. Например, предикат передать_кабель(Вася, Петя, USB) будет онтологически подобным предикату передать_кабель (Вася, Петя) другого бизнес-процесса, в котором уже будет не важно, какой именно кабель передается. Также онтологический гомоморфизм позволит в этот же предикат передавать другие предикаты: так, если определен унарный предикат usb(Тип), где в качестве параметра передается тип USB-кабеля (type-c, type-b), мы можем создать вложенность, передав данный предикат в качестве третьего параметра: передать_кабель(Вася, Петя, usb(type-c)). Замена онтологически подобных предикатов происходит в зависимости от конкретной ситуации. Таким образом, переиспользование заменяемых шаблонов при построении смарт-контрактов позволяет автоматизировать процесс их создания.

Можно выделить три вида онтологических гомоморфизмов:

1) Изменение номенклатуры (объектов)

2) Изменение отношений

a. одноместные: болонка => такса

b. двуместные: покупка ⇔ аренда c. многоместные

3) Предикаты-действия: банковский перевод ⇔ с карты на карту ⇔ быстрый платеж

Чем более разнообразная и богатая онтология, тем больше возможностей онтологических гомоморфизмов.

Чтобы проиллюстрировать моделирование на основе моделей ситуаций, рассмотрим пример со смарт-контрактом поставки товара (рис. 1)

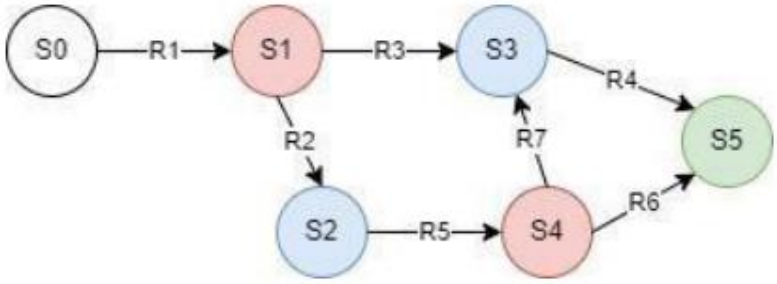


Рис.1. Модель ситуаций бизнес-процесса поставки товара.

Ситуации (состояния): S0 = товар на складе; S1 = ожидание отправки товара; S2 = товар отправлен; S3 = ожидание возврата средств заказчику; S4 = товар получен заказчиком; S5 = конец.

Сигналы управления (события): R1 = внесена оплата; R2 = произошла отправка товара; R3 = время ожидания отправки товара истекло; R4 = оплата возвращена заказчику; R5 = подтверждено получение товара; R6 = получен товар соответствующего вида/качества; R7 = получен товар не соответствующего вида/качества.

Ситуации S1 и S4 (помечены красным) в зависимости от приходящих сигналов могут переходить в разные ситуации, поэтому являются недетерминированными, а значит смарт-контракт, имплементирующий данную модель, будет учитывать вышеописанные риски. То есть в данном случае описываемая модель является частичной, так как не все переходы в ней детерминированы. Также в модели могут быть не определены некоторые данные (например – какой товар поставляется), которые в других бизнес-процессах необходимо указывать при описании подобных ситуаций. Это позволяет не только избавиться от переполнения описательных моделей ненужными для принятия стратегических решений данными, но и добавляет гибкость подстановки неконкретизированных данных, учитывая требования текущего бизнес-процесса. Также можно отметить, что все взаимодействия между ситуациями локальны: на ряду с валидацией и верификацией необходимо обеспечить и локализацию смарт-контракта.

3 Описание программной системы

Многие бизнес-процессы включают несколько подпроцессов, каждый из которых реализуется отдельным смарт-контрактом, что объясняется сложностью описания (с целью автоматизации) всего бизнес-процесса одним смарт-контрактом. Поэтому необходимо обеспечить взаимодействие смарт-контрактов общего бизнес-процесса между собой. Такие смарт-контракты могут быть запущены одновременно или же в разное время, при этом имея возможность отправлять друг другу сигналы о происшествии определенных событий и обрабатывать их в зависимости от условий. Таким образом, выполнение этапа одного смарт-контракта может служить внешним триггерным условием для второго (или нескольких других) смарт-контрактов.

В рамках данной работы решается задача автоматизации создания смарт-контрактов с использованием моделей ситуаций. Причем автоматизированные смарт-контракты могут выполняться как однократно, так и запускаться циклически, в зависимости от задач бизнес-процесса. Основным этапом реализации данной задачи является создание инструмента моделирования процесса исполнения смарт-контракта с использованием формального описания частичных моделей с последующей генерацией sDSL на основе описанных частичных моделей. В основе аппарата трансляции лежит представление частичной модели в виде набора атомарных диаграмм с последующей трансляцией в sDSL.

Разрабатываемая программная система создания и валидации смарт-контрактов включает в себя следующие компоненты:

- Программа на Java;
- Модель ситуаций;
- Генератор формализованного текста договора.

После получения информации из внешних источников о происшествии конкретного события, исполняемая программа обращается к ситуационной модели. Согласно описанным

правилам, эта модель запускает необходимые функции программы, таким образом производящие полезные действия. Генератор текста отвечает за создание формализованного текста контракта, поясняющего юристам суть договора. Схема функционирования программной системы представлена на рис. 2.

Данная программная система реализует итеративный процесс аудита контракта. Сначала упрощенный текст естественного языка, который представляет из себя текст, состоящий из простых предложений, которые не содержат референтных индексов, преобразуется в набор фрагментов атомарных диаграмм, по которым производится поиск подходящих шаблонов с последующим заполнением их необходимыми параметрами.

Рис.2. Схема функционирования программной системы.

Полученные таким образом шаблоны будут приближенными к требованиям смарт-контракта, и затем с каждой итерацией будут уточняться. По этим шаблонам система порождает текст естественного языка, который валидирует пользователь-эксперт предметной области на предмет корректной трансляции. Как только контракт будет полностью соответствовать ожиданиям пользователя, итеративный процесс валидации завершится, в противном случае пользователь вносит изменения во входные данные и запускается новая итерация. В результате такого редактирования шаблоны доводятся до полного соответствия требованиям контракта.

Согласно предложенному подходу, пользователь будет иметь возможность управлять бизнес-логикой контракта с помощью изменения только его спецификации, что позволяет ему без навыков программирования валидировать создаваемые контракты.

4 Перспективы интеграции нейронных сетей при оценке рисков

На любом из этапов разработки, внедрения и поддержки бизнес-процессов необходимо учитывать возможность возникновения различных рисков для их своевременного купирования. Реализованная система позволяет стейкхолдерам моделировать различные ситуации бизнес-процесса через построение семантической модели, тем самым предоставляя возможность учесть все риски еще на этапе планирования.

На сегодняшний день довольно популярным и прогрессивным является подход использования нейросетей для оценки и предотвращения подобных рисков. Нейронные сети обладают следующими достоинствами:

• Универсальность. Нейронные сети не зависят от свойств входных данных, для них не существует требования к определенному типу распределения исходных данных, либо требования к линейности целевых функций.

• Простота. Использование нейронных сетей не требует специальной подготовки, и для практического применения нет необходимости глубоко вникать во внутренние механизмы работы сети, в отличие от статистических методов, требующих фундаментальных знаний из области теории вероятностей и математической статистики.

• Не существует проблемы «проклятия размерности». Нейросети способны моделировать зависимости в случае большого числа переменных.

• Нейронные сети ускоряют процесс нахождения зависимостей за счет одновременной обработки данных всеми нейронами.

В то же время нейронные сети обладают рядом серьезных недостатков. Ключевыми из них являются следующие:

• Сложность построения архитектуры сети для конкретной задачи. Для подавляющего большинства реальных задач не разработано стандартных схем, в результате в каждом случае конструирование приходится начинать «с нуля»;

• Сложность интерпретации результатов обучения. Значения параметров элементов сети почти всегда невозможно объяснить в терминах решаемой задачи, в результате нейронная сеть остается «черным ящиком» не только для пользователей, но, отчасти, и для разработчиков;

Для использования такого подхода необходим большой объем тестовой выборки, работа с которой требует соответствующих ресурсов. Проблема решения больших задач может быть решена с помощью декомпозиции, при которой каждому свойству сопоставляется группа прецедентов из выборки, в результате чего получается четкая булевозначная модель.

5 Заключение

В данной работе были представлены разработанные методы частичной автоматизации создания, валидации и верификации смарт-контрактов при помощи моделей ситуаций. Проведен анализ существующих подходов к созданию смарт-контрактов, основанных на семантическом моделировании. Целью приведенных в работе методов и алгоритмов является предоставление возможности автоматизации верификации смарт-контрактов, а также значительное упрощение их валидации. Более того, описанные методы могут позволить конструировать смарт-контракты, описывающие намного более комплексные бизнес-процессы, приближенные к реальным бизнес-процессам крупных компаний.

В качестве продолжения разработки автоматизированных методов создания смарт-контрактов выступает реализация управления рисками бизнес-процессов с помощью построения вероятностной семантической модели рисков, которая будет обладать объяснительной функцией выведенных прогнозов, что отличает этот подход от использования нейронных сетей, которые строят вероятности без семантического подкрепления.

Литература

[1] J. De Andrés “On the impact of smart contracts on auditing” The International Journal of Digital Accounting Research, 2021, pp. 155-181

[2] N. Zemlyansky “Blockchain development: from zoo-stack to SDK in one language and with low-code support [Razrabotka na blokchejne: ot steka-zooparka — k SDK na odnom yazyke i s podderzhkoj low-code]” (in Russian), [electronic resource] https://tproger.ru/articles/razrabotka-na-blokchejne-ot-steka-zooparka-k-sdk-na-odnom-jazyke-i-s-podderzhkoj-low-code/

[3] A. Davies “How to perform a smart contract audit” [electronic resource] https://www.devteam.space/blog/how-to-audit-a-smart-contract-a-guide/

[4] Forklog guest materials, “Hacker “roasting”: experts talked about the inticacies of the audit using the Yggdrasil smart contract as an example [Hakerskaya «prozharka»: specialisty rasskazali o tonkostyah audita na primere smart-kontrakta Yggdrasil]” (in Russian) [electronic resource] https://forklog.com/hakerskaya-prozharka-spetsialisty-rasskazali-o-tonkostyah-audita-smart-kontraktov/

[5] R. R. Shaikhutdinov “The concept and problems of implementing smart contracts” XIV All-Russian Youth Scientific Conference Mavlyutov Readings Nov. 2-3, 2020, Ufa, Russia

[6] V. S. Gumirov, P. Y. Matyukov, D. E. Palchunov “Semantic Domain-specific Languages” 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Novosibirsk, Russia, 21-27 Oct. 2019. IEEE Press, 2019. P. 0955–0960.

[7] A. G. Galieva, D. E. Palchunov “Logical Methods for Smart Contract Development” 2019 International Multi-Conference on Engineering, Computer and Information Sciences SIBIRCON), Novosibirsk, Russia, IEEE Press, 2019. P. 0881-0885.

[8] F. Lautenbacher, B. Bauer, C. Seitz “Semantic Business Process Modeling – Benefits and Capability” AI Meets Business Rules and Process Management, Papers from the 2008 AAAI Spring Symposium, Technical Report SS-08-01, Stanford, California, USA, March 26-28, 2008

[9] D. Palchunov, A.S. Tregubov. Semantic methods of intelligent assistant developing. // In: 2021 International Symposium on Knowledge, Ontology, and Theory (KNOTH).

[10] D. Palchunov, G. Yakhyaeva, O. Yasinskaya. Software system for the diagnosis of the spine diseases using case-based reasoning. In: Proceedings of the International Conference on Biomedical Engineering and Computational Technologies (SIBIRCON / SibMedInfo — 2015), 28-30 October, 2015, Novosibirsk, p. 205-210

[11] Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard. Case-Based Reasoning Technology: From Foundations to Applications. — Springer, 2003

[12] D. E. Palchunov. Modeling Reasoning and Argumentation for the Development of Intelligent Assistants, 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Yekaterinburg, Russian Federation, 2022, pp. 820-825. doi: 10.1109/SIBIRCON56155.2022.10017050.

[13] Richter, Michael M.; Weber, Rosina O. (2013). Case-based reasoning: a textbook. Heidelberg: Springer-Verlag.

[14] D. Palchunov, A. Yakobson. Automated methods for conducting an argumented dialogue with the user, 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Yekaterinburg, Russian Federation, 2022, pp. 880-885. doi: 10.1109/SIBIRCON56155.2022.10016975.

[15] D. E. Palchunov. Methodological Aspects of the Application of Model Theory in Knowledge Engineering and Artificial Intelligence, 2022 Ural-Siberian Conference on Computational Technologies in Cognitive Science, Genomics and Biomedicine (CSGB), 2022, pp. 210-215. doi: 10.1109/CSGB56354.2022.9865602.

Лингвистические графы знаний для представления моделей тюркских языков

Гатиатуллин А.Р.[1], Прокопьев Н.А. [1], Сулейманов Д.Ш. [1]

[1]*Институт прикладной семиотики Академии наук Республики Татарстан*

ayrat.gatiatullin@tatar.ru, nikolai.prokopyev@gmail.com, dvdt.slt@gmail.com

Аннотация. *В статье представлено описание лингвистических графов знаний, которые составляют основу базы знаний портала «Тюркская морфема». Особенность данных графов знаний в том, что, онтологические модели, которые составляют основу данных графов знаний разработаны на основе прагматически-ориентированного подхода создания лингвистических моделей и ориентированы на наиболее полное описание структурно-функциональных особенностей тюркских языков. Данные графы знаний, являются полезным ресурсом, как для создания информационно-справочных систем по тюркским языкам, так и базой данных для создания прикладных программ для компьютерной обработки тюркских языков.* В составе портала, базу знаний которого образуют описанные графы знаний, имеются различные готовые программные модули для анализа текстов на тюркских языках.

Ключевые слова: граф знаний, лингвистический портал, лингвистическая единица

В настоящее время наблюдается бурное развитие интеллектуальных систем, работающих на основе технологий машинного обучения, которые для своей работы требуют большие объемы размеченных данных. А ситуация с ресурсами для тюркских языков такова, что они практически все относятся к малоресурсным языкам [1]. Особенность этих технологий в том, что они требуют большие объемы данных, и в случае нехватки данных для обеспечения качественной и эффективной работы разработчики ищут различные способы улучшения качества другими способами. Одним из таких подходов является комбинирование текстовых данных с графами знаний. Одно из наиболее удачных на наш взгляд определений графа знаний дано в работе [2], согласно которому граф знаний — это граф данных, предназначенный для накопления и передачи знаний о реальном мире, узлы которого представляют интересующие объекты, а ребра представляют отношения между этими объектами. Графы знаний также можно представить, как:

Графы знаний = онтологии + данные

В работе [3] описывается один из подходов, где графы знаний используются для увеличения объема обучающих выборок, состоящих из текстовых данных. А в работе [4] утверждается, что в основе графов знаний лежит парадигма, согласно которой сочетание структурированного и неструктурированного знания может принести пользу при выполнении целого ряда задач NLP. На рис. 1 приведена схема из работы [4], демонстрирующая использование графов знаний в NLP.

Разработчики систем для решения NLP задач с помощью графов знаний пытаются не только повышать эффективность работы этих систем, а также и объяснить полученные результаты [5], поскольку одним из основных недостатков технологии машинного обучения является сложность или даже невозможность объяснения получаемых результатов.

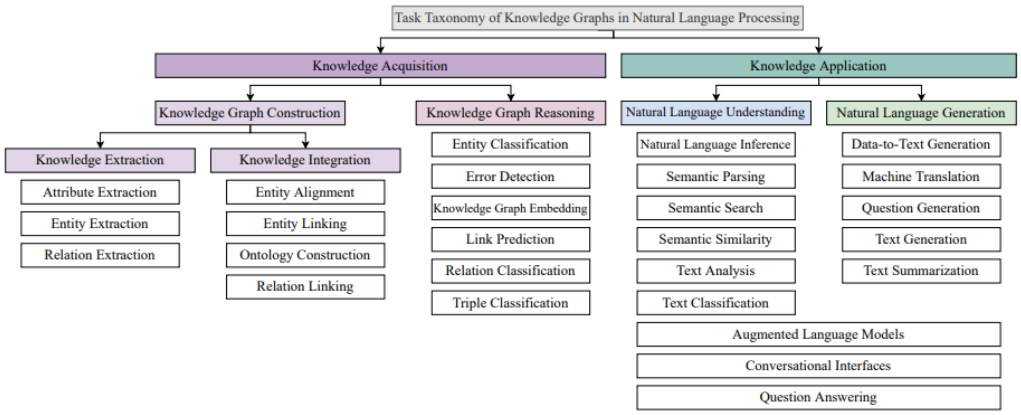


Рис.1. Использование графов знаний в NLP

Графы знаний являются достаточно эффективным инструментом для представления самой лингвистической информации. Есть целый ряд работ, которые посвящены разработке лингвистических графов знаний. Рассмотрим здесь пару работ, которые на наш взгляд являются наиболее интересными с точки зрения описания лингвистических данных. Один из таких лингвистических графов знаний описан в работе [6], модель которого представлена на рис.2.

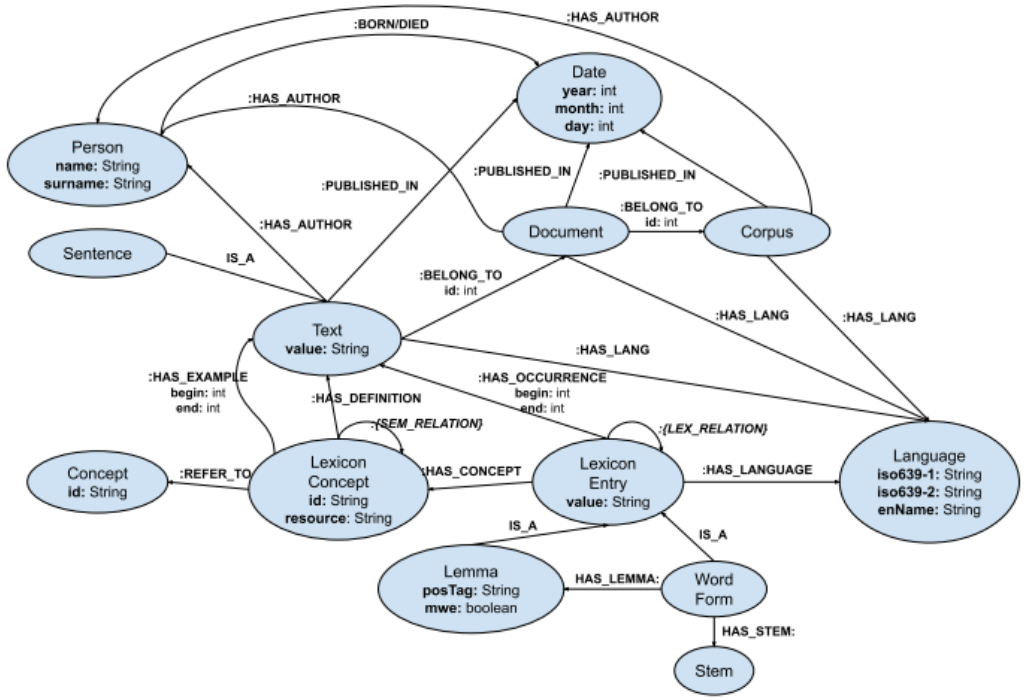


Рис.2. Модель лингвистического графа знаний

По мнению авторов данной работы, их граф знаний позволяет моделировать:

1) отношения между понятиями и словами;

2) информацию о встречаемости слов;

3) диахроническую информацию, о понятиях и словах.

Вершинами лингвистического графа знаний, представленного на рис.2 являются такие сущности, как Концепты (Concept), Лексические концепты (Lexicon Concept), Лексемы (Lexicon Entry) и другие. Лексические концепты графа взаимосвязаны между собой семантическими отношениями типа гипонимии и гиперонимии, что позволяет формировать тезаурусы понятий. Здесь также представлена и морфологическая информация, так Лексемы взаимосвязаны с леммой (Lemma) и основой словоформы (Stem). Особенность данного графа знаний в том, что

они не описывают ситуационную семантику, поэтому семантическое описание является достаточно ограниченным и отражает только таксономические отношения, аналогичные тем, что представлены в известном ресурсе WordNet. Для описания данной семантики подходят графы знаний фреймового типа, например такой достаточно известный ресурс, как FrameNet.

В другой работе описывается лингвистический граф знаний, который называется Lexicon Model for Ontologies (LeMOn) [7], где для описания значения лингвистических единиц уже используются семантические фреймы

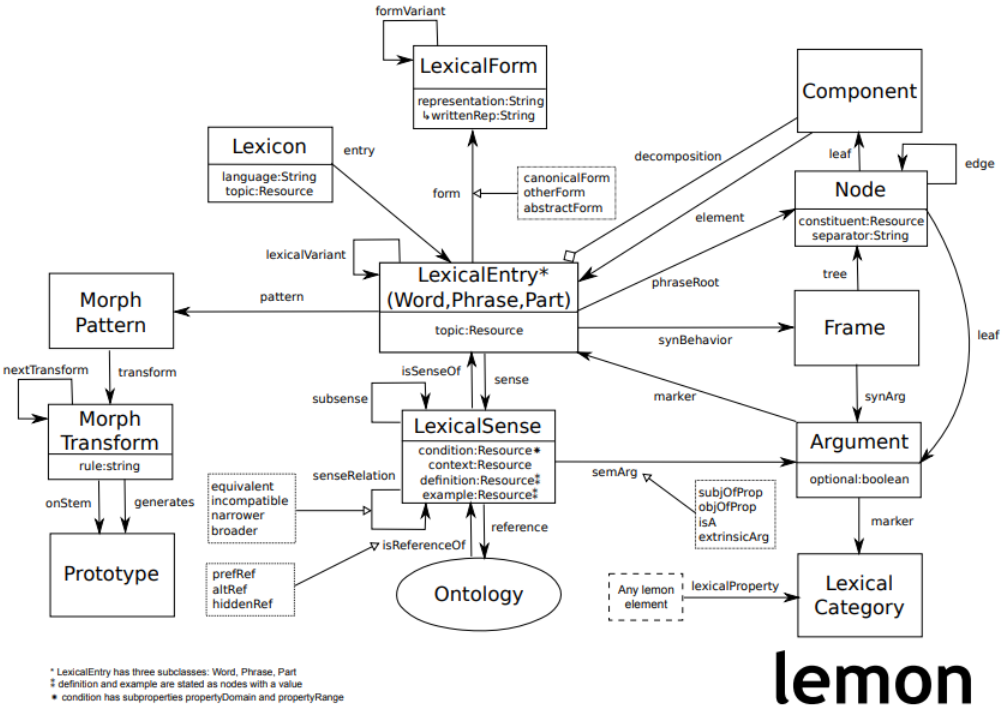


Рис.3. Модель лингвистического графа знаний Lemon

Рассмотрим сущности и отношения между ними, которые представлены на рис.3.

Лексикон (*Lexicon*) – это словарь, которые имеет такие атрибуты, как язык и тема.

Центральным элементом в данной модели является **Словарная статья** (*Lexical Entry*), которая имеет наибольшее количество связей с другими сущностями и может являться словом, фразой или частью слова. Поскольку морфосинтаксическая информация связана со словарной статьей, каждая статья должна иметь стандартной написание. Варианты терминов, например, такие как аббревиатура, представляются в виде отдельных словарных статей и помечаются как lexicalVariants.

Словарная статья может состоять из нескольких **Форм** (Lexical Form), одна из которых может быть помечена, как лемма (*canonical form*).

Лексическое значение (*Lexical Sense*) представляет собой некое соответствие между словарной статьей (*Lexical Entry*) и сущностью онтологии (*Ontology*), которое устанавливается с помощью ссылки (*Reference*). Оно может включать дополнительную спецификацию этого соответствия, такую как контекст, условие, определения или примеры.

С помощью сущностей **Фрейм** (Frame) и **Аргумент** (Argument) в модели могут быть представлены валентность глаголов и других лексических предикаторов. Каждый аргумент также представлен как отдельная сущность, связанная как с фреймом, чтобы указать синтаксическую роль, так и с лексическим значением, чтобы указать семантическую роль.

Словарная статья может представлять многословные выражения и составные слова, для этого в модели имеется элемент **компонент** (Component), каждый из которых ссылается на другие словарные статьи.

Для представления в модели структуры словарной статьи, являющейся фразой существуют элементы **Узел** (*Node*). Они представляют собой ряд узлов, связанных ребрами или листовыми дугами с компонентами.

Разработчики этих графов знаний утверждают, что они разработаны в соответствии со стандартами LMF (Lexical Markup Framework), который классифицируется как ISO-

TC37/SC4/WG4. Данный стандарт подробно описан в работе [8]. На рис. 4 представлена одна из диаграмм, демонстрирующих структуру Базы данных в соответствии со стандартом LMF.

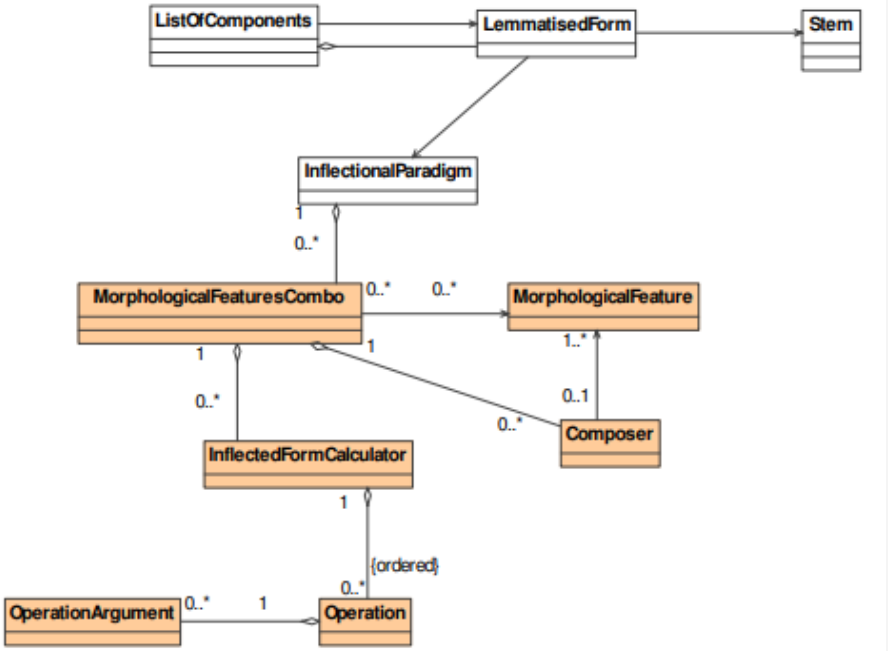


Рис.4. Структура БД для представления данных в соответствии со стандартом LMF

Вышеописанные лингвистические графы знаний подходят для описания языков типа английского с достаточно бедной морфологией и не позволяют адекватно описывать всю полноту лингвистической информации, имеющихся в агглютинативных языках с богатой морфологией. Для тюркских языков, также важно отобразить в онтологических моделях их морфологическую структуру, поскольку аффиксальные морфемы также несут отдельные значения. Особенность тюркских языков в том, что в отличие от языков флективного типа в них существует четкое деление словоформ на структурные компоненты, которые называются морфемами. Такое четкое деление позволяет морфологическую структуру словоформы представить в виде графа, вершинами которого являются морфемы.

Рассмотрим лингвистические графы знаний, которые отражают структурно-функциональные свойства тюркских языков и используются в качестве базы знаний портала “Тюркская морфема”. Интернет-портал «Тюркская морфема» [9] представляет собой web-сайт (modmorph.turklang.net), который предоставляет доступ к набору различных интернет-сервисов о тюркских языках и для работы с тюркскими языками.

Граф знаний портала “Тюркская морфема” можно условно поделить на несколько подграфов, каждый из которых представляет информацию соответствующего языкового уровня. Схема такого условного разделения представлена на рис.5.

Рассмотрим из каких основных условных блоков состоит граф знаний портала. Их можно разделить на три уровня:

1. Семантический блок;
2. Лингвистический блок;
3. Геолингвистический блок.

Семантический блок представляет собой описание языконезависимых элементов, которые можно назвать семантическими универсалиями. В этот блок входит описание таксономических структур типа WordNet и ситуационно-фреймовых структур типа FrameNet. В ситуационно-фреймовых структурах приводятся только типовые фреймы для описания ситуаций и наборы их ролей. Сами способы выражения этих ситуаций в текстах представлены в лингвистическом блоке.

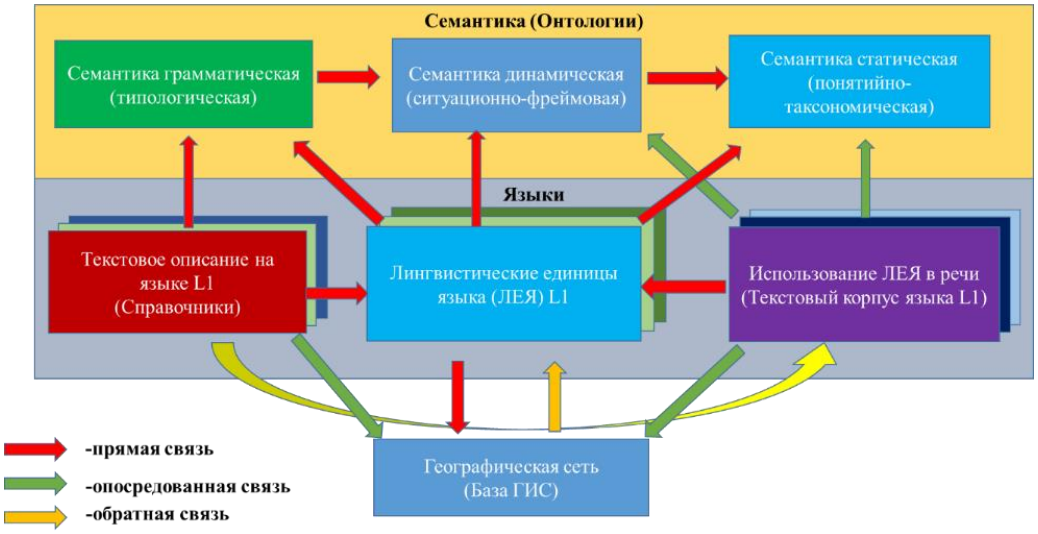


Рис.5. Архитектура подграфов графа знаний портала

Лингвистический блок представляет собой описание языковых единиц на различных языковых уровнях. При этом приводится как языковое описание в виде различных словарей, так и речевое описание с использованием этих языковых единиц в реальных текстовых корпусах. Также в лингвистический блок входит текстовое описание лингвистических единиц, которое в совокупности образует информационно-справочную систему по подобию Википедии.

Геолингвистический блок является привязкой лингвистической информации к пространственно-временным данным. Этот блок позволяет формировать сервисы типа инструмента для работы диалектолога, создавая описания не только литературных языков принятых за некий стандарт в определенном регионе, но и отдельных диалектов и говоров.

На рис.6 представлена онтологическая модель графа знаний портала “Тюркская морфема” без геолингвистического блока, поскольку геолингвистический блок еще находится на стадии разработки.

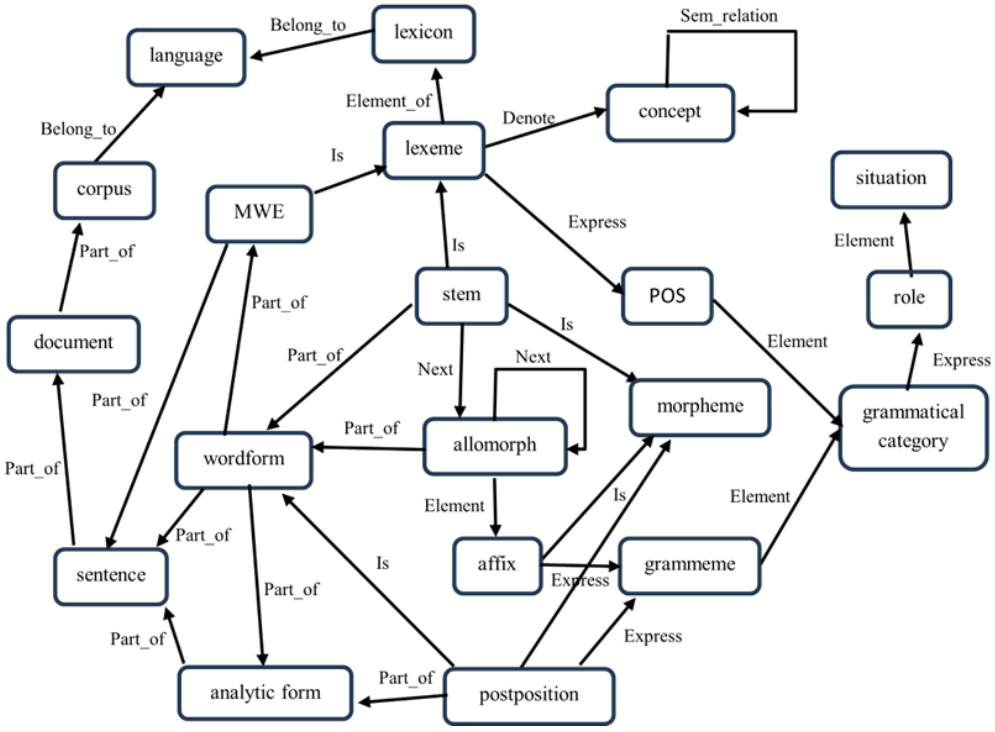


Рис.6. Фрагмент графа знаний портала

Центральным элементом графа знаний портала “Тюркская морфема”, представленного на рис.6 является Морфема (Morpheme). Как показано на схеме, морфема может быть трех видов:

1. Корневая морфема (stem);
2. Аффиксальная морфема (affix);
3. Аналитическая морфема, на данной схеме это один из видов аналитических морфем -послелоги (postposition).

Аффиксальные морфемы и аналитические выражают грамматические значения, которые названы на схеме граммемами (grammeme). Корневые морфемы выражают лексические значения (Lexeme), которые связаны с концептами (concept) тезауруса для описания таксономических значений. Лексемы могут быть выражены не только корневыми морфемами, но и многословными выражениями (MWE – multiword expression).

Для определения грамматических значений используются типовые ситуационные фреймы (situation) и роли (role) в этих ситуационных фреймах.

Данная модель заполняется конкретными лингвистическим данными для разных тюркских языков. На рис.7. представлен фрагмент, полученного графа знаний с уже конкретными данными для татарского и турецкого языков.

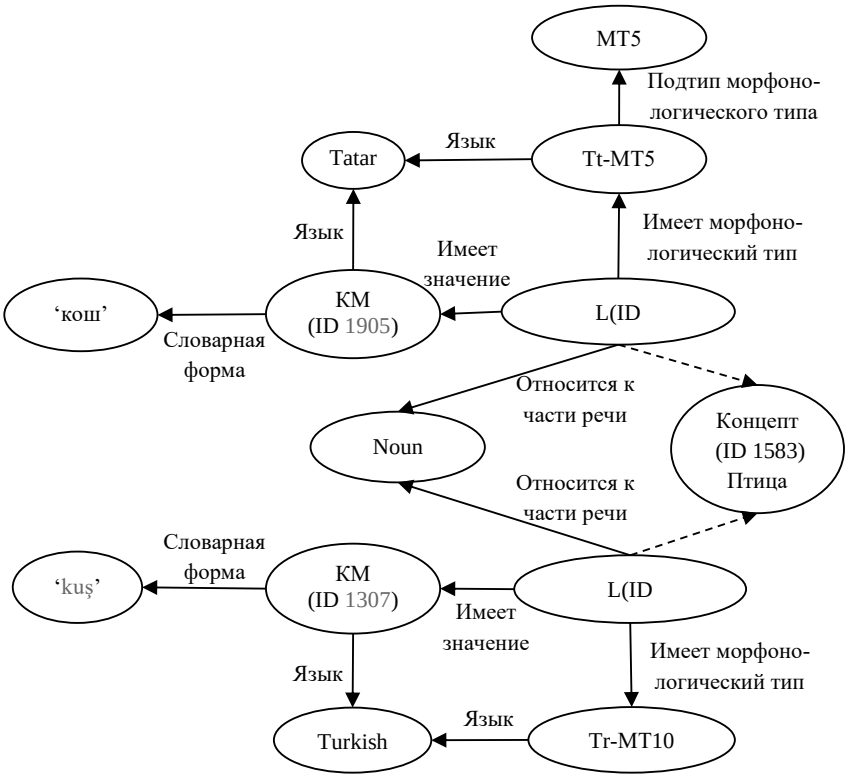


Рис.7. Пример фрагмента таксономического графа знаний

Рассмотрим фрагмент схемы взаимосвязей данных вершин графа, который представлен на рис.7. На данном примере КМ (ID 1905) и КМ (ID 1307) – это объекты – корневые морфемы татарского и турецкого языков с ID 1905 и 1307. Обе корневые морфемы в качестве значения имеют один и тот же концепт, но связь в графе осуществляется не напрямую, а через связующие вершины графа L(ID 1307x1583) и L(ID 1905x1583). ID этих элементов показывает, какие узлы графа знаний связывают эти элементы.

Все связующие элементы привязаны к морфонологическим типам Tr-MT10 и Tt-MT5. Эти вершины графа, обозначающие морфонологические типы, сами являются связующими элементами между вершинами графа, соответствующих корневым и аффиксальным морфемам тюркских языков. Они содержат информацию о морфотактических правилах, показывающих какие аффиксальные морфемы могут следовать в словоформе после этих корневых морфем.

Таким образом, данные фрагменты графа знаний являются мультифункциональными лингвистическими ресурсами. Они, с одной стороны, используются для представления справочной информации, которая может быть выдана пользователю по запросам, а с другой

стороны, описывают правила следования, используемые в морфологических анализаторах для тюркских языков.

Заключение

В статье описаны лингвистические графы знаний, которые позволяют представить лингвистическую информацию с учетом структурно-функциональных особенностей тюркских языков. В основе этих графов знаний лежат онтологические модели, которые являются общими для всех тюркских языков. В настоящее время модели, заполненные данными для более чем 30 тюркских языков позволяют получить единый лингвистический граф знаний портала. В портале реализуются механизмы для интеграции внешних электронных корпусов тюркских языков с порталом. Это должно позволить, с одной стороны, пополнять базы данных портала за счет анализа корпусных данных, а с другой — использовать конвейер программных модулей для обработки естественного языка и унифицированную систему аннотации для разметки корпусов.

В дальнейшем планируется комбинирование графов знаний портала с технологиями машинного обучения, как это было описано в блоке обзора разработок данной статьи. Авторы надеются, что таким образом это будет способствовать решению проблемы малоресурсности для тюркских языков, имеющих небольшие лингвистические ресурсы, или даже не обладающих соответствующими языковыми базами данных, при создании программных продуктов типа машинных переводчиков, распознавателей и синтезаторов речи.

Литература

[1] Krauwer S. The basic language resource kit (BLARK) as the first milestone for the language resources roadmap // Proc. International workshop on speech and computer SPECOM-2003, Moscow. 2003.

[2] A. Hogan et al., "Knowledge graphs," ACM Computing Surveys (CSUR), vol. 54, no. 4, pp. 1-37, 2021.

[3] S. Sharifirad, B. Jafarpour, and S. Matwin, “Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs,” in Proceedings of the 2nd workshop on abusive language online (ALW2), 2018, pp. 107–114.

[4] Pedro Colon-Hernandez, Catherine Havasi, Jason B. Alonso, Matthew Huggins, and Cynthia Breazeal. 2021. Combining pre-trained language models and structured knowledge. CoRR, abs/2101.12294.

[5] I. Tiddi and S. Schlobach, “Knowledge graphs as tools for explainable machine learning: A survey,” Artif. Intell., vol. 302, p. 103627, Jan. 2022, doi: 10.1016/j.artint.2021.103627.

[6] Pierpaolo Basile, Pierluigi Cassotti, Stefano Ferilli1, and Barbara McGillivray New Time-sensitive Model of Linguistic Knowledge for Graph Databases // Proceedings of the 1st Workshop on Artificial Intelligence for Cultural Heritage co-located with the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022), CEUR Workshop Proceedings. 2022. Vol.3286.

[7] J. P. McCrae, D. Spohr, P. Cimiano, Linking lexical resources and ontologies on the semantic web with lemon, in: G. Antoniou, M. Grobelnik, E. P. B. Simperl, B. Parsia, D. Plexousakis, P. D. Leenheer, J. Z. Pan (Eds.), The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I, volume 6643 of Lecture Notes in Computer Science, Springer, 2011, pp. 245–259. URL: https://doi.org/10.1007/978-3-642-21034-1_17.

[8] Francopoulo, G., George, M., Calzolari, N., Monachini, M., Bel, N., Pet, M., Soria, C.: Lexical markup framework (LMF). In: Proceedings of the Fifth International Conference on Language Resource and Evaluation (LREC 2006) (2006)

[9] Gatiatullin A., Suleymanov D., Prokopyev N., Khakimov B. (2020) About Turkic Morpheme Portal. CEUR Workshop Proceedings Institute for history, language and literature, Ufa scientific center, Russian Academy of Sciences Proceedings of TurkLang 2020, pp. 226-243.

Знания-Онтологии-Теории (ЗОНТ-23)

Глушкова В.Н.[1], Коровина К.С.[2]

[1]*Донской государственный технический университет, пл. Гагарина , д. 1, г. Ростов-на-Дону, 344000, Россия.*
[2] *Донской государственный технический университет, пл. Гагарина , д. 1, г. Ростов-на-Дону, 344000, Россия.*

lar@aaanet.ru, ksenichka_@inbox.ru

Полиномиально вычислимые регулярные модели систем реального времени

Аннотация. *В рамках концепции Σ-программирования [1], основанной на методах математической логики, разработан метод спецификации и верификации вычислимых регулярных ΣR-моделей дискретных систем реального времени. Модель строится по теории из универсальных квазитождеств специального вида из Δ_0 - формул с ограниченными кванторами по регулярным спискам. Динамика поведения моделируемой системы задается правилами регулярной грамматики G вида $st_1 \to act_1 \ldots act_k st_2$. Правила грамматики являются «шаблоном» для аксиом теории модели и отражают иерархию переходов системы от одного состояния (st_1) в следующее состояние (st_2) под воздействием действий $act_i,\ 1 \le i \le k$.*

Одноместные предикаты модели St_j, Act_j , соответствующие символам грамматики st_j, act_j , интерпретируются на множестве натуральных чисел – области значений функции «дискретное время» fdt. Функция fdt определяется на элементах списка из списочной надстройки модели, который представляет рекуррентное конечное дерево вывода tr в грамматике G. Дерево tr отражает все пути циклического графа без учета значений локальных переменных времени темпорального конечного автомата, на путях которого верифицируются RTCTL-формулы в идеологии model checking [2]. Значение функции fdt равно номеру позиции соответствующей вершины дерева (st_j, act_j) в упорядоченной последовательности всех его вершин при прохождении дерева от корня по стратегии "сверху-вниз", "слева-направо". Допускаются альтернативные действия (ветвление времени) и цикличное возвращение к исходному значению времени первоначального состояния. Вид временных функций реального (континуального) времени и ограничения на их значения зависят от типов проверяемых свойств анализируемой модели. Алгоритм построения модели реализует прямой логический вывод и имеет линейную временную сложность относительно "размера" модели. Выделены полиномиально реализуемые классы Δ_0T-формул для верификации свойств формальных моделей дискретных систем реального времени. Выразительные возможности Δ_0T-формул позволяют специфицировать все свойства систем реального времени, определяемые формулами темпоральных логик. Приведен пример построения формальной модели упрощенного варианта FDDI-протокола из [3]. Сформулированы свойства достижимости и безопасности для верификации формальной модели протокола.

Ключевые слова: верификация дискретных систем реального времени, Σ-программирование, многосортная модель со списочной надстройкой, Δ_0 - формулы с ограниченными кванторами, теория из универсальных квазитождеств с ограниченными кванторами, регулярная грамматика

1 Введение

Метод model checking имеет существенные трудности при верификации дискретных систем реального времени из-за неадекватного представления времени в темпоральном автомате. Автомат включает конечное множество X – локальных переменных-часов, принимающих действительные значения. На переходах, которые совершаются мгновенно, некоторым часам присваивается значение 0. Каждый переход между двумя состояниями помечен ограничением времени – защитой перехода. С состояниями автомата также связаны временные ограничения, которые называют инвариантами состояний. Время всех локальных переменных непрерывно

синхронно возрастает, пока инвариант состояния будет истинным. Временные ограничения задаются выражениями вида $x \prec c$, $c \prec x$, где $\prec \in \{<, =, \leq\}$, c – неотрицательное рациональное число. Поведение временного автомата определяется его операционной семантикой, которая представляется графом переходов T с континуальным числом вершин. Состояния и дуги автомата помечены значениями часов в соответствии с интерпретацией υ: $X \to R^+$, где R^+ – множество положительных рациональных чисел. Интерпретация υ трактуется как присваивание локальным часам $x \in X$ всех вещественных значений из соответствующих сегментов, удовлетворяющих инвариантам состояний и ограничениям переходов.

Для автоматической верификации вместо бесконечной системы переходов T используют фактор-систему T_π по отношению эквивалентности π, которая определяется на основе понятия временного региона. Каждый регион определяется набором ограничений на часы, а число регионов определяется соотношением $|X|!\, 2^{|X|} \prod_{x \in X}(2C_x + 2)$, где C_x – наибольшая константа, с которой x сравнивается в инвариантах или защите переходов.

Поиск более выразительных и простых в применении логических языков спецификации одна из открытых проблем model checking [2]. В качестве такого языка будем использовать выделенные в концепции Σ-программирования Σ-спецификации. Σ-спецификации позволяют адекватно и полно отражать специфические статические и динамические аспекты предметных областей. Особенность этой концепции состоит в изучении многосортных моделей с наследственно конечной списочной надстройкой. Значимость таких моделей мотивируется универсальностью списочных структур, которые можно использовать для представления конструктивных объектов (графов, деревьев, автоматов и др.). Система знаний об этих объектах представляется формальной теорией.

В [4] рассматривались Σ-модели с КС-списочной надстройкой, порождаемой КС-грамматикой. Там же выделен полиномиально вычислимый класс $\Delta_0 T$ -формул с ограниченными кванторами по КС-спискам, который использовался для спецификации семантики языков программирования [5] – [6] и верификации технических систем [7]. Аналогично можно определить модели со списочной надстройкой, порождаемой регулярными грамматиками и выделить классы $\Delta_0 T$ - формул для построения и верификации ΣR-моделей дискретных систем реального времени.

2 Модели со списочными надстройками

2.1 Основные определения

По теории предметной области и теории списочных расширений *GES* [1] построим вычислимую многосортную модель M сигнатуры $\sigma = < I_M, C, F, P, \rho >$, где $I_M = I \cup \{list\}$ – множество сортов, *list* – сорт "список"; C, F, P – множества констант, функций и предикатов соответственно; ρ – функция из $F \cup P$ во множество упорядоченных наборов из I. Универс U модели M состоит из семейства носителей $\{U_i\}_{i \in I}$, где U_i – множество элементов сорта i и списочной надстройки $S^{fin}(M)$ сорта *list*. Надстройка состоит из списков, которые формируются из элементов множества $B = \bigcup_{i \in I} U_i$, $S^{fin}(M) = \bigcup_{n \geq 0} S^n(M)$, где $S^0(M) = \{< \alpha_1, \ldots, \alpha_n > | \alpha_j \in B,$ $1 \leq j \leq n\}$, $S^{i+1}(M) = S^i(M) \cup \{< \alpha_1, \ldots, \alpha_n > | \alpha_j \in S^{i-1}(M) \cup B, 1 \leq j \leq n\}$. На множестве $S^{fin}(M)$ определяются основные операции и отношения для списков: *conc* (конкатенация списков), *cons* (присоединение к списку нового элемента), $\in$ (отношение принадлежности элемента списку) и др. Все сигнатурные символы, термы, переменные, константы индексируются типами – упорядоченными наборами сортов . Термы определяются традиционно: переменные и константы сорта $i \in I_M$ – термы этого же сорта. Если $f \in F$ функциональный символ типа $< i_1 \ldots, i_n, i >$, то f интерпретируется в модели M как операция: $f: U_{i_1} \times \ldots U_{i_n} \to U_i$; если $t_1 \ldots., t_n$ – допустимые для f термы, то $f(t_1 \ldots., t_n)$ – терм сорта i. Аналогично определяются формулы с использованием всех логических связок за исключением формул с ограниченными кванторами. Пусть φ – формула , t – терм сорта *list*, x – переменная сорта $< I \cup \{list\} >$, тогда выражения вида $\forall x \in t, \exists x \in t, \forall x \subseteq t, \exists x \subseteq t$, где x – переменная произвольного типа , t – терм сорта *list* (t не содержит x) также будут формулами; кванторы в этих формулах называют ограниченными.

Определение 1. Формула, в записи которой встречаются только ограниченные кванторы называется формулой с ограниченными кванторами, или Δ_0 – формулой.

В [8] отмечено, что наиболее важным и общим примером инициальной многосортной алгебраической семантики является семантика для КС-грамматик $G = (T, N, P)$, где T, N, P — множества терминальных, нетерминальных символов и правил соответственно. Носителями сорта A в инициальной G - алгебре являются все деревья вывода из символа A, порожденные правилами грамматики G. Каждому правилу $A_0 \to u_0 A_1 \ldots u_{n-1} A_n$, $u_i \in T^*$, $A_i \in N$, $0 \le i \le n$, соответствует операция (функция) G – алгебры типа $<A_1, \ldots, A_n, A_0>$.

Над семейством констант $C = \{C_X\}_{X \in V}$ определим список, представляющий произвольное упорядоченное дерево вывода tr с корнем X_0 в КС-грамматике G. Пусть $V = N \cup T$ и C_X – не более чем счетное множество констант сорта $X \in V$. Сопоставим взаимно однозначно каждой вершине, помеченной символом X, константу из C_X . Пусть $X_0 \to X_1 \ldots X_n$ — первое правило дерева вывода tr.

Определение 2. Дерево tr представляет список $l = <t_1, \ldots . t_n>$, где $t_i \in C_{X_i}$, если X_i — лист tr, иначе t_i – список**,** который представляет поддерево tr с корнем X_i, $1 \le i \le n$.

Считаем, что сорт ρ всех списков задается меткой корня соответствующего дерева, $\rho(l) = X_0$, $\rho(t_i) = X_i$, $1 \le i \le n$.

При таком представлении деревьев списками индукцией по высоте дерева легко доказать, что отношению принадлежности элемента e списку l соответствует отношение непосредственного подчинения вершин v_1 и v_2 дерева, где v_1, v_2 – корни поддеревьев, представленные элементом e и списком l соответственно. Транзитивному замыканию отношения принадлежности списков соответствует отношение подчинения соответствующих вершин дерева. Каждый элемент e списка l взаимно однозначно сопоставляется вершине дерева вывода.

Приведем регулярную грамматику, эквивалентную конечному временному автомату FDDI-протокола [3] (без учета временных переменных).

Пример1: $G_1 = (N, T, P)$. Множество состояний протокола $N = \{ idle, str, atr\}$, где *idle* – ожидание специального маркера (token); *str* – синхронная передача данных; *atr* – асинхронная передача данных. Множество действий протокола $T = \{tt, bs, rt, es, ba, ea\}$, где *tt* – получение маркера; *bs* – начало синхронной передачи; *rt* – возвращение маркера; *es* – окончание синхронной передачи данных; *ba* – начало асинхронной передачи данных; *ea* – окончание асинхронной передачи данных. Правила P:

idle $\to$ *tt bs str;*

str $\to$ *rt es idle | es ba atr;*

atr $\to$ *ea rt idle*

Дерево вывода tr (Рис 1) представляется списком $l = < tt_1, bs_1, < es_2, ba_2, < ea_3, rt_3, idle_3 >>>$, $\rho(l) = idle$, $\rho(< es_2, ba_2, < ea_3, rt_3, idle_3 >>) = str$, $\rho(< ea_3, rt_3, idle_3 >) = atr$. Индексы всех вершин равны их глубине в дереве.

Класс $\Delta_0 T$ - формул с ограниченными кванторами по КС-списку использовался в [7] для анализа и верификации технической системы. Ограниченность этого подхода по сравнению с model checking обусловлена способом определения выводов в КС-грамматике, т.к. в одном дереве вывода не учитываются все альтернативы нетерминальных символов в отличие от графа конечного темпорального автомата. Классическое дерево вывода не отражает ветвления времени и его цикличности. Приведем определение $\Delta_0 T$ – формулы**,** в которых используются лишь ограниченные кванторы вида $\forall\, v \in r$, $\forall\, v \overset{*}{\in} r$, $\forall\, u \prec v$, где u, v – переменные любого сорта; r – переменная сорта *list*. Отношение $\overset{*}{\in}$ – рефлексивное транзитивное замыкание $\in$; $\prec$ – отношение «левее» для элементов списочных объектов. Пусть $\bar{x}$ обозначает индексированную последовательность переменных x, $\dot{\in}$ – отношение $\in$ или $\overset{*}{\in}$.

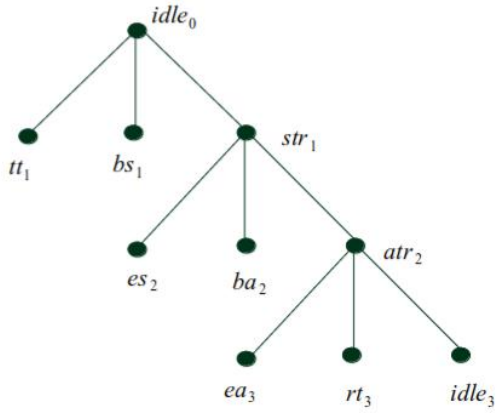


Рис 1. Дерево вывода *tr* в грамматике G_1 .

Определение 2. Δ_0 –формула вида

(1) $(\forall v_1 \dot{\in} r_1) \ldots (\forall v_m \dot{\in} r_m)\ (w_1 \prec u_1) \ldots\ (w_k \prec u_k)\ \Phi(\bar{v}, \bar{r})\ ,\ m \geq 1,\ k \geq 0$

называется $\Delta_0 T$ – формулой, если w_j , $u_j \in (\bar{v}\ , \bar{r}\ )$, $1 \leq j \leq k$, и переменные префикса удовлетворяют условию "древесности": для всех $1 \leq i \leq m$, $l \leq i$ выполняется $r_{i+1} = r_i$ или $r_{i+1} = v_l$. Если $r_{i+1} = v_l$, то для всех $p \leq i$ выполняется $v_{i+l} \neq v_p$, $v_{i+l} \neq r_p$.

Если представить все переменные префикса узлами с дугами, ведущими от r_i к v_i, то получится дерево с корнем r_1 .

В [7] построена формальная модель автоматизированного технологического комплекса по теории из $\Delta_0 T$ – квазитождеств вида:

$(\forall v_1 \dot{\in} r_1) \ldots (\forall v_m \dot{\in} r_m)\ (w_1 \prec u_1)\ \ldots\ (w_k \prec u_k)\ (\varphi(\bar{v}, \bar{r}) \rightarrow \psi(\bar{v}, \bar{r}))$, $m \geq 1,\ k \geq 0$, где

формула $\varphi\,(\psi)$ – это конъюнкция атомных формул (или их отрицаний) вида: r, $\tau_1 = \tau_2$, $(f = \tau)$, $f \in F, r \in R$, τ_1, τ_2 – термы сигнатуры σ.

Правила КС-грамматики описывают последовательность операций *act* (действий) комплекса с учетом дискретного и реального времени для состояний комплекса *st*:

$r \rightarrow \{act\}^*$

$act \rightarrow act_1 | \ldots | act_n$

$act_i \rightarrow st, \quad 1 \leq i \leq \mathrm{n}$

$st \rightarrow td \mid tc$

$td \rightarrow nat \mid real$

$tc \rightarrow (td, td) \mid [td, td\ ] \mid (td, td\ ] \mid [td, td)$

Символы грамматики соответствуют предикатам сигнатуры модели. В процессе интерпретации аксиом теории по правилу вывода modus ponens формируется дерево вывода грамматики с конкретными значениями дискретного и реального времени в соответствующих вершинах дерева, извлекаемые из области истинности одноименных предикатов. Выбор действий $Act_i(n)$ с конкретным значением n зависит от сигнала из потока данных, представленного списком. На списочном представлении дерева проверяются полиномиально реализуемые $\Delta_0 T$ – формулы, которые выражают требования к модели. Адекватное представление времени – существенное преимущество предлагаемого способа спецификации системы. Однако его существенное ограничение состоит в линейном способе представления действий без учета альтернативных действий в дереве вывода.

Формализм временных конечных автоматов поддерживает концептуальную структуру динамической системы, основанную на представлении действий переходами из одного состояния в следующее с учетом ветвления и цикличности времени. Ниже выделим класс регулярных грамматик, правила которых соответствуют переходам конечного временного автомата. Введем понятие полного вывода, содержащего все альтернативы для всех нетерминальных символов. Выделим класс линейно вычислимых моделей по теории из $\Delta_0 T$ –квазитождеств, интерпретируемых на регулярной списочной надстройке модели.

2.2 Регулярные **Σ** - модели

Грамматика $G = (N, T, P, st_0)$ на верхнем уровне специфицирует моделируемую систему,

где $N = \{st_j | \ 0 \leq j \leq m\}$– множество символов – состояний; $T = \{act_i | 0 \leq i \leq m\}$– множество символов – действий. Правила грамматики из упорядоченных альтернатив символа $st \in N$ имеют вид:

(2) $st \rightarrow \propto_1 st_1 \ | \quad \ldots | \propto_k st_k$, где $\propto_j \in T^+$, $st_j \in N$, $0 \leq j \leq k, k \geq 1$.

Все правила вида (2) объединим в одно полное (complete) *c*- правило вида

(3) $st \rightarrow \propto_1 st_1 \ldots \propto_i st_i \ \ldots \ \propto_k st_k, \ 1 \leq i \leq k.$

Учитывая шаблон правил *G*, все альтернативы символа *st* из цепочки $\propto_1 st_1 \ldots \propto_i st_i \ \ldots \ \propto_k st_k$ выделяются однозначно.

Вывод, в котором применяются только *c*- правила для символов с несколькими альтернативами, назовем *полным c- выводом*.

Пример 2 *c - вывода в грамматике* G_1:

idle $\Longrightarrow$ *tt bs str* $\Longrightarrow$ *tt bs rt es idle es ba tr* $\Longrightarrow$ *tt bs ea rt idle es ba atr* $\Longrightarrow$ *tt bs ea rt idle es ba ea rt idle*

c- вывод, в котором каждое *c*-правило встречается только один раз, называется рекуррентным *c*-выводом. Этому выводу соответствует рекуррентное полное *c*-дерево вывода на Рис.2.

Основные предикатные символы сигнатуры модели соответствуют символам грамматики:$\{Act_i, St_j | \ 0 \leq i \leq m, 0 \leq j \leq n\} \subseteq P$. Для ясности изложения пусть правила грамматики имеют вид:

(4) $st_0 \rightarrow act_1 st_1 \ | \quad \ldots | act_k st_k, \ k \geq 1$.

Одноместные предикаты $Act_i, St_j \subseteq N$ зависят от значения функции *fdt* – “дискретное время”, которая определяется на элементах регулярного списка *r*, соответствующего рекуррентному *c*-дереву *tr*. Сначала зададим функцию *fdt* на вершинах дерева. Обозначим все вершины дерева символами грамматики, индексированными их глубиной в дереве и упорядочим их в соответствии с обходом дерева по стратегии “сверху-вниз”, “слева-направо”:

(5) $st_0^0, act_1^1, st_1^1, \ \ldots, \ act_i^j, st_i^j, \ \ldots, \ act_n^m, st_n^m$.

Тогда $fdt\,(st_0^0) = 0$, $fdt\,(act_i^j) = fdt\,(st_{i-1}^j) + 1$; если в последовательности (5) нет вершины, предшествующей st_i^j вида st_i^k, $k < j$, то $fdt\,(st_i^j) = fdt\,(act_i^j)$, иначе $fdt\,(st_i^j) = fdt\,(st_i^k)$. Такое определение времени поддерживает его цикличность. Функцию *fdt* определим на элементах e_i списка *r* таким образом, что значение $fdt\,(e_i)$ совпадает со значением дискретного времени вершины дерева, соответствующей элементу e_i.

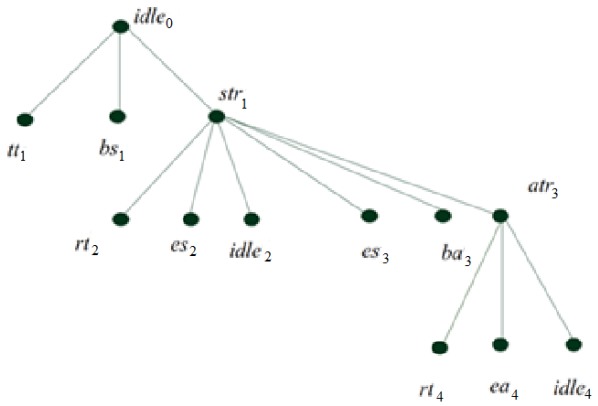


Рис 2. Рекуррентное дерево вывода *ctr* в грамматике G_1

Если в дереве *ctr* отождествить вершины $idle_0, idle_2, idle_4$ с одинаковым значением 0 дискретного времени, то дерево "сворачивается" в граф станции FDDI-протокола на Рис.3, который идентичен графу станции временного автомата FDDI-протокола из [3, Fig.1, стр. 213].

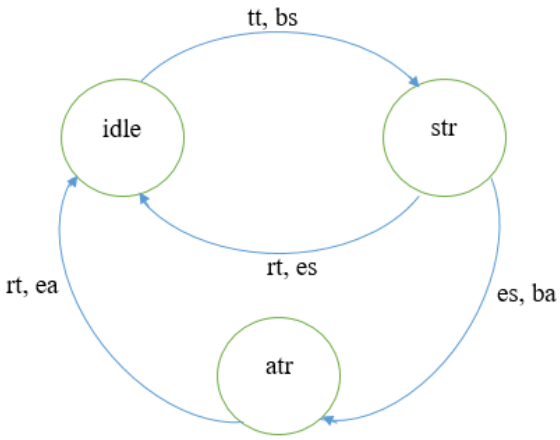


Рис 3. Станция FDDI- протокола

Границы значений функций реального времени *frt* с областью определения N задаются как в model checking отношениями вида *frt* $(n) \sim c_i$, где $\sim \in \{\leq, \geq, =, <, >\}$, $c_i \in C$. Правила грамматики являются шаблоном аксиом теории модели. Аксиомы, не содержащие функций реального времени, имеют вид:

$$(6) \quad (\forall \overline{v} \in v_0)(\forall \overline{w} \in v_0)(\overline{v} \prec \overline{w})(St_0\left(fdt(v_0)\right) \longrightarrow \bigvee_{i=1}^{k}(Act_i(fdt(v_i)) \wedge St_i\left(fdt(w_i)\right)),$$

где $\overline{v}$, ($\overline{w}$) – индексированные последовательности $v_1, .., v_k$ ($w_1, \ldots, w_k$). Если аксиома содержит функции реального времени, то функции вида *frt* $(dt(v_i)) \sim c_i$ входят в левую и правую часть аксиомы (6).

Для вычисления функции fdt необходимо в теорию добавить аксиомы, содержащие в левой части импликации ограниченные негативные кванторы $\exists$ по списку элементов тех сортов $st_j, 0 \leq j \leq m$, которые многократно встречаются в списке r. Обозначим через $D_G(C)$ – мно-

жество всех списков, представляющих *e*-деревья выводов в регулярной грамматике G над семейством констант $C = \{C_x\}_{x \in V}$. Для существования модели на теорию накладываются такие же ограничения как в [9] за исключением условия ацикличности теории, которая следует из способа определения функции *fdt*.

Теорема. Всякая определимая во множестве констант сигнатуры σ эквационально корректная теория *Th* из $\Delta_0 T$– формул вида (6) имеет модель из констант сигнатуры с регулярной надстройкой $D_G(C)$.

Алгоритм построения модели *Int* реализует правило вывода "modus ponens" (если φ и $\varphi \rightarrow \psi$, то ψ). Входным данным для интерпретатора является факт *fdt(r)* = 0, если теория расширена квазитождествами для определения функции *fdt* на элементах всего списка *r*. При построении модели все дизъюнкты из левой части квазитождества (6) устанавливаются истинными для отражения всех альтернативных действий и состояний. Если количество цикличных состояний ограничено, то интерпретация теории осуществляется за линейное время *n*, относительно "размера" моделируемой системы, иначе сложность интерпретации *O* (n^2), где n −количество элементов списка *r* равная мощности множества $\{e | e \in^{*} r\}$.

2.3. Спецификация FDDI-протокола

Стандартный FDDI- протокол передает потоки синхронных и асинхронных сообщений на основе технологии Tooken Ring [10]. Станция получает данные от предшествующей ей в кольце станции и передает поток данных следующей станции. Данные может предавать только станция, которая обладает маркером (token – сигнал, вращающийся по кольцу). Передача данных осуществляется в двух режимах: синхронном и асинхронном. При инициализации сети определяется ожидаемое время обхода кольца маркером *ttrt* (target token rotation time). Каждой станции отводится гарантированное время для передачи данных в кольцо. По истечении этого времени станция заканчивает передачу сообщений и посылает маркер в кольцо следующей станции. Каждая станция имеет 2 таймера: первый измеряет временной интервал *trt* (token rotation timer) после последнего прибытия маркера; второй – считает время *tht* (token holding timer) удержания маркера в сети. В FDDI-сети время владения разделяемой средой не является постоянной величиной. Сначала стация передает синхронные сообщения за ограниченное константой время. Если маркер возвратился к станции раньше ожидаемого времени обхода кольца *ttrt*, то станция может продлить время передачи своих данных. В этом случае выделяется временной интервал для асинхронных сообщений, который равен разности между ожидаемым и реальным временем обхода кольца маркером.

Синтаксическое описание протокола регулярной грамматикой G_1 = *(N, T, P)* приведено в примере1 раздела 2.1. с рекуррентным деревом вывода на Рис 2. Особенность этой модели состоит в том, что цикличное дискретное время используется только в одном начальном состоянии *Idle* со значением 0, известном до начала интерпретации. Для таких систем значение функции *fdt* определяется шагом интерпретации *n*. Теория *Th* модели протокола состоит из пяти аксиом. Интерпретация предикатов и функций, начинается, исходя их факта $Idle(i, 0)$, где i – номер станции, $1 \leq i \leq m$. Первая аксиома специфицирует переход из состояния ожидания маркера *Idle* (*i, n*) на шаге *n* в состояние передачи синхронных сообщений $Str(i, n+1)$ на следующем шаге. После получения маркера на этом шаге (действие $Tt(i, n+1)$) начинается передача синхронных сообщений ($Bs(i, n+1)$) . Функция $Frt(i, n+1) = trt_i$ определяется равной неинициализированной константе trt_i – времени, прошедшему после последнего приема *i*-ой станцией маркера.

Аксиомы 2.1 и 2.2 описывают альтернативное поведение протокола, выраженное двумя дизъюнктами. Первый дизъюнкт реализуется при $Frt(i, n+1) \geq ttrt$, тогда маркер возвращается *i*-ой станцией $Rt(i, n+1)$ с окончанием передачи синхронных сообщений $Es(i, n+1)$. В этом случае функция *Fht* (*i*, n+1) = sa_i и *i*+1- станция переходит в начальное состояние $Idle(i+1, 0)$. При $Frt(i, n+2) < ttrt$ выполняется альтернативное действие – заканчивается передача синхронных сообщений ($Es(i, n+2)$) и начинается передача асинхронных сообщений *Ba* (*i, n+2*) с переходом в следующее состояние *Atr* (*i, n+2*) – передачи асинхронных состояний на шаге *n+2*. В этом случае *Fht* (*i, n+2*) = $sa_i + aa_i$. В аксиоме 2.2 специфицируется кольцевой способ передачи маркера: последняя станция с номером *m* передает маркер первой станции.

С учетом кольцевой структуры передачи маркера аксиомы 3.1 и 3.2 специфицируют: после передачи асинхронных сообщений маркер возвращается, передача асинхронных сообщений заканчивается $(Ea(i,n+1))$ и следующая станция переходит в начальное состояние "ожидание маркера" $Idle(i+1,0)$. Во всех аксиомах с помощью функции $Gr_i\ (n)$ последовательно формируется список смежных вершин графа на Рис.3. Для его формирования используются начальные факты: $Gr\ (0) = <<<idle,0>>>$, $Na\ (0) = Nv\ (0)=1$. В аксиомах все переменные i связаны квантором $\forall i \in [1,m]$, переменная n квантором $\forall n \in [0,3]$.

1) $Idle(i,n) \to Tt(i,n+1) \wedge Bs(i,n+1) \wedge (Frt(i,n+1) = trt_i \wedge Str(i,n+1) \wedge$

$(Gr(n+1) = cons(Gr(n), \langle\langle tt, bs, Na(n)\rangle, \langle str, Nv(n)\rangle\rangle)) \wedge$

$(Nv(n+1) = Nv(n)+1\rangle \wedge (Na(n+1) = Na(n)+1);$

2.1) $\forall i \in [1,m-1](Str(i,n) \to (Rt(i,n+1) \wedge Es(i,n+1) \wedge Frt(i,n+1) \geq trtt \wedge Fht(i,n+1) = sa_i \wedge$

$Idle(i+1,0) \vee Es(i,n+2) \wedge Ba(i,n+2) \wedge (Frt(i,n+2) < trtt \wedge Atr(i,n+2) \wedge$

$Fht(i,n+2) = sa_i + aa_i) \wedge Gr(n+2) = cons(Gr(n), \langle\langle rt, es, Na(n)\rangle,$

$\langle idle, 0\rangle, \langle es, ba, Na(n)+1\rangle, \langle atr, Nv(n)+1\rangle\rangle \wedge Nv(n+1) = Nv(n)+1 \wedge\ Na(n+1) = Na(n)+2);$

2.2) $Str(m,n) \to (Rt(m,n+1) \wedge Es(m,n+1) \wedge Frt(m,n+1) \geq trtt \wedge Fht(m,n+1) = sa_i \wedge$

$Idle(1,0) \vee Es(m,n+2) \wedge Ba(m,n+2) \wedge (Frt(m,n+2) < trtt \wedge Atr(m,n+2) \wedge$

$Fht(m,n+2) = sa_i + aa_i) \wedge Gr(m+2) = cons(Gr(m), \langle\langle rt, es, Na(n)\rangle,$

$\langle idle, 0\rangle, \langle es, ba, Na(n)+1\rangle, \langle atr, Nv(n)+1\rangle\rangle \wedge Nv(n+1) = Nv(n)+1 \wedge\ Na(n+1) = Na(n)+2;$

3.1) $\forall i \in [1,m-1](Atr(i,n) \to Rt(i,n+1) \wedge Ea(i,n+1) \wedge Idle(i+1,0) \wedge$
$Gr(n+1) = cons(Gr(n), << rt, ea, Na(n), < idle, 0 >>)$

3.2) $Atr(m,n) \to Rt(m,n+1) \wedge Ea(m,n+1) \wedge Idle(1,0) \wedge$
$Gr(n+1) = cons(Gr(n), << rt, ea, Na(n), < idle, 0 >>$

В результате интерпретации получаем следствия: $Tt(i,1)$, $Bs(i,1)$, $Str(i,1)$, $Rt(i,2)$, $Es(i,2)$, $Frt(i,2) \geq ttrt$, $Fht\ (i,2) = sa_i$, $Es(i,3)$, Ba(i,3), Frt$(i,3) < ttrt$, $Atr(i,3)$, $Fht(i,3) = sa_i + aa_i$, $Rt(i,4)$, $Ea(i,4)$, $1 \leq i \leq m$. Список смежности графа (Рис.3) равен значению функции:

$Gr(4) = <<<idle, 0>, << tt, bs, 1>, < str, 1>>, <<str, 1 >, <<rt, es, 2 >, <idle, 0>, <es, ba, 3 >>, < atr, 2 >>, <<atr, 2 >, <rt, ea, 4 >, < idle, 0>>>$

2.3. Верификация FDDI-протокола

Цель верификации модели протокола состоит в проверке требований к FDDI-протоколу. На модели должны выполняться определенные причинно-следственные связи событий: "получение маркера i-ой станцией приведет к передаче синхронных или асинхронных событий". Некоторые события не должны выполняться одновременно. Традиционно выделяются следующие классы свойств: свойства достижимости (reachability), свойства безопасности (safety), свойства живости (liveness), свойства справедливости (fairness). [11] Свойства достижимости одни из наиболее часто проверяемых свойств: некоторая конкретная ситуация в процессе функционирования FDDI-протокола случится. Например: если i-ая станция находится в состоянии ожидания token, то на некотором шаге своего функционирования при выполнении определенных действий она будет посылать асинхронные сообщения. Это свойство выражается Δ_0 – формулой:

1) $(\forall\, i \in [1,m] \exists\, n \in [0,3])\ (\ Idle(i,0)\ \to Es(i,n)\ \wedge Frt(i,n)\ < ttrt\ \wedge Art(i,n))$;
2) "i-ая станция всегда возвращает маркер" специфицируется формулой:

$(\forall\, i \in [1,m] \exists\, n \in [0,3])\ (Idle(i,0)\ \rightarrow Rt(i,n))$;

3) в модели реализуются две ситуации $Frt\ (i, n_1\ ) \geq ttrt$ *и* $Frt\ (i, n_2) < ttrt$:
$(\forall\, i \in [1,m]\ \exists\, n_1, n_2 \in [0,3])\ (Idle(i,0)\ \rightarrow Frt\ (i,n_1) \geq ttrt\ \wedge Frt(i,n_2) < ttrt)$

Свойства безопасности гарантируют, что при определенных условиях некоторые ситуации никогда не выполняются:

1) ситуация “любая станция никогда одновременно не посылает синхронные и асинхронные сообщения” проверяется проверкой формулы:
$(\forall\, i \in [1,m]\ \neg\exists\, n \in [0,3])(Str(i,n)\ \wedge Atr(i,n))$;
2) безопасность может формулироваться с дополнительным условием: передача асинхронных и синхронных сообщений осуществляется только после получения станцией маркера:

$(\forall\, i \in [1,m] \forall\, n_1, n_2\, \epsilon [0,3]\ \exists\, l \in [0,3]\ )(Str(i,n_1)\ \vee\ Atr(i,n_2\ )\ \rightarrow Tt(i,l)) \wedge l < n_1 \wedge l < n_2)$

3. Заключение

В [2] отмечается: “проверку на модели с использованием темпоральных логик нельзя естественно и эффективно использовать для верификации многих типов систем реального времени, которые часто встречаются на практике”. В отличие от темпоральных логик язык ИП 1-го порядка, используемый в Σ- спецификации, позволяет адекватно выразить такие временные свойства как “событие p произойдет не позднее чем через n – единиц времени”, время реакции и количество произошедших событий. Используемые для верификации систем Δ_0 – формулы можно доказывать индукцией по параметрам системы и позициям вершин в дереве вывода грамматики. Например, оценки двух свойств FDDI-протокола “ограниченного доступа к кольцу” в формальной модели протокола более низкие чем в [3]. Первое свойство оценивает время, прошедшее между двумя последовательными событиями $Tt\ (i,n)$ – получения token i-ой станцией. Это время ограничено константой $ttrt + \sum_{i=1}^{m} sa_i$ в отличие от оценки $ttrt + 2m \cdot sa_i$ в TCTL-логике . Второе свойство “ограниченного времени для посылки асинхронных сообщений”: каждая станция имеет возможность послать асинхронное сообщение прежде времени $(m - 1)\ ttrt + \sum_{i=1}^{m} sa_i$. Эта оценка ниже оценки $(m - 1)\ ttrt\ +\ 2m \cdot sa_i$ в TCTL-логике. Эти формулы можно доказать индукцией по количеству станций в кольце. В [3] эти оценки верифицируются для конкретных значений констант $ttrt$, m, sa_i.

Выразительные возможности ИП 1-го порядка, обогащенного списочными структурами, позволяют построить вычислимые стохастические модели для протоколов, используя вероятностные методы model checking. Внесение вероятностных оценок на переходы состояний позволяет расширить область применения классической верификации, выполняя более детальный анализ свойств поведения дискретных систем. Актуальным направлением является апробация предлагаемого подхода для верификации параллельных и распределенных систем, не имеющих регулярного шаблона взаимодействия.

Литература

[1] S.S. Goncharov, Yu.L Ershov., and D.I. Sviridenko. Semantic programming. Information Processing, 86, 1093-1100, North-Holland, Amsterdam, 1986.

[2] М. Кларк, мл. О. Грамберг, Д. Пелед. Верификация моделей программ. Изд-во Московского центра непрерывного математического образования. Москва, 2002.

[3] C. Daws, , A. Olivero, S. Tripakis, S. Yovine, C. Daws. The Tool Kronos. Book Lecture Notes in Computer Science, V.1066, 1966, Hybrid Systems III, Verification and Control, p.208-219.

[4] Glushkova V.N. Estimating the execution complexity of logical specifications based on context-free grammars. Kibernetika i Sistemnyi Analiz. No. 4, July-August, 1996. pp.50-58.

[5] Глушкова В.Н. Σ- спецификация языков программирования. Языки программирования и компиляторы -2017. Всероссийская научная конференция памяти А.Л. Фуксмана. Труды конференции, 3-5 апреля, Ростов-на-Дону, 2017, стр. 78-81.

[6] Глушкова В.Н., Ильичева О.А. К проблеме видимости и типизации в языке Модула-2. Язык Модула-2, его реализация и использование. Сборник научных трудов. Новосибирск, 1989, стр. 111-12

[7] V. N. Glushkova, K. S. Korovina. Polynomially computable Σ−specifications of hierarchical models of reacting systems. Advanced Engineering Research, 2020. V. 20, no. 4. p. 422−429.

[8] J. A. Goguen, J.W. Thatcher, E.G. Wagner, J.W.Wright. Initial Algebra Semantics and Continuous Algebras. Journal of Association for Computing Machinery, V.24, No.1, January 1977, pp. 66-95.

[9] Ильичева О.А. О семантике квазитождеств, определяющих модели из констант. Прикладная логика. Вычислительные системы, 116. Новосибирск, 1986, стр.16 – 32.

[10] Букатов А.А., Гуда С.А. Компьютерные сети: расширенный начальный курс, Питер, Санкт-Петербург, 2019.

[11] Карпов Ю.Г. Model checking. Верификация параллельных и распределенных программных систем, Санкт-Петербург, 2010.

Опыт Выбора Инструмента для Визуальной Разработки Онтологии

Эльвира Яковлевна Гринберг[1], Мирослав Валерьевич Кубельский[2]

[1]Санкт-Петербургский Государственный Университет, Лаборатория цифрового моделирования, Университетская наб., д. 7/9, г. Санкт-Петербург, 199034, Россия.
[2] СеилзАИ, ул. Донская, 4 стр.3, Москва, 119049, Россия.

st057454@student.spbu.ru, mirqube@gmail.com

Аннотация. *Онтологическое моделирование является неотъемлемым этапом построения карты знаний сложной предметной области. Визуальная разработка онтологии совместно с экспертом значительно повышает скорость и качество построения модели. Эксперт предметной области нуждается в упрощенном и интуитивно-понятном средстве визуализации знаний, в то время как аналитик заинтересован в высокой детализации модели и возможности (полу)автоматической формализации полученных диаграмм. В данной работе выявлены средства визуальной разработки онтологий, которые позволяют снять противоречие между потребностями эксперта и аналитика. Для определения таких средств было выполнено исследование, результатами которого стали: 1) обзор актуальных средств визуальной разработки онтологий, 2) рекомендации по выбору доступных средств для разработки семантической структуры карты знаний.*

Ключевые слова: представление знаний, онтологическое моделирование, карта знаний, визуальные редакторы

1 Введение

Карта знаний как инструмент, предназначенный для обеспечения навигации по источникам знаний, является чувствительной к способам её визуализации. Наиболее требовательной к различным аспектам визуализации является карта знаний, основанная на онтологии, т.к. визуализация необходима на всех этапах разработки, включая онтологическое моделирование. Онтологическое моделирование является сегодня стандартом de facto в области создания структуры знаний предметной области. В этой области за 30 лет сформировались несколько методологий, основанных на пионерских работах Грубера и Ушхольда [Gruber, 1993; Uschold, 1998]. Позднее они были доработаны до промышленных технологий настоящего времени [Holsapple & Joshi, 2002; Human, 2017; Patel & Jain, 2018; Krieg-Brückner, Mossakowski & Codescu, 2021]. Активно в этой области работают и российские исследователи [Боргест, 2018; Загорулько и Боровикова, 2019; Ворожцова, Массель, 2019]. Разработка онтологий представляет ядро современной инженерии знаний [Загорулько, 2020; Гаврилова, Кудрявцев, Муромцев, 2020].

Процесс практической разработки онтологий имеет несколько критических моментов (bottleneck problems):

a. Работа с источником знаний и их извлечение [Гаврилова, 2001; O'Hagan, 2019]

b. Структурирование знаний и формирование «скелета онтологии» [Peer, Brunec, Newcombe, and Epstein, 2021]

c. Практическая разработка и визуализация.

Ведущей практикой в картировании знаний является разработка онтологии совместно с экспертом и его дальнейшее активное участие в формировании иерархии и набора параметров. В

то же время, эксперт предметной области зачастую не обладает знаниями и навыками онтологического инжиниринга и нуждается в простых способах коммуникации с аналитиком. Таким образом, вопрос визуализации онтологии на этапе разработки базовых структур выходит на передний план.

2 Инструменты разработки онтологий

Данный доклад описывает предварительный подход к формированию рекомендаций по указанному критическому моменту разработки. Именно визуализация позволяет понятийно верифицировать онтологию с экспертами предметной области и использовать ее в дидактических целях. Имеется большой задел работ по анализу визуальных метафор, начатый пионерскими работами Эпплера и Сен-Галленской школы [Eppler, 2004-2008], и продолженный уже в последние годы [Гаврилова, Алсуфьев, Гринберг, 2017; Kydryavtsev, Gavrilova, 2017, Rostoka et al., 2020]. Практический дизайн связан с преодолением множества проблем концептуального и когнитивного характера, но встают также и чисто технологические вопросы программной реализации. На основании обзора и анализа более, чем 30 современных инструментов разработки онтологий (ОРИ), был сформирован список из 15 наиболее перспективных и доступных, а главное, работающих (см. Таблицу 1.). Многие известные ОРИ поменяли название, изменились функционально, или потеряли поддержку.

Представленная апробация и обзор литературы позволили выявить пять основных классов ОРИ:

1) Графовые СУБД

2) Онтологические редакторы: плагины для визуализации, разные нотации (uml, visualOWL)

3) Графовые платформы со встроенным редактором онтологий

4) Адаптеры (для наполнения онтологий)

5) Инструменты полуформального графического моделирования

Нетривиальной оказалась работа по соотнесению конкретного ОРИ с классом, так как некоторые классы размыты и включают в себя элементы других.

Таблица 1. Современные инструменты визуализации онтологий

Имя ОРИ	**ТИП ОРИ**	**адрес**	**Особенности**
AnzoGraph	Графовая СУБД	https://www.cambridgesemantics.com/anzograph/	
GraphDB	Графовая СУБД	https://graphdb.ontotext.com/	
OBO-Edit	Редактор	https://sourceforge.net/projects/geneontology/	
OntoStudio	Редактор	OntoStudio: https://ontostudio.semantic-web.com/	
OWLGrEd	Редактор	OWLGrEd: https://owlgred.lumii.lv/	онлайн, визуальное редактирование
PoolParty	Графовая платформа	https://www.poolparty.biz/	
Protégé (вкл. WebProtégé)	Редактор	https://protege.stanford.edu/	плагины для визуализации (VOWL, Ontograf)
Semantic Turkey	Редактор	http://semanticturkey.uniroma2.it/ https://vocbench.uniroma2.it/	платформа, вкл. VocBench
TopBraid Composer	Редактор	https://www.topquadrant.com/products/topbraid-composer/	визуализатор (навигация, редактирование)
Cytoscape	Платформа для визуализации	https://cytoscape.org/	

VOWL	Web-редактор	http://vowl.visualdataweb.org/	плагин для Protege со своей нотацией отображения онтологии
Graphol	Нотация для моделирования онтологий	https://www.mdpi.com/1999-5903/14/3/78	
YWORKS	Визуализатор онтологий	https://www.yworks.com/use-case/visualizing-an-ontology	
Onto4all	Редактор	https://onto4all.com/en	онлайн, визульное редактирование

3 Фрагмент онтологии для карты знаний

3.1 Подход к решению задачи

С целью выявления оптимальных средств визуальной разработки онтологии для карты знаний были выполнены следующие шаги:

1) выполнен обзор существующих средств визуальной разработки онтологий;

2) сформирован перечень характеристик, которыми обладает онтология карты знаний

3) разработан тестовый фрагмент фрагмента онтологии, соответствующий указанным характеристикам;

4) разработанный файл в формате *turtle* загружен на различные ресурсы и в программы, выявленные на первом шаге;

5) смоделирована ситуация изменения фрагмента онтологии экспертом.

3.2 Разработка тестового фрагмента онтологии

В целях обеспечения полноты представления онтологии был разработан тестовый фрагмент, содержащий кластер концептов сфокусированный вокруг понятия «Преподаватель» (*Professor*). За основу тестового фрагмента были взяты материалы из открытых источников, которые были доработаны для удовлетворения требованиям проекта.

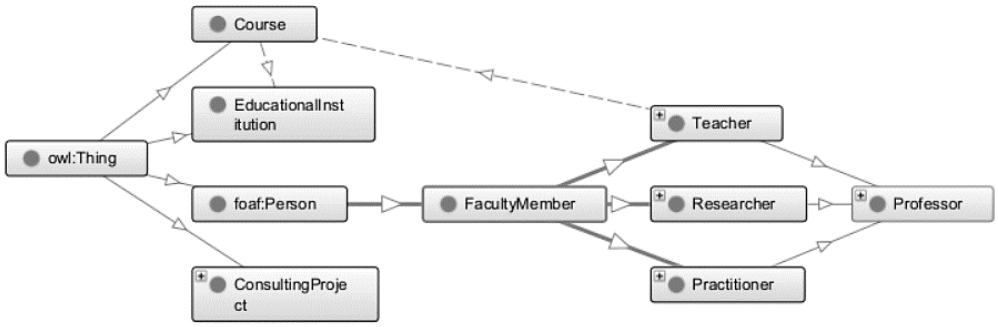


Рис.1. Три аспекта представления члена кафедры: преподаватель (Teacher), исследователь (Researcher), практик (Practitioner)

Разработанный фрагмент отвечает следующим требованиям:

1. Имеет четкий фокус: подробное представление одного человека АКО источника знаний
2. Отражает три различных подхода к анализу деятельности преподавателя: обучение, практико-ориентированная деятельность (роль – практик, *Practiotioner)* и исследование (Рис. 1). В качестве объекта для роли «практик» выбран на данном этапе исследования только один класс – консалтинговый проект (*ConsultingProject*).

3. Содержит ключевые иерархические типы связей: класс-подкласс (*subClassOf* - здесь и далее наименования связей представлены в нотации RDF), часть-целое (*partOf*),
4. Содержит функциональные типы связей (например *supervises* – рис.2), включая инверсивные (*isTaughtBy* inverse of *teaches*)
5. Содержит аксиомы классов, например *DisjointClasses*

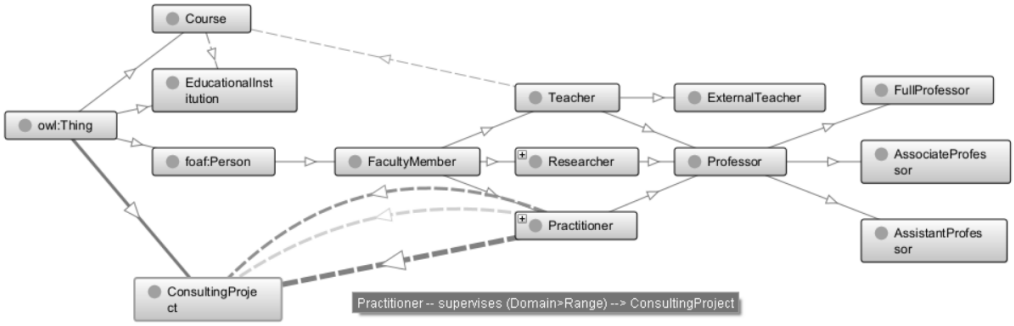


Рис.2. Иерархические (отмечены сплошной линией) и функциональные (пунктиром) типы связей

Далее представленный фрагмент онтологии был населен тестовыми экземплярами. Часть экземпляров было решено заносить кириллицей, чтобы проверить корректность отображения букв русского алфавита в онлайн-системах, как правило зарубежных. Другая часть занесена латиницей, чтобы обеспечить читабельность даже при сбоях в распознавании кириллицы.

4 Выбор средства визуальной разработки

Выбор инструмента разработки и визуализации зависит в основном от специфики требований к онтологии, квалификации коллектива разработчиков и доступности конкретного инструмента.Также к критериям выбора можно отнести:

- Возможность поддерживать различные стадии жизненного цикла онтологии,
- Доступность средств работы с разными уровнями детализации онтологии,
- Наличие масштабирования.

Наш собственный опыт разработки, интервью с экспертами и анализ литературы позволяют назвать лидеров из списка Таблицы 1 среди полновесных инструментов онтологического инжиниринга. Это PROTÉGÉ и TopBraid Composer (который является коммерческим продуктом). Именно редактор PROTÉGÉ был использован для создания тестового фрагмента. Это программный продукт первого выбора, т.к. распространяется на некоммерческой основе, сопровождается подробными инструкциями разных форматов и обладает самым широким набором плагинов.

Однако, как было указано ранее, для совестной разработки онтологии с экспертом, необходимо легковесное и эстетичное решение, позволяющее моделировать онтологию сразу в графическом формате. Этот критерий стал определяющим при выборе инструмента для разработки карты знаний.

Тестирование фрагмента карты знаний в различных средах позволил выявить два решения, которые на данный момент представляются оптимальными. Первый – OWLGrEd – это созданный более дести лет назад бесплатный графический редактор в стиле UML для онтологий OWL. Он имеет дополнительные функции для исследования и разработки графических онтологий, включая совместимость с Protege. Второе, достаточно молодое решение, Onto4all - это основанный на методологии OntoForInfoScience, редактор. Он предназначен для облегчения работы пользователей с различным уровнем опыта, разработчики стремятся с его помощью популяризировать построение онтологий. Указанные решения обладают следующими достоинствами:

- Не требуют обязательной установки на персональный компьютер

- Корректно работают с кириллицей
- Позволяют создавать с нуля (Рис. 3) или загружать готовые онтологии (Рис. 4), редактировать их и выгружать в нужном формате
    - Отображают различные типы связей
    - Позволяют создавать новые классы и свойства
    - Корректно и наглядно отображают инстансы (экземпляры, индивиды)

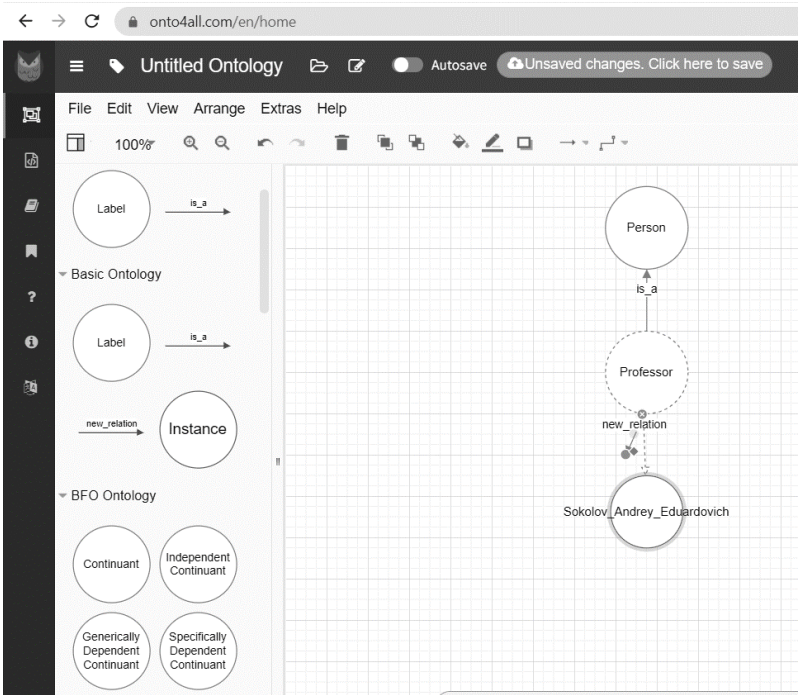


Рисунок 3. Пример создания онтологии с нуля в редакторе onto4All

- Интуитивно понятны
    - визуальное представление OWLGrEd ближе к отображению ERP-диаграммы и будет привычнее экспертам работающим с моделями данных (Рис. 4)
    - формат Onto4all приближен к инструменту c-map (Рис. 3) и может быть рекомендован для работы с носителями знаний, область экспертизы которых не затрагивает работу с базами данных

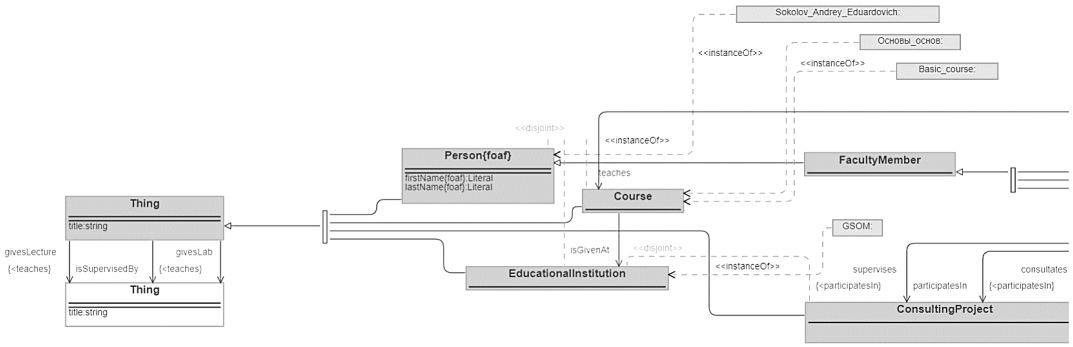


Рисунок 4. Фрагмент онтологии, представленный в онлайн редакторе OWLGrEd. Визуализация целиком в редактируемом формате доступна по ссылке http://owlgred.lumii.lv/online_visualization/oo1u

Гонка решений в области визуализации продолжается, интерес к онтологиям остается, и новые ОРИ появляются на рынке технологических новинок.

5 Благодарности

Работа выполнена при финансовой поддержке РНФ (проект № 23-21-00168)

Литература

[1] Боргест, Н. М. (2018). Онтологии проектирования от Витрувия до Виттиха. Онтология проектирования, 8(4 (30)), 487-522.
[2] Ворожцова, Т. Н., Макагонова, Н. Н., & Массель, Л. В. (2019). Онтологический подход к проектированию базы данных для оценки влияния энергетики на окружающую среду. Информационные и математические технологии в науке и управлении, (3 (15)).
[3] Гаврилова Т.А. Извлечение знаний: психологический аспект // Ж. «Корпоративные системы» (Enterprise Partner) N8 (25), 2001.- с.24-26.
[4] Гаврилова Т., Алсуфьев А., Гринберг Э. Визуализация знаний: критика Сент-Галленской школы и анализ современных трендов // Бизнес–информатика N3, 2017. – с. 7-19.
[5] Гаврилова Т.А., Кудрявцев Д.В., Муромцев Д.И. Инженерия знаний. Модели и методы. СПб, Лань, 2020.
[6] Загорулько, Ю. А., & Боровикова, О. И. (2019). Проблемы построения онтологий научных предметных областей на основе паттернов онтологического проектирования. In Информационные технологии и системы (pp. 157-161).
[7] Загорулько, Ю. А., & Загорулько, Г. Б. (2020). Искусственный интеллект. Инженерия знаний.
[8] Eppler M. 2004. Making knowledge visible through knowledge maps: concepts, elements, cases. In Handbook on Knowledge Management 1, 189-205. Springer, Berlin, Heidelberg.
[9] Eppler M. 2008. A process-based classification of knowledge maps and application examples. Knowledge and process Management, 15,1, 59-71.
[10] Gruber T. 1993. A translation approach to portable ontology specifications. Knowledge acquisition, 5, 2, 199-220.
[11] Holsapple, C. W., & Joshi, K. D. (2002). A collaborative approach to ontology design. Communications of the ACM, 45(2), 42-47.
[12] Human, S., Fahrenbach, F., Kragulj, F., & Savenkov, V. (2017, November). Ontology for representing human needs. In International Conference on Knowledge Engineering and the Semantic Web (pp. 195-210). Springer, Cham.
[13] Kudryavtsev, D., & Gavrilova, T. From anarchy to system: A novel classification of visual knowledge codification techniques // Knowledge and Process Management, 24(1), 2017, 3-13. doi:10.1002/kpm.1509.
[14] Krieg-Brückner, B., Mossakowski, T., & Codescu, M. (2021). Generic Ontology Design Patterns: Roles and Change Over Time. Advances in Pattern-Based Ontology Engineering, 51, In *WOP (Book)* . p.25-47.
[15] O'Hagan, A., 2019. Expert knowledge elicitation: subjective but scientific. The American Statistician, 73(sup1), pp.69-81.
[16] Patel A, Jain S. Formalisms of representing knowledge. Procedia Computer Science. 2018 Jan 1;125:542-549.
[17] Peer, M., Brunec, I.K., Newcombe, N.S. and Epstein, R.A., 2021. Structuring knowledge with cognitive maps and cognitive graphs. Trends in cognitive sciences, 25(1), pp.37-54.
[18] Rostoka, Marina, Andrii Guraliuk, Olha Kuzmenko, Tetiana Bondarenko, and Lyudmyla Petryshyn. "Ontological visualization of knowledge structures based on the operational management of information objects." In International Conference on Interactive Collaborative Learning, pp. 832-840. Cham: Springer International Publishing, 2020.
[19] Uschold, M., King, M., Moralee, S., & Zorgios, Y. (1998). The enterprise ontology. The knowledge engineering review, 13(1), 31-89.

О построении профилей в тематическом пространстве теории управления

Губанов Д.А, Кузнецов О.П., Суховеров В.С., Чхартишвили А.Г.

Институт проблем управления им. В. А. Трапезникова Российской академии наук, ул. Профсоюзная, д. 65, г. Москва, 117997, Россия

gubanov@ipu.ru, olpkuz@yandex.ru, vicst09@yandex.ru, sandro_ch@mail.ru

Аннотация. *Рассматривается задача классификации научных текстов в области теории управления. Классифицируемый текст располагается в тематическом пространстве, формируемом онтологией теории управления. Описывается структура онтологии. Вводится понятие профиля – стохастического вектора, позиционирующего объект классификации (публикацию, ученого и т. д.) в тематическом пространстве. Предложены алгоритмы построения профиля публикации и профиля ученого на основе его публикаций. Вводится расстояние между профилями. Приведены конкретные примеры профилей ученых, построенные с использованием базы публикаций сотрудников ИПУ РАН. Описанные методы реализуются в Информационной системе автоматизации научной деятельности (ИСАНД), которая разрабатывается в ИПУ РАН с начала этого года.*

Ключевые слова: классификация, онтология, публикация, профиль, вектор, тематическое пространство, тема, термин

1 Введение

Среди задач классификации неструктурированных текстов особое место занимает тематическая классификация научных текстов. Классификаторы научной тематики появились уже довольно давно: это существующий более 100 лет классификатор УДК [1], международный классификатор OECD [2], классификатор Российского научного фонда [3], ГРНТИ [4] и др.

В настоящее время актуальными стали задачи автоматизированного анализа научных публикаций (см., например, [5 - 8]) и поддержки принятия научно-организационных решений в научных учреждениях, редакциях журналов и других научных организациях (кому направить на рецензирование статью или заявку на грант; в какую рубрику журнала поместить отрецензированную статью; не принимать диссертацию к защите в данном диссертационном совете из-за несоответствия его тематике и т.д.), а также сопровождения повседневной работы ученых (например, подборка статей по заданной тематике [9]. Все возможные применения таких методов требуют прежде всего определения тематики научного текста и/или сферы компетентности ученого, иначе говоря, позиционирования объектов научной деятельности в тематическом пространстве науки. Координаты научного объекта (например, научной статьи или ученого) в этом пространстве в дальнейшем будем называть его *профилем*.

Автоматизация методов работы в тематическом пространстве предъявляет повышенные требования как к формальной структуре классификатора, формирующего пространство, так и к его содержанию. Этим требованиям упомянутые выше классификаторы не вполне соответствуют, в том числе в силу своей универсальности (нацеленности на удовлетворение потребностей всех научных направлений). Одним из перспективных путей решения возникающих проблем является разработка специализированных классификаторов, соответствующих современным стандартам построения онтологий.

Именно этот путь избран в Информационной системе автоматизации научной деятельности (ИСАНД), которая разрабатывается в ИПУ РАН с начала этого года. В ее основе лежит онтология теории управления, к описанию которой мы переходим.

2 Структура онтологии и понятие профиля публикации

Классификатором в системе ИСАНД является *онтология теории управления (ТУ)*.

Существенным отличием предлагаемого принципа классификации от известных является его многомерность. Все упомянутые выше классификаторы построены на основе таксономического принципа: каждый объект классификации должен характеризоваться ровно одной вершиной дерева классификатора. Наш специализированный классификатор является своего рода «системой координат» тематического пространства, позволяющей реализовать взгляд на совокупность научных направлений с определенной точки зрения, а также отразить возможную многотемность научного документа или многообразие компетенций ученого. Характеристикой объекта в этом пространстве является вектор, называемый *профилем*.

Онтология системы ИСАНД представляет собой существенно расширенную версию онтологии теории управления, предложенной в [10]. Она имеет четырехуровневую структуру, уровни которой (кроме нижнего) представляют собой дерево. Уровни нумеруются числами от 0 до 4.

Нулевой уровень содержит 3 фиксированных вершины «Математический аппарат», «Проблемная область», «Сфера применения». Предполагается, что при возможных расширениях онтологии этот уровень не изменяется. Он отражает не конкретные темы теории управления, а различные аспекты научных исследований: математический инструментарий, используемый в исследовании (теорию игр, теорию вероятностей, …), проблемную область, т.е. некоторую прикладную теорию (анализ данных, теорию управления в организационных системах, …) и конкретную сферу применения (энергетику, финансы, медицину, …). Поэтому вершины нулевого уровня называются *аспектами*.

Каждая из вершин нулевого уровня является корнем тематического поддерева, раскрывающего ее содержание. Например, поддерево «Математический аппарат» содержит вершину «Теория игр» (1-й уровень) и детализирующие ее вершины 2-го уровня: «Теория некооперативных игр» и др. Соответственно, поддерево «Проблемная область» среди прочих содержит вершину «Теория управления в организационных системах» и детализирующие ее вершины 2-го уровня, например, «Механизмы планирования», а поддерево «Сфера применения» содержит вершины «Энергетика» (1-й уровень) и «Атомная энергетика» (2-й уровень).

Множество вершин 1-го уровня, называемых *темами*, обозначим через $V = \{v_1, \ldots, v_n\}$. При этом i-я вершина 1-го уровня связана с множеством $V_i = \{v_{i1}, \ldots, v_{in_i}\}$ вершин 2-го уровня – *подтем*. Обозначим через m общее число подтем: $m = \sum_{i \in N} n_i$.

Третий уровень – это вершины-*термины*, характеризующие подтемы. Обозначим через $Q_{ij} = \{1, \ldots, q_{ij}\}$ множество терминов, характеризующих ij-ю подтему. Каждый термин, как правило, характеризует одну подтему.

Обозначим:

K - множество ученых;

L - множество публикаций;

Δ_{lij} - сумма числа вхождений в l-ю публикацию базовых терминов из ij-й подтемы;

$$\omega(k,\ l) = \begin{cases} 1, \text{ если } k\text{-й ученый является автором } l\text{-ой публикации;} \\ 0, \text{ в противном случае;} \end{cases}$$

$r(l)$ – количество авторов l-ой публикации.

Определим *профиль второго уровня публикации* l:

(1) $x_l = \left(x_{l1}, \ldots, x_{lij}, \ldots, x_{lnm}\right)$, где $x_{lij} = \dfrac{\Delta_{lij}}{\sum_{i \in N} \sum_{j \in N_i} \Delta_{lij}},\ l \in L,\ j \in N_i,\ i \in N.$

Очевидно, что этот вектор является стохастическим, т.е. $\sum_{i,j} x_{lij} = 1$.

Замечание. В дальнейшем возможно рассмотрение более сложных методов определения профиля (в том числе основанных на сетевых связях (ссылок) публикаций).

Для нахождения *профиля первого уровня публикации* l просуммируем для каждой темы значения компонент профиля второго уровня, отвечающих связанным с ним подтемам:

(2) $X_l = (X_{l1}, \ldots, X_{li}, \ldots, X_{ln})$, где $X_{li} = \sum_{j \in N_i} x_{lij},\ l \in L,\ i \in N.$

Наконец, для нахождения *профиля нулевого уровня публикации* просуммируем для каждой из трех вершин нулевого уровня значения компонент профиля первого уровня, отвечающих связанным с ней вершинам первого уровня.

Таким образом, каждая публикация характеризуется трехмерным вектором профиля нулевого уровня, *n*-мерным вектором профиля первого уровня, *m*-мерным вектором профиля второго уровня. Все три вектора являются стохастическими.

3 Расчет профилей ученого

На основе аддитивного принципа агрегирования можно определить *профили второго и первого уровня k*-го *ученого*, используя массив его публикаций

$$(3)\ y_{ij}^{k} = \frac{\sum_{l\in L}\omega(k,l)\frac{x_{lij}}{r(l)}}{\sum_{i\in N}\sum_{j\in N_i}\sum_{l\in L}\omega(k,l)\frac{x_{lij}}{r(l)}},\ \ k\in K,\ \ j\in N_i,\ \ i\in N,$$

$$(4)\ Y_{i}^{k} = \sum_{j\in N_i} y_{ij}^{k},\ \ k\in K,\ \ i\in N.$$

Профиль нулевого уровня определяется суммированием для каждой из трех вершин нулевого уровня значений компонент профиля первого уровня, отвечающих связанным с ней вершинам первого уровня.

4 Расстояние между профилями

Как было показано в предыдущих разделах, профили публикации, ученого, организации, журнала могут быть заданы в виде стохастических векторов. Для решения ряда задач требуется уметь оценивать степень близости этих векторов. Предлагается применять следующее расстояние между двумя профилями, задаваемыми стохастическими векторами $p = (p_1,\ \ldots,\ p_n)$ и $q = (q_1,\ \ldots,\ q_n)$:

$$d(p,q) = 1 - \sum_{j=1}^{n}\min(p_j,q_j) = \frac{1}{2}\sum_{j=1}^{n}|p_j - q_j|.$$

Отметим, что эта метрика является частным случаем хорошо известного в теории вероятностей расстояния общей вариации (см. также [11]) и принимает значения от 0 до 1 включительно. В справедливости второго равенства (для стохастических векторов) легко убедиться:

$$\frac{1}{2}\sum_{j=1}^{n}|p_j - q_j| = \frac{1}{2}\sum_{j=1}^{n}\left(\max(p_j,q_j) - \min(p_j,q_j)\right) = \frac{1}{2}\left(2 - \sum_{j=1}^{n}\min(p_j,q_j) - \sum_{j=1}^{n}\min(p_j,q_j)\right)$$
$$= 1 - \sum_{j=1}^{n}\min(p_j,q_j).$$

Замечание. Также легко убедиться, что в данной метрике расстояние между профилями первого уровня всегда не больше расстояния между профилями второго уровня и не меньше расстояния между профилями нулевого уровня тех же объектов. Действительно, пусть имеются два объекта, профили первого уровня которых X_1 и X_2, а профили второго уровня – x_1 и x_2. Обозначим через *d* и *D* расстояния между профилями первого и второго уровня соответственно. Тогда справедлива следующая цепочка соотношений:

$$d = \frac{1}{2}\sum_{i\in N}|X_{1i} - X_{2i}| = \frac{1}{2}\sum_{i\in N}\left|\sum_{j\in N_i}x_{1ij} - \sum_{j\in N_i}x_{1ij}\right| = \frac{1}{2}\sum_{i\in N}\left|\sum_{j\in N_i}(x_{1ij} - x_{2ij})\right| \le$$
$$\le \frac{1}{2}\sum_{i\in N}\sum_{j\in N_i}|x_{1ij} - x_{2ij}| = D.$$

Аналогично доказывается, что расстояние между профилями нулевого уровня не превосходит расстояние между профилями первого уровня.

5 Анализ профилей публикаций и ученых на примере публикаций в области теории управления (ТУ)

Онтология теории управления содержит 3 вершины 0-го уровня, 52 вершины 1-го уровня, 160 вершин 2-го уровня и около 3000 терминов. Рассчитаем на основе такой онтологии профили публикаций и ученых на примере корпуса аннотаций к русскоязычным научным публикациям сотрудников Института проблем управления им. В. А. Трапезникова РАН (БД ИПУ РАН) за 1921–2022 годы (https://www.ipu.ru/).

Общая схема построения профилей такова:

1) извлечение и предварительная обработка текстов (в данном случае – аннотаций);
2) расчет и анализ профилей публикаций;
3) расчет и анализ профилей авторов публикаций (ученых);

Предварительная обработка текстов производится при помощи разработанных методов автоматизированной обработки (см. подходы к обработке в [12] и [13]), в частности, выполняется токенизация, лемматизация, выявление *n*-грамм и преобразование текстов в векторное представление на основе терминов онтологии теории управления.

Расчет профилей 1-го уровня осуществляется при помощи формул (2) и (4), для ускорения вычислений разработаны векторные версии формул. Перед расчетом профилей ученых была выполнена работа по устранению ошибок и идентификации авторов публикаций.

Корпус ИПУ РАН насчитывает примерно 19,5 тыс. русскоязычных публикаций[1], для анализа использовались названия и аннотации публикаций.

Рассчитаем сначала профили 1-го уровня для этих публикаций. Для визуализации профили публикаций преобразуем в профили 0-го уровня (*Математический аппарат*, *Проблемная область*, *Сфера применения*). Распределение показано на тепловой карте симплекса профилей (см. рис. 1).

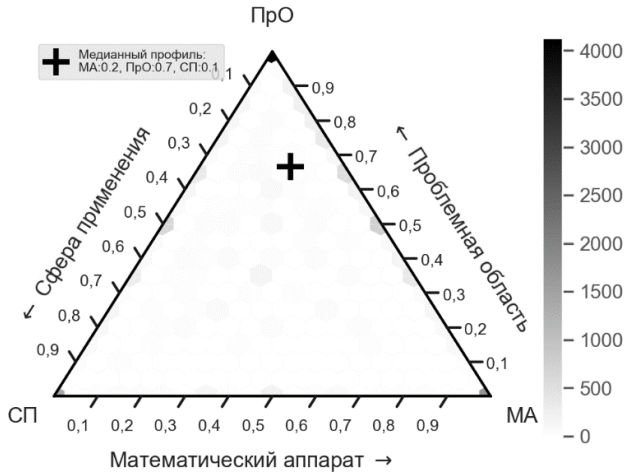


Рис.1. Тепловая карта профилей 0-го уровня публикаций БД ИПУ РАН

Черным плюсом на рис. 1 обозначен медианный[2] профиль (0,2; 0,7; 0,1) – точка, на которой достигается минимум суммы расстояний до всех рассматриваемых профилей в смысле введенного расстояния (см. обоснование и алгоритм расчета оптимальной точки в работе [14]). Таким образом в ТУ медианный профиль публикации (с точки зрения названий и аннотаций публикаций БД ИПУ РАН) «затрагивает» на 20 % математический аппарат, 70 % проблемную область и 10 % сферу применения.

После вычисления профилей публикаций вычисляются профили 1-го уровня для ученых – авторов публикаций (6,4 тыс.). Далее вычисляется расстояние между профилями ученых, затем для нелинейного снижения размерности пространства профилей применяется алгоритм машинного обучения UMAP [15], который стремится сохранить предварительно рассчитанное

[1] Авторы признательны А.В. Батову и А.М. Сальникову за описание и выгрузку корпуса публикаций ИПУ РАН.
[2] Отметим, что это одна из возможных многомерных медиан множества точек [16].

расстояние между точками в исходном пространстве как на локальном, так и на глобальном уровне (см. на рис. 2 визуализацию преобразованных профилей в двумерном пространстве).

Рис.2. Профили ученых в двумерном пространстве (центральный фрагмент)

На рис. 2 отмечены крестиками профили авторов данного доклада и некоторых других ученых ИПУ РАН (Губанов Д.А., Кузнецов О.П., Новиков Д.А., Поляк Б.Т., Суховеров В.С., Хлебников М.В., Чхартишвили А.Г.). Расположение этих профилей в двумерном пространстве согласуется с экспертным представлением о том, в каких областях теории управления работают данные ученые.

На рис. 3 представлены некоторые характерные профили 1-го уровня в усеченном виде: темы упорядочены по убыванию «внимания» (значения компоненты профиля); для каждого аспекта приведены только пять наиболее весомых тем.

1-й уровень	Значение
Теория игр	0,032
Теория графов	0,016
Теория вероятностей и мат	0,014
Теория множеств и отноше	0,008
Комбинаторика	0,005
Теория управления в органи	0,408
Теория выбора и принятия р	0,111
Кибернетика и системный а	0,064
Исследование операций	0,055
Информационная безопасно	0,035
Образование	0,070
Социальные системы	0,027
Социально-экономические с	0,011
Робототехника	0,011
Военное дело	0,008

(а) Новиков Д.А.

1-й уровень	Значение
Теория графов	0,085
Математическая логика	0,038
Дифференциальные и интег	0,017
Функциональный анализ	0,012
Теория алгоритмов и форма	0,012
Искусственный интеллект и	0,319
Теория управления в органи	0,133
Информационная безопасно	0,104
Теория выбора и принятия р	0,087
Теория автоматического уп	0,032
Социальные системы	0,050
Робототехника	0,011
Образование	0,008
Технологические процессы	0,005
Авиация	0,000

(б) Кузнецов О.П.

1-й уровень	Значение
Теория оптимизации	0,113
Информатика и теория инф	0,079
Дифференциальные и интег	0,054
Алгебра и теория чисел	0,012
Математическая логика	0,007
Теория автоматического уп	0,491
Кибернетика и системный а	0,058
Навигация и управление дв	0,042
Мехатроника	0,020
Теория выбора и принятия р	0,020
Образование	0,012
Робототехника	0,004
Космос	0,004
Социальные системы	0,002
Военное дело	0,000

(в) Поляк Б.Т.

Рис.3. Примеры профилей ученых 1-го уровня

6 Заключение

Приведенные примеры профилей (рис.3) говорят о том, что разрабатываемая система ИСАНД вполне адекватно оценивает компетенции ученых на основе анализа аннотаций их публикаций, а анализ аспектов (рис.1 и медианный профиль в нем) ожидаемо показывает, что

основные исследования ученых ИПУ РАН ведутся в области прикладных теорий. Расчет близости в тематическом пространстве (рис. 2) удобен для определения компетентных рецензентов статьи, для подбора команды наукоемкого проекта в области управления и многих других задач.

Литература

[1] ГОСТ 7.90-2007. Система стандартов по информации, библиотечному и издательскому делу. Универсальная десятичная классификация. Структура, правила введения и индексирования: издание официальное.

[2] Коды международной классификации (OECD) [Электронный ресурс]. URL: https://storage.tusur.ru/files/134958/kody_OECD.pdf (дата обращения: 14.06.2023).

[3] Классификатор Российского научного фонда (РНФ) [Электронный ресурс]. URL: https://rscf.ru/contests/classification/ (дата обращения: 14.06.2023).

[4] Государственный рубрикатор научно-технической информации [Электронный ресурс]. URL: https://grnti.ru/ (дата обращения: 14.06.2023).

[5] Печников А.А., Чебуков Д.Е. Структура графа цитирования журналов Math-Net.Ru // Научный сервис в сети Интернет: труды XXIII Всероссийской научной конференции (20-23 сентября 2021 г., онлайн). — М.: ИПМ, 2021. – С. 265-278.

[6] Yang S., Wang F. Visualizing information science: Author direct citation analysis in China and around the world // Journal of Informetrics, Volume 9, Issue 1, 2015. – P. 208-225.

[7] Бусыгина Т.В., Рыкова В.В. Наукометрический анализ научного направления «палеопочвоведение» // Труды ГПНТБ СО РАН, № 3, 2021. – С. 66-75.

[8] Li J., Goerlandt F., Reniers G. An overview of scientometric mapping for the safety science community: Methods, tools, and framework // Safety Science, Volume 134, 2021,105093.

[9] Крыжановская С.Ю., Власов А.В., Еремеев М.А., Воронцов К.В. Полуавтоматическая суммаризация тематических подборок научных публикаций: задачи и подходы / Тезисы докладов 20-й Всероссийской конференции с международным участием "Математические методы распознавания образов". Москва: Издательство Российская академия наук, 2021. С. 333-338.

[10] Кузнецов О.П., Суховеров В.С. Онтологический подход к оценке тематики научного текста // Онтология проектирования, 2016, т.6, №1, 55-66.

[11] Губанов Д.А., Петров И.В., Чхартишвили А.Г. Многомерная модель динамики мнений в социальных сетях: индексы поляризации // Проблемы управления. 2020. № 3. С. 26-33.

[12] Губанов Д.А., Новиков Д.А. Методы извлечения и анализа терминологических структур смежных предметных областей (на примере методологии) // Онтология проектирования. 2018. Т. 8, № 3 (29). С. 347-365.

[13] Новиков Д.А., Губанов Д.А., Макаренко А.В. Методы анализа терминологической структуры предметной области // Управление большими системами. 2013. Выпуск 43. С. 5-33.

[14] Бызов Л.Г., Губанов Д.А., Козицин И.В., Чхартишвили А.Г. Идеальный политик для социальной сети: подход к анализу идеологических предпочтений пользователей // Проблемы управления. 2020. №4. С. 15-26.

[15] McInnes L., Healy J., Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction // https://arxiv.org/abs/1802.03426, 2020.

[16] Small C.G. A survey of multidimensional medians // International Statistical Review. – № 58. – 1990. – P. 263–277.

Применение исчисления позитивно-образованных формул в задачах управления дискретно-событийными системами

Давыдов А.В. [1], Ларионов А.А. [1], Нагул Н.В.[1]

[1]*Институт динамики систем и теории управления им. В.М. Матросова СО РАН, ул. Лермонтова, д. 134, г. Иркутск, 665033, Россия.*

artem@icc.ru, bootfrost@zoho.com, sapling@icc.ru

Аннотация. *В статье представлены основные идеи применения автоматического доказательства теорем на основе исчисления позитивно-образованных формул (ПОФ) для решения задач теории супервизорного управления (ТСУ) логическими дискретно-событийными системами. Язык ПОФ представляет собой полный язык первого порядка, формулы которого построены на основе позитивных типовых кванторов. Разработанное для задач управления динамическими системами, исчисление ПОФ позволяет решать базовые задачи ТСУ, такие как проверка критерия существования супервизорного управления, автоматическая модификация ограничений на поведение управляемой системы, реализация супервизора, и др. Кроме того, свойства исчисления позволяют организовать дополнительную обработку событий в процессе вывода.*

Ключевые слова: позитивно-образованная формула, автоматическое доказательство теорем, прувер, дискретно-событийная система, супервизорное управление

1 Введение

После угасания интереса к автоматическому доказательству теорем (АДТ) в 1980-х-1990-х годах, в настоящее время предпринимаются многочисленные попытки повышения интеллектуальности систем управления на основе применения логического подхода к построению подсистем планирования и построения управления [1-3]. Такой подход связан с применением наиболее эффективных формализмов, лежащих в основе современных методов АДТ [4-6]. Однако ряд существенных препятствий не позволяют достичь необходимой эффективности программных систем АДТ для применения их в режиме реального времени, например: доказывающая программа делает слишком много заключений, большая часть которых избыточна или нерелевантна; программе приходится хранить слишком много информации в базе данных; правила вывода и шаги содержательного доказательства не совпадают по размеру; неадекватная фокусировка, программа легко сбивается на ложный путь поиска; необходимость метаправил для эффективного поиска доказательства.

В данной работе рассматривается логическое исчисление позитивно-образованных формул (ПОФ-исчисление) и построенный на его основе метод АДТ. Исчисление ПОФ было разработано в 1990-х годах ак. С.Н. Васильевым и А.К. Жерловым для решения задач управления динамическими системами [7, 8]. Язык ПОФ представляет собой полный язык первого порядка, формулы которого могут изображаться в виде деревьев: каждый узел дерева есть суть позитивный типовый квантор всеобщности или существования с условием на кванторные переменные в виде конъюнкции атомов (или тождественно истинного предиката True), а вдоль ветвей дерева типовые кванторы всеобщности и существования чередуются. ПОФ-исчисление выгодно отличается от возможностей других логических средств формализации предметной области и поиска логических выводов выразительностью в сочетании с компактностью представления знаний, «естественным» параллелизмом их обработки, крупноблочностью и

меньшей комбинаторной сложностью выводов, высокой совместимостью с эвристиками, а также большими возможностями для интерактивного доказательства (interactive proof-search). В выделенном классе формул возможно построение конструктивного доказательства. Данный класс формул существенно шире класса хорновских дизъюнктов, используемых в Прологе: на логическую формализацию аксиоматической базы предметной области не накладывается никаких ограничений, а целевое утверждение – это конъюнкция запросов в смысле языка Пролог.

Особенности исчисления позволяют совмещать автоматический поиск логических выводов со специальными эвристиками, настраиваемыми для каждой решаемой задачи. Кроме того, модифицируемость семантики исчисления, которая может быть конструктивной, немонотонной или временной, а также способность создавать интуиционистские выводы некоторых формул, не относящихся к хорновым, помогают при решении задач управления динамическими системами. Авторами исчисления решены несколько задач, для которых эффективно (с точки зрения выразительности языка и производительности средств доказательств теорем) использовать данный формализм, в том числе были решены задачи управления группой лифтов, планирования действий группы мобильных роботов, наведения телескопа на центр планеты в неполной фазе, и другие [5]. В этой статье рассматривается использование исчисления ПОФ для построения супервизорного управления дискретно-событийными системами (ДСС).

Дискретно-событийные модели, описывающие изменения дискретных состояний системы как результат возникновения дискретных событий, являются удобным способом представления основных аспектов функционирования технических систем, а также абстракцией некоторых непрерывных систем [9, 10]. В качестве событий рассматриваются изменения состояний системы, получение дополнительной информации о среде, и т.п. Существуют различные дискретно-событийные модели, самой популярной из которых является конечный автомат. Для ДСС в автоматной форме разработана теория супервизорного управления (ТСУ), основной задачей которой является обеспечение выполнения ограничений на поведение системы, или спецификации, путем запрета некоторого подмножества возможных событий, реализуемых системой и рассматриваемых как управляемые. Последовательности событий, возникающих в системе, рассматриваются как слова формального языка, свойства которого анализируются для построения и реализации управления. Современное состояние ТСУ представлено в [11, 12]. Операции над конечными автоматами, такие как построение параллельной композиции, позволяют моделировать взаимодействие различных компонент системы или нескольких систем. В случае изменения условий функционирования роботов вследствие внешних или внутренних факторов, парадигма ТСУ позволяет изменить спецификацию для отражения новых требований к системе, не меняя всей структуры системы. Изменения спецификации требуют построения нового управления для обеспечения адаптации к новым условиям функционирования.

Применение ПОФ-исчисления на данный момент позволило решить такие базовые задачи ТСУ, как проверка управляемости спецификации [13], построение наибольшего управляемого подъязыка заданного языка спецификации [14] и реализация монолитного супервизора [15]. В настоящей статье описаны основные элементы предлагаемого подхода, в том числе принцип использования обработки событий для охвата знаний об окружающей среде. Разрабатываемый подход будет использован для создания многоуровневых иерархических систем управления мобильными роботами и группами роботов [16].

2 Предварительные сведения

2.1 Исчисление позитивно-образованных формул

Рассмотрим [8] язык первопорядковых логических формул (ПЛФ), состоящий из формул, построенных из атомарных с помощью связок $\vee, \&, \neg, \rightarrow, \leftrightarrow$, кванторов $\forall$ и $\exists$, констант *true* и *false*. Понятия терм, атом, литера определяются обычным способом. В данном параграфе, неатомарные формулы и подформулы будут обозначаться большими полужирными буквами ($\boldsymbol{F}$, $\boldsymbol{P}$, $\boldsymbol{Q}$ и т. д.), возможно с индексами. Множества формул будут обозначаться большими греческими буквами (Φ, Ψ и т. д.), возможно с индексами.

Пусть $X = \{x_1, \dots, x_k\}$ - множество переменных, $A = \{A_1, \dots, A_m\}$ - множество атомарных формул, называемых *конъюнктами*, и $\Phi = \{\boldsymbol{F}_1, \dots, \boldsymbol{F}_n\}$ - множество первопорядковых логических формул. Следующие формулы $\forall x_1 \dots \forall x_k (A_1 \& \dots \& A_m) \rightarrow (\boldsymbol{F}_1 \vee \dots \vee \boldsymbol{F}_n)$ и $\exists x_1 \dots \exists x_k (A_1 \& \dots \& A_m) \& (\boldsymbol{F}_1 \& \dots \& \boldsymbol{F}_n)$ обозначаются как $\forall x_1, \dots, x_k A_1, \dots, A_m \{\boldsymbol{F}_1, \dots, \boldsymbol{F}_n\}$ и $\exists x_1, \dots, x_k A_1, \dots, A_m \{\boldsymbol{F}_1, \dots, \boldsymbol{F}_n\}$ соответственно. Последние могут быть сокращены следующим образом: $\forall_X A\ \Phi$ и $\exists_X A\ \Phi$ соответственно, учитывая при этом, что $\forall$-квантор соответствует $\rightarrow \Phi^{\vee}$, где $\Phi^{\vee}$ обозначает дизъюнкцию всех формул из Φ, а $\exists$-квантор соответствует $\& \Phi^{\&}$, где $\Phi^{\&}$

обозначает конъюнкцию всех формул из Φ. Любое из множеств X, A, Φ может быть пустым, в этом случае они могут быть опущены при записи формулы. Таким образом, если $Q \in \{\forall, \exists\}$, то $Q_X A\ \emptyset \equiv Q_X A$, $Q_X \emptyset\ \Phi \equiv Q_X\ \Phi$ и $Q_\emptyset A\ \Phi \equiv QA\ \Phi$. Поскольку пустая дизъюнкция идентична $false$, в то время как пустая конъюнкция идентична $true$, то корректны следующие эквивалентности: $\forall_X A\ \emptyset \equiv \forall_X A \to false \equiv \forall_X A$ и $\exists_X A\ \emptyset \equiv \exists_X A \& true \equiv \exists_X A$ и $\forall \emptyset\ \Phi \equiv true \to \Phi \equiv \forall\ \Phi$ и $\exists \emptyset\ \Phi \equiv true \& \Phi \equiv \exists\ \Phi$. В некоторых сокращениях формул нет никаких переменных при кванторах $\forall$ и $\exists$, что довольно необычно. Такие кванторы будем называть *фиктивными*, они присутствуют в структуре формулы для того чтобы правильно интерпретировать множество ПЛФ Φ. Если $\boldsymbol{P} = \forall A\ \Phi$, то мы знаем, что в этом случае интерпретируем формулу $\boldsymbol{P}$ как $A \to \Phi^{\vee}$. И если $\boldsymbol{P} = \exists A\ \Phi$, то интерпретируем $\boldsymbol{P}$ как $A \& \Phi^{\&}$. На практике фразы вида $\forall_X A\ \Phi$ и $\exists_X A\ \Phi$ означают следующие выражения на естественном языке: «для всех X, удовлетворяющих A, справедливо Φ»; «существуют X, удовлетворяющие свойству A, такие, что Φ»; «для всех целых $x, y, z \neq 0$ и $n > 2$ справедливо $x^n + y^n \neq z^n$». Первоначально конструкции $\forall_X A$ и $\exists_X A$ назывались *ти'повыми кванторами*, они были введены Н. Бурбаки как часть нотации для формализации математики.

Определение 1. Пусть X - множество переменных, A - конъюнкт, оба могут быть пустыми.

1. $\exists_X A$ называется ∃-формулой.
2. $\forall_X A$ называется ∀-формулой.
3. Если $\Phi = \{\boldsymbol{F}_1, \dots, \boldsymbol{F}_n\}$ - множество ∀-формул, тогда $\exists_X A\ \Phi$ называется ∃-формулой.
4. Если $\Phi = \{\boldsymbol{F}_1, \dots, \boldsymbol{F}_n\}$ - множество ∃-формул, тогда $\forall_X A\ \Phi$ называется ∀-формулой.
5. Любая ∃-формула или ∀-формула будет называться ПОФ.
6. Других ПОФ не существует.

Определенные формулы называются «позитивными» потому, что не содержат логического отрицания (¬). Если ПОФ имеет вид $\forall\ \Phi$, то такая формула называется *канонической ПОФ*. Очевидно, что любая ПОФ может быть преобразована к каноническую виду. Если $\boldsymbol{P}$ неканоническая ∃-ПОФ, то $\boldsymbol{P} \equiv \forall\ \boldsymbol{P}$, так как $\forall\ \boldsymbol{P} \equiv true \to \boldsymbol{P} \equiv \boldsymbol{P}$. Если $\boldsymbol{P}$ неканоническая ∀-ПОФ, то $\boldsymbol{P} \equiv \forall\ \{\exists\ \boldsymbol{P}\}$, так как $\forall\ \{\exists\ \boldsymbol{P}\} \equiv true \to (true \& \boldsymbol{P}) \equiv \boldsymbol{P}$. Для удобства чтения мы будем представлять ПОФ в виде дерева, узлами которого являются ти'повые кванторы. Относительно частей ПОФ будет использоваться соответствующая терминология: *узел, корень, лист, ветвь*. Например, ПОФ

$$\forall\ \{\exists_{X_1} A_1\ \{\forall_{Y_1} B_1, \forall B_2\ \{\exists_{X_2} A_2, \exists A_3\ \{\forall_{Y_2} B_3\}, \exists A_4\}\}, \exists A_5\ \{\forall_{Y_3} B_4\}\}$$

представляется в виде дерева следующим образом:

$$\forall \begin{cases} \exists_{X_1} A_1 \begin{cases} \forall_{Y_1} B_1 \\ \forall B_2 \begin{cases} \exists_{X_2} A_2 \\ \exists A_3 - \forall_{Y_2} B_3 \\ \exists A_4 \end{cases} \end{cases} \\ \exists A_5 - \forall_{Y_3} B_4 \end{cases}$$

В формуле $\boldsymbol{P} = \forall\{\boldsymbol{F}_1, \dots, \boldsymbol{F}_n\}$, $\boldsymbol{F}_i = \exists_{X_i} B_i\{\boldsymbol{Q}_{i1}, \dots, \boldsymbol{Q}_{im}\}$, $i = \overline{1,n}$, называются *базовыми подформулами* P, B_i называются *базами фактов* или просто *базами*, $\boldsymbol{Q}_{ij}$ называются *подформулами-вопросами*, а корни подформул-вопросов - *вопросами* к базам B_i, $i = \overline{1,n}$. Вопрос вида $\forall_X A$ (без каких-либо потомков) называется *целевым вопросом*.

Внутри каждой базовой подформулы любая переменная не может быть свободной и связанной одновременно, кроме того, она не может быть одновременно связана различными кванторами.

В процессе построения умозаключений, часто доказывают утверждение F, опровергая его отрицание. Описываемый метод построен аналогичным образом, и обрабатывает только канонические ПОФ (если не оговорено иначе).

Определение 2[Ответ] Рассмотрим некоторую базовую подформулу $\exists_X A\ \Psi$. Будем говорить, что вопрос подформулы $\boldsymbol{Q} = \forall_Y B\ \Phi$, $Q \in \Psi$ имеет ответ θ тогда и только тогда, когда θ является подстановкой $Y \to H^{\infty} \cup X$ и $B\theta \subseteq A$, где H^{∞} - эрбрановский универсум, основанный на константах и функциональных символах, встречающихся в соответствующей базовой подформуле.

Определение 3. Пусть $\boldsymbol{P}_1 = \exists_X A\ \Psi$ и $\boldsymbol{P}_2 = \exists_Y B\ \Phi$, тогда $merge(\boldsymbol{P}_1, \boldsymbol{P}_2) = \exists_{X \cup Y} A \cup B\ \Psi \cup \Phi$.

Определение 4. Рассмотрим некоторую базовую подформулу $\boldsymbol{B} = \exists_X A\ \Psi$. Если подформула-вопрос $\boldsymbol{Q} \in \Psi$ имеет вид $\forall_Y D\ \{\boldsymbol{P}_1, \dots, \boldsymbol{P}_n\}$, где $\boldsymbol{P}_i = \exists_{Z_i} C_i\ \Gamma_i, i = \overline{1,n}$, тогда $split(\boldsymbol{B}, \boldsymbol{Q}) = \{merge(\boldsymbol{B}, \boldsymbol{P'}_1), \dots, merge(\boldsymbol{B}, \boldsymbol{P'}_n)\}$, где $'$ - оператор переименования переменных. Будем говорить, что $\boldsymbol{Q}$ расщепляет $\boldsymbol{B}$, а $split(\boldsymbol{B}, \boldsymbol{Q})$ - результат расщепления $\boldsymbol{B}$. Очевидно, $split(\boldsymbol{B}, \forall_Y D) = \emptyset$.

Определение 5 [Правило вывода ω]. Рассмотрим некоторую формулу $\boldsymbol{F} = \forall\ \Phi$. Если существует базовая подформула $\boldsymbol{B} = \exists_X A\ \Psi$, $B \in \Phi$ и подформула вопрос $\boldsymbol{Q} \in \Psi$, и некоторый вопрос в $\boldsymbol{Q}$ имеет ответ θ, тогда $\omega(\boldsymbol{F}) = \forall\ \Phi \backslash \{\boldsymbol{B}\} \cup split(\boldsymbol{B}, \boldsymbol{Q}\theta)$.

Отметим, что когда множество Φ становится пустым после применения правила ω и ПОФ приобретает вид просто $\forall$, то можно сделать вывод, что формула F является противоречивой.

Любая конечная последовательность формул $\boldsymbol{F}, \omega\boldsymbol{F}, \omega^2\boldsymbol{F}, \dots, \omega^n\boldsymbol{F}$, где $\omega^s\boldsymbol{F} = \omega(\omega^{s-1}\boldsymbol{F}), \omega^1 = \omega, \omega^n\boldsymbol{F} = \forall$, называется *выводом* $\boldsymbol{F}$ в исчислении ПОФ (с аксиомой $\forall$).

Предположим, что стратегия поиска проверяет вопросы в последовательном цикле без пропусков и не использует повторное применение ω к вопросу с тем же самым θ (вопросно-ответная процедура поиска вывода).

2.2 Дискретно-событийные системы

Логическая дискретно-событийная система как генератор регулярного языка описывается пятеркой $G = (Q, \Sigma, \delta, q_0, Q_m)$, где Q – множество состояний q; Σ – множество событий; $\delta: \Sigma \times Q \to Q$ – функция переходов; $q_0 \in Q$ - начальное состояние; $Q_m \subset Q$ - множество маркированных состояний. Пусть Σ^* - замыкание Клини, т.е. множество всех строк над Σ, включая пустую строку ε. Множество $L(G) = \{w: w \in \Sigma^*$ и $\delta(w, q_0)$ определено$\}$ представляет собой язык, генерируемый G, а множество $L_m(G) = \{w: w \in L(G)$ и $\delta(w, q_0) \in Q_m\}$ - язык, маркированный G. Маркированный язык используется для описания завершенных системой задач, например, миссии автономного аппарата или обследовательского цикла. Заметим, что в отличие от классического конечного автомата, множество Q_m не имеет терминального характера.

Пусть $\Sigma_c \subseteq \Sigma$ будет множеством управляемых событий, $\Sigma_{uc} = \Sigma \backslash \Sigma_c$, $\Sigma_c \cap \Sigma_{uc} = \emptyset$. Супервизор как средство управления ДСС способен препятствовать возникновению управляемых событий с целью ограничить поведение системы в некоторых заданных рамках, определяемых регулярным языком, например, K, генерируемым автоматом $\mathcal{H}$. Обозначим $L(J/G)$ язык, генерируемый ДСС G под управлением супервизора. Пусть $L_m(J/G)$ обозначает язык, маркированный супервизором: $L_m(J/G) = L(J/G) \cap L_m(G)$. Обозначим через $\overline{K}$ множество всех префиксов слов из K. Основная задача супервизорного управления заключается в построении такого супервизора, что $L(J/G) = \overline{K}$. Известно, что условием существования решения этой задачи является управляемость K. Язык K управляем (относительно $L(G)$ и Σ_{uc}) если $\overline{K}\Sigma_{uc} \cap L(G) \subseteq \overline{K}$. Таким образом, для построения управления необходимо проводить проверку управляемости спецификации K как желаемых ограничений на функционирование ДСС. Как правило, природа таких ограничений вытекает из внешней обстановки или изменении структуры системы. Перемены обстановки или состояния системы приводят к необходимости адаптации к ним путем изменения спецификации. При этом вновь возникает задача проверки управляемости, и, в случае ее отсутствия, построения максимальной выполнимой спецификации на основе исходной. Для решения этой и других задач ТСУ предлагается использовать автоматическое доказательство теорем в исчислении ПОФ.

Общая форма ПОФ $\mathcal{F}_\mathcal{G}$, представляющей генератор $\mathcal{G}$, имеет базу $B_\mathcal{G} = \{L(\varepsilon, q_0)$, $L^m(\varepsilon, q_0)$, $\delta(q_1^i, \sigma^i, q_2^i)$, $\delta^m(q_1^j, \sigma^j, q_2^j)\}$, $i, j \in \{1, \dots, n\}$, где n – число событий, и два вопроса, см. (Рис. 1). После каждого применения единственного правила вывода исчисления ПОФ к этой формуле в базу формулы добавляются атомы вида $L(s, q)$ или $L^m(s, \mathrm{q})$, где первые аргументы представляют собой слова языков $L(G)$ и $L_m(G)$.

$$\mathcal{F}_{\mathcal{G}} = \exists B \begin{cases} \forall \sigma, q, \sigma', q'\ L(\sigma, q), \delta(q, \sigma', q') - \exists L(\sigma \cdot \sigma', q') \\ \forall \sigma, q, \sigma', q' L(\sigma, q), \delta^m(q, \sigma', q') - \exists L^m(\sigma \cdot \sigma', q') \end{cases}$$

Рис.1. Общая форма ПОФ-представления ДСС как генератора регулярного языка.

2.3 Примеры использования ПОФ

Как правило, ДСС, моделирующие сложные системы, состоят или могут быть представлены как набор подсистем, объединенных с помощью различных операции над конечными автоматами, прежде всего, произведения и параллельной композиции [10, 12]. Для иллюстрации применимости ПОФ для построения конструкций над множеством генераторов, приведем пример построения автомата, представляющего собой параллельное произведение $G_1||G_2$ двух автоматов G_1 и G_2 (Рис. 2).

$$F_{\mathcal{G}_1 \| \mathcal{G}_2} = \begin{matrix} \exists \delta_3(\phi, \varepsilon, q_0^1 \cdot q_0^2), \\ \delta_1(q_1^i, \sigma^i, q_2^i), \\ \delta_2(q_1^k, \sigma^k, q_2^k) \end{matrix} \begin{cases} \forall \sigma, q_1^1, q_1^2, q_2^1, q_2^2\ \ \delta_1(q_1^1, \sigma, q_2^1), \delta_2(q_1^2, \sigma, q_2^2), \\ \qquad\qquad \delta_3(_, _, q_1^1 \cdot q_1^2) \end{cases} \exists \delta_3(q_1^1 \cdot q_1^2, \sigma, q_2^1 \cdot q_2^2)$$
$$\begin{cases} \forall \sigma, q_1^1, q_1^2, q_2^1, q_2^2\ \ \delta_1(q_1^1, \sigma, q_2^1), \sigma \in \Sigma_1, \sigma \notin \Sigma_2, \\ \qquad\qquad \delta_3(_, _, q_1^1 \cdot q_1^2) \end{cases} \exists \delta_3(q_1^1 \cdot q_1^2, \sigma, q_2^1 \cdot q_1^2)$$
$$\begin{cases} \forall \sigma, q_1^1, q_1^2, q_2^1, q_2^2\ \ \delta_2(q_1^2, \sigma, q_2^2), \sigma \notin \Sigma_1, \sigma \in \Sigma_2, \\ \qquad\qquad \delta_3(_, _, q_1^1 \cdot q_1^2) \end{cases} \exists \delta_3(q_1^1 \cdot q_1^2, \sigma, q_1^1 \cdot q_2^2)$$

Рис.2. Общая форма ПОФ-представления ДСС как генератора регулярного языка.

ПОФ $F_{\mathcal{G}_1 \| \mathcal{G}_2}$ состоит из одной базовой подформулы, база которой $B_{\mathcal{G}_1 \| \mathcal{G}_2} = \{\delta_1(q_1^i, \sigma^i, q_2^i), \delta_2(q_1^k, \sigma^k, q_2^k),\ \delta_3(\phi, \varepsilon, q_0^1 \cdot q_0^2)\}$ содержит атомы для переходов δ_1 автомата $\mathcal{G}_1$, $i = \overline{1, n_1}$, и переходов δ_2 автомата $\mathcal{G}_2$, $k = \overline{1, n_2}$, а также атом $\delta_3(\phi, \varepsilon, q_0^1 \cdot q_0^2)$, соответствующий автомату параллельной композиции. Он определяет начальное состояние композиционного автомата как комбинацию начальных состояний q_0^1, q_0^2 автоматов $\mathcal{G}_1$ и $\mathcal{G}_2$. Преимущество метода построения параллельной композиции на основе ПОФ состоит в том, что композиционный автомат после завершения вывода не имеет недоступных состояний, поэтому нет необходимости использовать операцию удаления таких состояний. Это обеспечивается контролем связности графа, представляющего автомат, на каждом шаге построения вывода, реализованным с помощью атомов $\delta_3(_, _, x)$.

Произведение n генераторов может быть получено с помощью ПОФ $\mathcal{F}_{\mathcal{G}_1 \times \mathcal{G}_2 \times \ldots \times \mathcal{G}_n}$ на Рис. 3. Здесь атомы $C(_, \ldots, _)$ обеспечивают связность графа, представляющего параллельную композицию:

$$\mathcal{F}_{\mathcal{G}_1 \times \mathcal{G}_2 \times \ldots \times \mathcal{G}_n} = \begin{matrix} \exists C(q_0^1, \ldots, q_0^n), \\ \delta_1(q_1^{i_1}, \sigma_{i_1}, q_2^{i_1}), \ldots, \\ \ldots, \delta_n(q_1^{i_n}, \sigma_{i_n}, q_2^{i_n}) \end{matrix} - \begin{matrix} \forall \sigma, q_1^1, \ldots, q_1^n, q_2^1, \ldots, q_2^n \\ \delta_1(q_1^1, \sigma, q_2^1), \ldots, \delta_n(q_1^n, \sigma, q_2^n), \\ C(q_1^1, \ldots, q_1^n) \end{matrix} - \begin{matrix} \exists \delta_{n+1}(q_1^1 \cdot_{\ldots} \cdot q_1^n, \sigma, q_2^1 \cdot_{\ldots} \cdot q_2^n), \\ C(q_2^1, \ldots, q_2^n) \end{matrix}$$

Рис.3. ПОФ, строящая произведение n автоматов.

Если вовлеченные автоматы некоторым образом модифицированы, то произведение автоматов дает результат, эквивалентный параллельной композиции исходных автоматов. Модификация состоит в добавлении в состояния каждого автомата петель по событиям, уникальным для этого автомата, т.е. не входящих в множества событий других автоматов. Затем ПОФ ($\mathcal{F}_{\mathcal{G}_1 \times \mathcal{G}_2 \times \ldots \times \mathcal{G}_n}$) для модифицированных автоматов $\mathcal{G}'_1$, ..., $\mathcal{G}'_n$ дает результат, эквивалентный параллельной композиции исходных автоматов $\mathcal{G}_1$, ..., $\mathcal{G}_n$. Использование операции произведения для построения параллельную композицию автоматов значительно уменьшает вычислительную сложность этой процедуры. Для добавления вспомогательных петель воспользуемся вспомогательными ПОФ. При этом будем использовать одну из существенных особенностей исчисления ПОФ, которая состоит в том, что мы можем построить немонотонный вывод, лишь слегка изменив определение правила вывода. Для этого введем оператор *, который будет помечать атомы в вопросах. Теперь, если на вопрос с атомами, помеченными оператором *, есть ответ, то после применения правила вывода атомы в базе, участвовавшие в поиске соответствия с помеченными атомами, должны быть удалены из базы. В общем случае оператор * влияет на свойство полноты исчисления ПОФ, но для рассматриваемой в статье задачи благодаря

правильной формализации вывод всегда заканчивается. Для любых двух автоматов $\mathcal{A}_1$ и $\mathcal{A}_2$ ПОФ $\mathcal{F}_{ComEvents}$ удаляет общие события из базы $B_{\mathcal{A}_1,\mathcal{A}_2} = B_{\mathcal{A}_1} \cup B_{,\mathcal{A}_2}$:

$$\mathcal{F}_{ComEvents} = \exists\ B_{\mathcal{A}_1,\mathcal{A}_2} - \forall\sigma\ \Sigma^{\mathcal{A}_1}(\sigma)^*, \Sigma^{\mathcal{A}_2}(\sigma)^* - \exists\ \Sigma^{\mathcal{A}_1 \cap \mathcal{A}_2}(\sigma)$$

Рис.4. Получение общих событий автоматов $\mathcal{A}_1$ и $\mathcal{A}_2$.

Оставшиеся в базе события можно использовать для добавления вспомогательных петель с помощью ПОФ $\mathcal{F}_{loops}$ на Рис 5.

$$\mathcal{F}_{loops} = \exists\ B_{loops} \begin{cases} \forall\sigma, q\Sigma^{S_2}(\sigma), Q^{S_1}(q) \exists\delta^{S_1}(q,\sigma,q) \\ \forall\sigma, q\Sigma^{S_1}(\sigma), Q^{S_2}(q) \exists\delta^{S_2}(q,\sigma,q) \end{cases}$$

Рис.5. ПОФ, одновременно добавляющая петли по уникальным событиям в автоматы $\mathcal{A}_1$ и $\mathcal{A}_2$.

Пусть ПОФ на Рис. 3 теперь содержит в базе полученные на предыдущем этапе события $\delta_1\left(q^{j_1}, \sigma_{k_1}, q^{j_1}\right)$, …, $\delta_n\left(q^{j_n}, \sigma_{k_n}, q^{j_n}\right)$. Тогда формула на Рис. 3 строит параллельную композицию n автоматов через операцию произведения.

3 Задачи теории супервизорного управления ДСС

3.1 Проверка критерия существования супервизорного управления

С помощью логического вывода в исчислении ПОФ на основе автоматов $\mathcal{G}$ и $\mathcal{H}$, генерирующих $L(\mathcal{G})$ и K, может быть построен наибольший управляемый подъязык $K^{\uparrow C}$ спецификации K во время процедуры проверки ее управляемости [13, 14]. Для проверки управляемости используется ПОФ

$$\mathcal{F}_{Sub} = \exists B_{Sub} \begin{cases} Q_1 \\ Q_2, \\ \dots \\ Q_7 \end{cases}$$

где база B_{Sub} содержит предикатное описание переходов $\mathcal{G}$, $\mathcal{H}$ и $\mathcal{G} \times \mathcal{H}$, а вопросы к базе показаны на Рис. 6.

$$Q_1\colon\ \forall q_1^1, q_2^1, q_1^3, q_2^3, \sigma\ Q_0^1(q_1^1), Q_0^3(q_1^3), \delta_1(q_1^1, \sigma, q_2^1), \delta_3(q_1^3, \sigma, q_2^3) - \exists\ N(q_1^1, q_1^3)$$

$$Q_2\colon\ \forall q_1^1, q_2^1, q^3, \sigma\ N(q_1^1, q^3), \delta_1(q_1^1, \sigma, q_2^1), E_{uc}(\sigma) - \exists\ Chk(q^3, \sigma, 0)$$

$$Q_3\colon\ \forall q_1^3, q_2^3, \sigma\ Chk^*(q_1^3, \sigma, 0), \delta_3(q_1^3, \sigma, q_2^3) - \exists\ Chk(q_1^3, \sigma, 1)$$

$$Q_4\colon\ \forall q^3, \sigma\ Chk(q^3, \sigma, 0) - \exists\ Del(q^3)$$

$$Q_5\colon\ \forall q_1^1, q_2^1, q_1^3, q_2^3, \sigma\ N(q_1^1, q_1^3), \delta_1(q_1^1, \sigma, q_2^1), \delta_3(q_1^3, \sigma, q_2^3) - \exists\ N(q_2^1, q_2^3)$$

$$Q_6\colon\ \forall q_{11}, q_{12}, q_{21}, q_{22}, \sigma\ Del(q_{11} \cdot q_{12}), \delta_2^*(q_{12}, \sigma, q_{22}), \delta_3^*(q_{11} \cdot q_{12}, \sigma, q_{21} \cdot q_{22}) - \exists\ Deleted(q_{12}, \sigma, q_{22})$$

$$Q_7\colon\ \forall q_{11}, q_{12}, q_{21}, q_{22}, \sigma\ Del(q_{21} \cdot q_{22}), \delta_2^*(q_{12}, \sigma, q_{22}), \delta_3^*(q_{11} \cdot q_{12}, \sigma, q_{21} \cdot q_{22}) - \exists\ Deleted(q_{12}, \sigma, q_{22})$$

Рис.6. ПОФ для проверки управляемости регулярного языка и построения наибольшего управляемого подъязыка.

Чтобы построить наибольший управляемый подъязык в K, мы используем следующую стратегию. Вывод строится путем проверки применимости правила вывода к вопросам в порядке от Q_1 до Q_7. Правила Q_1 - Q_5 проверяют, является ли спецификация управляемой или нет. Правила Q_6, Q_7 удаляют переходы, связанные с состоянием, в котором происходит событие, нарушающее условие управляемости. Если при применении правил Q_1 - Q_5 был найден ответ на вопрос Q_4, то в базу добавляется атом $Del()$, обозначающий неуправляемость K. Затем мы применяем правила Q_6, Q_7, игнорируя правила из Q_1 - Q_5, пока возможны подстановки для применения правила вывода к Q_6 и Q_7. Далее процесс повторяется до тех пор, пока вывод не остановится, исчерпав

все возможные замены. Заметим, что данный подход позволяет рассматривать в том числе языки, не являющиеся префиксно-замкнутыми.

3.2 Реализация супервизорного управления

В случае, если $K^{\uparrow C} \neq \emptyset$, на основе построенного языка $K^{\uparrow C}$, принятого за новую спецификацию, может быть построен супервизор, реализующий построенные ограничения. В качестве такого супервизора J может быть использован автомат S, распознающий $K^{\uparrow C}$, что гарантирует максимально возможную реализацию ограничений на функционирование ДСС G, заданных спецификацией K. В случае единственного генератора и одной спецификации после завершения этапа построения супервизора может быть построена управляемая система, при этом справедливо равенство $L(J/G) = L(S||G)$, т.е. супервизор может быт присоединен посредством операции над автоматами. Поскольку множества событий S и G совпадают, вместо операции $S||G$ может быть использована операция $S \times G$. Для построения языков, описывающих функционирование системы под воздействием супервизора, используется ПОФ, где базы Bи B_{SupC} и содержат атомы описания переходов системы и супервизора, δ_1, δ_1^m отвечает переходам автомата системы, а δ_2, δ_2^m– переходам супервизора (Рис. 7).

$$\mathcal{F}_{SupC} = \exists B, B_{SupC} \begin{cases} \forall \sigma, q, \sigma', q'\ L(\sigma, q), \delta_1(q, \sigma', q'), \delta_2(q, \sigma', q') - \exists L(\sigma \cdot \sigma', q') \\ \forall \sigma, q, \sigma', q' L(\sigma, q), \delta_1^m(q, \sigma', q'), \delta_2^m(q, \sigma', q') - \exists L^m(\sigma \cdot \sigma', q') \end{cases}$$

Рис.7. Общая форма ПОФ-представления управляемой системы.

Разработан алгоритм (рис. 8) построения модульного супервизорного управления на основе ПОФ-представления ДСС для систем, состоящих из m подсистем и n спецификаций. При этом существенно используется тот факт, что для уменьшения вычислительной сложности решения задачи для построения локального супервизора $\mathcal{J}_i^{loc}$, реализующего спецификацию, могут быть задействованы только те подсистемы $\mathcal{G}_1^j$, $\mathcal{G}_2^j$, …, $\mathcal{G}_{n_j}^j$, которые имеют общие события со спецификацией $\mathcal{H}_j$. Алгоритм включает этап проверки свойства управляемости языка спецификации K и построения подъязыка $K^{\uparrow C}$.

Algorithm 1 Подход к решению основной задачи ТСУ на основе ПОФ

Вход: Множество генераторов $\mathcal{G}_i$ и множество автоматов-спецификаций $\mathcal{H}_j$, $L_m(\mathcal{H}_j) = K_j$, $j = 1, ..., n$.

1: **for** $j = 1$ **to** n **do**
2: **for all** K_j **do**
3: построить ПОФ $\mathcal{F}_{\mathcal{G}'_{j_1}||_\times \mathcal{G}'_{j_n}}$ чтобы получить $\mathcal{G}_j^{loc} = \mathcal{G}_1^j||\mathcal{G}_2^j|| \ldots ||\mathcal{G}_{n_j}^j$.
4: построить ПОФ $\mathcal{F}_{\mathcal{G}_j^{loc} \times \mathcal{H}_j}$ чтобы получить $\mathcal{G}_j^{loc} \times \mathcal{H}_j$.
5: построить ПОФ $\mathcal{F}_{Sub_j}$ чтобы проверить управляемость K_j с учётом $\mathcal{G}_j^{loc}$; в ходе проверки, строятся переходы $\mathcal{S}_j$, распознавателя $K_j^{\uparrow C}$.
6: извлечь переходы $\mathcal{S}_j$ из атомов базы B_{Sub_j} ПОФ $\mathcal{F}_{Sub_j}$.
7: **if** $K_j^{\uparrow C} \neq \varnothing$ **then**
8: использовать $\mathcal{S}_j^{loc}$ в качестве супервизора $\mathcal{J}_j^{loc}$ такого, что $L(\mathcal{J}_j^{loc}/\mathcal{G}_j^{loc}) = L(\mathcal{S}_j^{loc}||\mathcal{G}_j^{loc})$.
9: **else**
10: изменить $\mathcal{G}_1^j, \mathcal{G}_2^j, \ldots, \mathcal{G}_{n_j}^j$ или K_j.
11: **end if**
12: **end for**
13: **end for**
14: Построить супервизор как параллельную композицию

$$\mathcal{S}^{locmod} = \mathcal{S}_1^{loc}||\mathcal{S}_2^{loc}|| \ldots ||\mathcal{S}_n^{loc}$$

используя ПОФ $\mathcal{F}_{\mathcal{S}_1^{loc'}||_\times \mathcal{S}_n^{loc'}}$.
15: **return** $\mathcal{S}^{locmod}$.

Рис.8. Схема построения модульного супервизорного управления на основе ПОФ-представления ДСС.

3.3 Обработка событий

Преимуществом предлагаемого подхода является возможность дополнительной обработки и контроля событий на основе данных об окружающей среде в режиме реального времени в процессе логического вывода. Это может быть реализовано специальными логическими правилами, представленными в ПОФ в виде вопросов обработки событий. При ответе на них запускаются подвыводы, в которых события, поступающие из рабочей среды, служат параметрами, используемыми в расчетах или в других решениях. На Рис. 9 представлена ПОФ, генерирующая язык ДСС, состоящей из n генераторов, функционирование которых ограничивается совокупностью m спецификаций, при этом проводится обработка возникающего события. Следующие атомы составляют базу $\mathcal{F}_{Proc}$. $E(\sigma)$ — атом, определяющий все события ДСС. При реализации в каждый такт времени он заменяется реальным событием, произошедшим в системе. Атомы $I_i^G\left(q_0^{G_i}\right)$ и $I_i^H\left(q_0^{H_i}\right)$ определяют начальные состояния соответствующих автоматов. Атомы $\{\delta_i^G(\cdot,\cdot,\cdot)\}$, $\{\delta_j^H(\cdot,\cdot,\cdot)\}$ соответствуют переходам рассматриваемых автоматов. $L^{G_i}\left(\varepsilon, q_0^{G_i}\right)$, $L^{H_i}\left(\varepsilon, q_0^{H_i}\right)$ — атомы, которые будут содержать строки, сгенерированные соответствующими автоматами. При выводе строки соответствующих языков накапливаются в первых аргументах этих атомов. Вторые аргументы соответствуют состояниям, в которых произошли последние события строк первых аргументов.

$$\mathcal{F}_{Proc} = \exists B \begin{cases} \forall\sigma\ E(\sigma) & \exists\ T^{\#}(\sigma) \\ \forall\sigma, q_i, \sigma', {q'}_i\ L^{G_i}(\sigma, q_i), \delta^{G_i}(q_i, \sigma', {q'}_i) & \exists\ N_{G_\parallel}(\sigma, \sigma', {q'}_1 \cdot \ldots \cdot {q'}_n) \\ \forall\sigma, q, \sigma', q', {q'}_G\ L^{H_j}(\sigma, q), N_{G_\parallel}(\sigma, \sigma', {q'}_G), \delta^{H_j}(q, \sigma', q') & \exists\ N_{G_\parallel \times H_j}(\sigma, \sigma', {q'}_G \cdot q') \\ \forall\sigma, \sigma', q, {q'}_{H_1}, \ldots, {q'}_{H_m}\ L_S(\sigma, q), N^*_{G_\parallel \times H_1}\left(\sigma, \sigma', {q'}_{H_1}\right), \ldots, N^*_{G_\parallel \times H_m}\left(\sigma, \sigma', {q'}_{H_m}\right), E^*_T(\sigma) & \exists L^S\left(\sigma \cdot \sigma', {q'}_{H_1} \cdot {q'}_{H_2}\right) \end{cases}$$

$$\forall i = \overline{1,n}, \forall j = \overline{1,m}$$

Рис.9. Схема построения языка управляемой системы с предварительной обработкой событий.

Первый вопрос $\mathcal{F}_{Proc}$ служит для внешней обработки событий. Произошедшее в системе событие входит в вычислимую конструкцию $T^{\#}(\sigma)$ как аргумент. $T^{\#}$ может иметь любую разумную сложность, например это может быть некоторая логическая формула, например ПОФ (Рис. 10), или любой черный ящик, выполняющий вычисления или обработку знаний. Второй вопрос $\mathcal{F}_C$ порождает следующий символ слова языка параллельной композиции автоматов $\mathcal{G}_1$, ... , $\mathcal{G}_n$, и добавляет в базу вспомогательный атом $N_{\mathcal{G}_\parallel}$, необходимый для дальнейшего поиска независимых пересечений языков. Следующие m вопросов добавляют аналогичные атомы для автоматов $\mathcal{H}_{\mathcal{G}_\parallel \times \mathcal{H}_1}$, ..., $\mathcal{H}_{\mathcal{G}_\parallel \times \mathcal{H}_m}$. Последний вопрос ограничивает языковое событие, порожденное вторым вопросом, произведением автоматов $\mathcal{H}_1$, ..., $\mathcal{H}_m$. Здесь мы используем оператор удаления *, чтобы не использовать эти атомы в дальнейшем. Атом $E^*_T(\sigma)$ в этом вопросе является результатом обработки события σ конструкцией $T^{\#}$.

$$T(\sigma) = \exists\ Event(\sigma) \begin{cases} \forall\ Trigger & \exists\ Ret\left(Process^{\#}(\sigma)\right) \\ \forall x\ Event(x) & \exists\ E_T(x) \\ \forall x\ E_T(x) & \exists\ Ret(E_T(x)) \end{cases}$$

Рис.10. Вычислимая конструкция обработки событий в виде ПОФ.

Для АТР на основе ПОФ-исчисления был разработан прувер Bootfrost, который специализируется на выводе ПОФ без неограниченных переменных. Разработаны эффективные алгоритмы поиска логического вывода для этого класса формул. Исходный код проекта, реализованный на языке Rust, можно найти на странице GitHub (https://github.com/snigavik/bootfrost).

На Рис. 11 представлена схема реализации супервизорного управления в составе иерархической системы управления группой мобильных роботов. Реальная рабочая среда роботов представлена блоком Robots environment. События, происходящие в рабочей среде, распознаются, именуются предопределенными символами и передаются в подсистему обработки символьных данных. В этой подсистеме реализуется ДСС, представляющая собой формализацию решаемой роботами задачи, в виде конечных автоматов $\mathcal{G}_1$, ..., $\mathcal{G}_m$, описывающих систему, и

множество автоматов $\mathcal{H}_1$, ..., $\mathcal{H}_l$, описывающих ограничения на поведение системы. Затем ДСС представляется в виде ПОФ, вывод которой осуществляется прувером Bootfrost. Автоматы, лежащие в основе ДСС, представлены логическими атомами в базе фактов ПОФ. Логические правила, представленные в виде вопросов ПОФ, генерируют очередное состояние системы на основе событий с учетом наложенных ограничений. Таким образом, реализуются супервизоры $\mathcal{J}_1$, ..., $\mathcal{J}_n$, ответственные за достижение заданной цели.

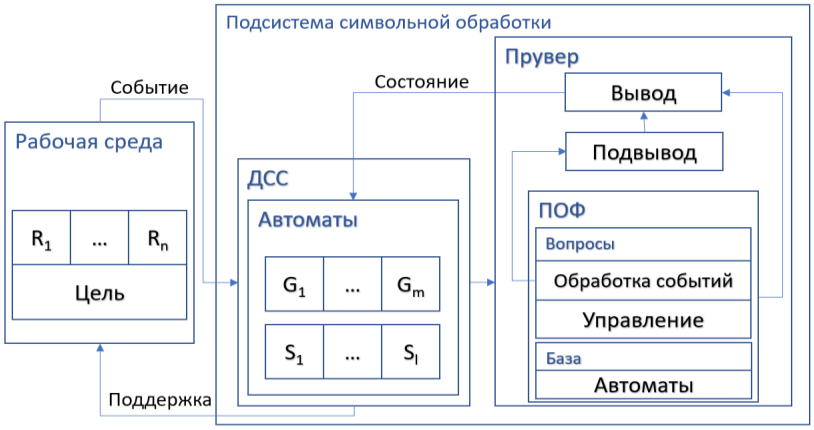


Рис. 11. Схема реализации символьной обработки данных на основе логического вывода в структуре контура управления.

4 Заключение

Предложенный подход к построению управления ДСС с помощью АДТ в исчислении ПОФ может быть использован для управления различными техническими системами и, среди прочего, в системах управления робототехническими комплексами. Логические средства исчисления ПОФ, возможность хранения и накопления знаний в базе фактов, возможность применения абдукции и модифицируемость семантики полезны в задачах с неопределенностями, возникающие в таких системах. Интеллектуализация алгоритмов теории супервизорного управления для ДСС в автоматном представлении позволит решать сложные практические задачи, поскольку полученные результаты позволят снизить вычислительную сложность алгоритмов построения супервизорного управления за счет использования информации, накапливаемой в процессе применяемого логического вывода. Подход может быть распространен на случай использования децентрализованных систем, а также на ДСС с частичным наблюдением событий.

Литература

[1] Karpas, E., Magazzeni, D. Automated Planning for Robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 2020, 3, 417-439, doi:10.1146/annurev-control-082619-100135.

[2] Zombori, Z.; Urban, J. & Brown, C. E. Peltier, N. & Sofronie-Stokkermans, V. (Eds.) Prolog Technology Reinforcement Learning Prover. *Automated Reasoning, Springer International Publishing*, 2020, 489-507, doi: 10.1007/978-3-030-51054-1_33

[3] Schader M., Luke S. Planner-Guided Robot Swarms. In: Demazeau Y., Holvoet T., Corchado J., Costantini S. (eds) Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness. The PAAMS Collection. PAAMS 2020. *Lecture Notes in Computer Science*, vol. 12092, 2020, Springer, Cham. https://doi.org/10.1007/978-3-030-49778-1_18

[4] Kovács L., Voronkov A. First-order theorem proving and vampire, in: N. Sharygina, H. Veith (Eds.), *Computer Aided Verification*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–35, 2013.

[5] Otten J. Nanocop: A non-clausal connection prover, in: *Proceedings of the 8th International Joint Conference on Automated Reasoning*, vol. 9706, Springer-Verlag, Berlin, Heidelberg, pp. 302–312, 2016, doi:10.1007/978-3-319-40229-1_21.

[6] Schulz S., Cruanes S., Vukmirović P. Faster, higher, stronger: E 2.3, in: P. Fontaine (Ed.), *Proc. of the 27th CADE*, Natal, Brasil, number 11716 in LNAI, Springer, pp. 495–507, 2019.

[7] Vassilyev S.N. Machine synthesis of mathematical theorems. *The Journal of Logic Programming.* 1990. Vol. 9. P. 235–266.
[8] Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунов Б.Е. Интеллектное управление динамическими системами, М.: Физмат-лит, 2000.
[9] Lafortune S. Discrete Event Systems: Modeling, Observation, and Control. *Annual Review of Control, Robotics, and Autonomous Systems.* 2019. Vol. 2. N. 1, P. 141-159.
[10] Cassandras C.G., Lafortune S. Introduction to Discrete Event Systems, 3rd ed., Cham, Switzerland: Springer, 2021, XXVI.-804 pp.
[11] W.M. Wonham, K. Cai, K. Rudie. Supervisory control of discrete-event systems: A brief history. *Annual Reviews in Control.* 2018. Vol. 45. P. 250-256.
[12] W.M. Wonham, K. Cai. Supervisory Control of Discrete-Event Systems. 2019, Springer International Publishing.
[13] Davydov A.V., Larionov A.A., Nagul N.V. On Checking Controllability of Specification Languages for DES. In: *Proc. of the 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, 2020. P. 1151-1156.
[14] Davydov A., Larionov A., Nagul N. The construction of controllable sublanguage of specification for DES via PCFs based inference. In: *Proc. of the 2nd International Workshop on Information, Computation, and Control Systems for Distributed Environments, ICCS-DE 2020*, Irkutsk, Russia, July 6-7, 2020: CEUR-WS Proceedings, 2020, Vol. 2638. P. 68-78.
[15] Davydov A.V., Larionov A.A., Nagul N.V. Application of the PCF calculus for solving the problem of nonblocking supervisory control of discrete event systems. *Journal of Physics: Conference Series.* 2021. Vol. 1864. P. 012048. DOI: 10.1088/17426596/1864/1/01204.
[16] Bychkov I., Ul'yanov S., Nagul N., Davydov A. Kenzin M., Maksimkin N. Hierarchical event-based control of multi-robot systems in unstructured environments. *Journal of Physics: Conference Series.* 2021. Vol. 1864. P. 012001. DOI: 10.1088/17426596/1864/1/012001.

Метод автоматического построения тезауруса технического документа

Демидов Д.В.[1], Мардер Л.М.[1]

[1]*Национальный Исследовательский Ядерный Университет «МИФИ», Каширское ш., д. 31, г. Москва, 115409, Россия.*

dvdemidov@mephi.ru, asertolpas@gmail.com

Аннотация. *Рассматривается задача автоматического построения тезауруса технического документа на примере требований по безопасности информации ФСТЭК России. Предлагается метод извлечения понятий на основе предварительного отбора терминов-кандидатов с помощью поверхностно-синтаксических шаблонов и последующей фильтрации терминов по статистической метрике. Предлагается метод извлечения отношений между понятиями на основе предварительного установления ассоциативных отношений на основе статистического подхода – матрицы встречаемости – с последующей классификацией отношений ассоциации по ряду онтологических отношений.*

Ключевые слова: извлечение терминов, извлечение отношений, автоматическое построение тезауруса, анализ технической документации

1 Введение

Программное обеспечение (ПО) на отечественном рынке, как правило, подлежит проверке на соответствие требованиям. Традиционно в фокусе внимания находятся функциональные требования, представляющие интерес для предприятий-потребителей, и требования к безопасности информации, являющиеся обязательными для выполнения при построении таких информационных систем как государственные информационные системы (ГИС), информационные системы персональных данных (ИСПДн), информационные системы значимых объектов критической информационной инфраструктуры (ИС ЗОКИИ). Полномочия по сертификации тех или иных классов ПО возложены на такие органы исполнительной власти как Федеральная служба по техническому и экспортному контролю (ФСТЭК) и федеральная служба безопасности (ФСБ), Служба защиты государственной тайны вооружённых сил РФ и др. Во многом процедура установления соответствия требованиям заключается в сопоставлении требований регулятора с технической и эксплуатационной документацией. При этом объём регламентирующей документации может исчисляться сотнями страниц, а объём документации на ПО в разы больше. Кроме того, с одной стороны, нормативная база постоянно развивается, а с другой стороны, выпускаются новые версии программных продуктов. В связи с этим, необходимость в автоматизации процессов анализа и сопоставления документации только обостряется. Попытки решения этих задач чрезвычайно важны, так как могут облегчить работу и производителям ПО, и органам по сертификации, и организациям-заказчикам.

Рассмотрим следующую задачу. Пусть есть некоторый свод требований по безопасности информации. Одно из требований содержит текст «Должно обеспечиваться безопасное сетевое взаимодействие (возможность фильтрации трафика) с внешними информационными системами». Пусть имеется руководство администратора (эксплуатационная документация), где в одном из абзацев описывается организация безопасного сетевого взаимодействия с внешними информационными системами. Выделяя из свода требований термины «безопасное сетевое взаимодействие» и «внешние информационные системы», можно провести их поиск в эксплуатационной документации. Однако, прямое сопоставление здесь слабо эффективно из-за больших отличий в языках описания требований и документации. Основной гипотезой данной

работы является то, что сопоставление фрагментов тезаурусов требований и документации может быть более эффективным. Для этого необходимо разработать методы и программные средства автоматического построения тезауруса по техническому документу, реализовать методы сопоставления терминов с использованием отношений между ними. В дальнейшем можно наращивать аппарат более сложными методами обработки естественного языка, онтологического инжиниринга, инженерии знаний.

2 Постановка задачи построения тезауруса технического документа

В качестве целевого объекта для построения был выбран информационно-поисковой тезаурус – разновидность онтологии, представляющая собой словарь терминов, в котором все элементы связаны между собой отношениями нескольких типов [1]. Такой тезаурус предлагается строить и для эксплуатационной документации, и для нормативной базы, с которой необходимо устанавливать соответствие.

Представим набор технических документов в виде следующей теоретико-множественной модели.

$TechDocs = \{Doc_1, \dots, Doc_n\}$ – набор технических документов

$Doc_i = < Title, Intro, TableOfContents, Chapters, Appendices, Literature >$ – отдельный документ

- $Title$ – название документа
- $Intro$ – введение
- $TableOfContents$ – содержание
- $Chapters = < Paragraphs_1, Tables_1 >$ – главы документа, основное наполнение
- $Appendices = < Paragraphs_2, Tables_2 >$ – приложения
- $Literature$ – список литературы
- $Paragraphs_i = \{ Paragraph_{i1}, \dots, Paragraph_{in} \}$ – абзацы в одном разделе
  - $Paragraph_{ij} = < PContent, PNumber >$ – отдельный абзац
    - $PContent$ – содержимое одного абзаца
    - $PNumber$ – номер абзаца
- $Tables_i = \{Table_{i1}, \dots, Table_{in}\}$ – таблицы в одном разделе
  - $Table_{ij} = < Headers, Cells, TNumber >$ – отдельная таблица
    - $Headers$ – заголовки таблицы
    - $Cells$ – ячейки таблицы
    - $TNumber$ – номер таблицы

Каждый документ состоит из нескольких частей, в данном случае выделяются введение, основное наполнение, приложение и т.д. Каждая из частей вносит своё влияние, может быть как обязательной, так и необязательной, иметь разный состав. Однако, для каждой части отмечается наличие абзацев, а также возможность наличия таблиц. Именно внутри абзацев и содержатся термины, которые требуется извлечь и отобрать. Также наличие абзацев позволяет устанавливать ассоциации между терминами, об этом – далее. Таблицы являются сложной структурой, отличной от абзацев, поэтому выделяются отдельно и требуют отдельной обработки.

В связи с этим, выделяются следующие этапы при автоматическом построении тезауруса технического документа.

1. Отбор терминов-кандидатов.
2. Фильтрация терминов.
3. Установление отношений ассоциации между терминами.
4. Классификация отношений между терминами.

3 Метод автоматического построения тезауруса технического документа

3.1 Отбор терминов-кандидатов

За основу представленного в данной работе метода выделения терминов-кандидатов взят механизм разделения текста (chunking) [2] с помощью лингвистических шаблонов. Выражения на языке шаблонов похожи на регулярные выражения, используемые для обработки текста. Каждое регулярное выражение может быть преобразовано в конечный автомат, согласно которому происходит сопоставление строки с описанным правилом.

Представим язык для описания лингвистических шаблонов терминов (см. Табл. 1).

Таблица 1. Конструкции языка описания шаблонов терминов.

Обозначение	***Расшифровка***
СА	Существительное
Прил	Прилагательное
Прич	Причастие
КрПрил	Краткое прилагательное
КрПрич	Краткое причастие
Предлог	Предлог
_ n	Требование наличия именительного падежа
(_ _ …)	Группировка
_ | _ | …	Выбор одного из вариантов (2 и более)
_ +	Вхождение одного или нескольких элементов
_ *	Вхождение нуля или нескольких элементов
_ ?	Вхождение одного или нуля элементов

Примерами выражений на таком языке могут быть:

1. СА Предлог? (Прил | Прич)
2. (СА)n+ (КрПрил | КрПрич)* Предлог? СА*

Для разбора выражений на предложенном языке будет использоваться анализатор – конечный автомат. Каждое выражение может быть преобразовано в граф, имеющий одну начальную и множество конечных вершин (начальная вершина может входить в множество конечных). Могут включаться петли и циклы, соответствующие повторениям одной и той же последовательности частей речи. Вершина графа – состояние конечного автомата, переходы между вершинами – часть речи. Так, последовательно обходя вершины, можно проверить фрагмент текста на соответствие описанному шаблону. На рисунке 1 представлен пример преобразования выражения в конечный автомат. Чёрным цветом помечены конечные вершины.

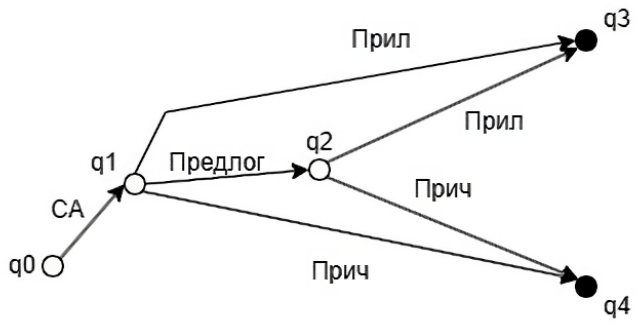


Рис.1. Пример преобразования выражения «СА Предлог? (Прил | Прич)» на языке описания шаблонов терминов в конечный автомат.

В общем случае подобный автомат не является детерминированным, возникают сложности с обработкой «петлей» конечного автомата, поэтому описания шаблонов можно автоматически трансформировать перед интерпретацией следующим образом:

- Если в конце описания шаблона стоит конечная вершина с образцом «_ *», то создаётся промежуточный необязательный элемент, который может как входить, так и не входить в термин. На языке это будет описываться как «_ ? _ +».
- Если в конце описания шаблона стоит конечная вершина с образцом «_ +» , то создаётся промежуточный элемент, который будет входить в термин, но обрабатывается отдельно. На языке это будет описываться как «_ _ +».

Рассмотрим выделение терминов-кандидатов с помощью таких шаблонов на примере предложения «Настоящий стандарт устанавливает основные понятия и принципы оценки безопасности ИТ». Оно преобразуется в цепочку частей речи «Прил СА глагол прил СА союз СА СА СА СА». Примеры шаблонов терминов:

- Правило «Прил САn». Выделяются термины: настоящий стандарт, основные понятия.
- Правило «САn СА». Выделяются термины: принципы оценки.
- Правило «(Прил | СА)n СА+». Выделяются термины: настоящий стандарт, основные понятия, принципы оценки безопасности ИТ.

Предложенный метод выделения терминов-кандидатов позволяет легко масштабировать процесс выделения терминов путём создания новых лингвистических шаблонов для сопоставления с текстом. Используя набор шаблонов можно добиться отбора терминов множества видов.

Итак, выделение терминов-кандидатов происходит с использованием множества шаблонов. Они позволяют выделять как отдельные слова, так и словосочетания. Каждый анализатор (конечный автомат, построенный по выражению) независимо от остальных производит обработку текста и выделяет собственные термины-кандидаты. Полученные множества объединяются.

Для выделения терминов из технического текста в рамках текущей работы предлагаются следующие лингвистические шаблоны.

- САn – существительное в именительном падеже.
- (Прил | Прич)+ САn – существительное в именительном падеже с прилагательными в начале.
- ((Прил | Прич)n | (САn Предлог?)) СА – словосочетание с двумя словами и опциональным предлогом.
- (Прил | СА)n (Прил | Прич | СА)+ СА – словосочетание с более тремя и более словами.
- (Прил | СА)n ((Прил | Прич | СА)* (СА Предлог?))+ (Прил | Прич | СА)* СА – сложное словосочетание.

3.2 Фильтрация терминов

Среди терминов-кандидатов могут быть слососочетания, не являющиеся терминами, редко встречаемые слососочетания и др. «мусорные» слососочетания. Поэтому предлагается проводить последующий отбор на основе статистической значимости каждого термина-кандидата в тексте. Значимость подсчитывается с помощью модифицированной метрики C-Value [3].

$$CValue(a) = \begin{cases} \log_2(L_a + 1) * N_a, \text{если } a - \text{не вложенный} \\ \log_2(L_a + 1) * \left(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b)\right), \text{иначе} \end{cases} \quad (1)$$

где L_a – количество слов в термине a, $f(.)$ – количество вхождений термина в набор текстов, T_a – множество терминов-кандидатов, которые содержат a как вложенный, $P(T_a)$ – число таких терминов-кандидатов. Модификация формулы из [3] заключается в добавлении $+1$ под логарифм

для того, чтобы не отсеивались термины, состоящие из одного слова – для них значение логарифма было бы равно 0.

Для отсеивания терминов по значению C-Value применяется некоторое пороговое значение. При исследовании получаемых значения C-Value эмпирически было найдено, что их распределение примерно соответствует нормальному распределению с большим значением μ. В связи с этим в данной работе предлагается в качестве порогового значения использовать 25-ый процентиль получаемых значений, который является аналогом 0.25 квантиля нормального распределения и используется для отсечения малозначимых значений в наборе терминов-кандидатов.

С учётом этого, предлагается следующая процедура отбора терминов:

1. Вычисление C-Value по формуле (1) для каждого термина-кандидата.
2. Сортировка терминов-кандидатов по значению C-Value.
3. Вычисление 25-го процентиля по C-Value в качестве порогового значения.
4. Отбор тех терминов, для которых полученное значение метрики строго больше порога.

3.3 Установление отношений между терминами

Для выявления онтологических связей необходимо сначала установить предварительные ассоциативные связи между терминами (по аналогии с извлечением терминов-кандидатов). В рамках данной работы введены следующие ограничения:

- Рассматриваются только бинарные отношения между терминами a и b.
- Ассоциативные связи симметричны.
- Отношения устанавливаются только для таких пар терминов, которые встречаются совместно в абзаце исходного текста.

Для установления ассоциативных связей между терминами используется матрица встречаемости размера $N \times N$, где N – это количество отобранных терминов. Заголовками столбцов и строк матрицы является перечень извлечённых терминов. В элементах матрицы – количество совместного присутствия пары терминов в одном или нескольких абзацах. Матрица нормализуется – каждый элемент делится на общее число терминов.

Математически матрица может быть записана следующим образом.

$$SM = \left\| a_{ij} \right\|_{N \times N} \tag{2}$$

$$a_{ij} = \frac{G(a,b)}{N} \tag{3}$$

где a_{ij} – элемент матрицы, $G(a,b)$ – число совместной встречаемости для пары терминов a и b.

Ясно, что матрица встречаемости симметрична (см. таблицу 2).

Таблица 2. Пример матрицы встречаемости для трёх терминов.

	Платформа виртуализации	VDI	Виртуальная машина
Платформа виртуализации	0	2	1
VDI	2	0	3
Виртуальная машина	1	3	0

После вычисления матрицы встречаемости требуется провести обработку пар терминов, подсчитать некоторую статистическую величину, по которой будет определяться сила ассоциативной связи. В рамках данной работы предлагается использовать парную информацию (Pointwise Mutual Information, PMI) [4]. Данная метрика показывает, насколько сильно влияние двух терминов друг на друга.

$$PMI(a,b) = \log_2\left(\frac{P(a,b)}{P(a) * P(b)}\right) \tag{4}$$

где $P(a)$ – вероятность вхождения термина a в текст, $P(a,b)$ – вероятность совместной встречаемости терминов a и b.

Как и в случае с C-Value, необходимо отфильтровать незначимые ассоциативные связи по некоторому пороговому значению. Здесь также предлагается использовать 25-ый процентиль.

Учитывая всё перечисленное, предлагается следующая процедура выделения ассоциативных отношений между терминами:

1. Построение матрицы встречаемости для отобранных терминов.
2. Заполнение матрицы встречаемости по формулам (2) и (3).
3. Подсчёт значений парной информации по формуле (4) для каждой пары терминов.
4. Вычисление 25-го процентиля по PMI в качестве порогового значения.
5. Отбор тех пар терминов, для которых полученное значение метрики строго больше порога.

3.4 Классификация отношений между терминами

В работе рассматриваются статистические и текстовые методы выделения отношений между терминами. Методы, требующие семантического анализа или специальных словарей, пока остаются за рамками. Так, например, не будут выделяться связи между понятиями «Операционная система» и «Linux» (отношение Выше-Ниже).

Рассмотрим подходы выявления отношений разных типов.

- Выявление отношений типа «Выше-Ниже». Проверяется лексическая вложенность одного термина в другой. Например, термин «операционная система в сочетании с рабочей станции» будет гипонимом термина «операционная система», т.к. первый термин уточняет второй. Видами отношения «Выше-Ниже» являются родовидовое отношение «Класс-подкласс», отношение «Класс-экземпляр» и, с некоторой натяжкой, отношение «Часть-Целое». Последнее не совсем корректно, но может быть в последствии отфильтровано, например, с помощью специализированных словарей.
- Выявление отношений синонимии для связывания терминов с их аббревиатурами. Для этого проверяется соответствие начальных букв слов, составляющих термин, (исключая союзы) с буквами, составляющими аббревиатуру. Например, будут связаны термины «ИТ» и «информационные технологии».
- Выявление отношений синонимии для различных форм словосочетаний. Для этого вычисляется мера близости двух терминов, например, расстояние Левенштейна [5], которое показывает, насколько два словосочетания лексикографически близки друг к другу. Например, так устанавливается отношение синонимии между «информационные технологии» и «информационная технология». Данная разновидность синонимии может быть в последствии уточнена.

3.5 Общий алгоритм автоматического построения тезауруса

С учётом предыдущих разделов, для первичного автоматического построения тезауруса предлагается последовательность следующих этапов:

1. Предварительная обработка текста из множества документов.
2. Отбор терминов-кандидатов.
3. Фильтрация терминов по статистической метрике C-Value.
4. Выделение ассоциативных связей между терминами.
5. Установление ассоциативных связей по метрике взаимной информации PMI.
6. Классификация выделенных связей.

Выходными данными алгоритма является тезаурус, созданный на основе входного набора документов. Полезно также строить указатель встречаемости терминов в документах, разделах, страницах и др. Такой тезаурус позволит выполнять информационно-поисковые запросы.

В случае пополнения набора документов новым документом уже существующий тезаурус необходимо перестроить. Для этого предлагается следующий алгоритм:

1. Из нового документа отбираются термины. Производится их фильтрация. Отобранные термины добавляются в тезаурус.
2. Строится новая матрица встречаемости терминов по всем абзацам с учётом нового документа.
3. Вычисляется множество ассоциативных связей, которое сравнивается с ранее построенным множеством. Далее рассматриваются только новые связи, которые не были выявлены ранее.
4. Новые связи классифицируются и добавляются в тезаурус.
5. Перестраиваются индекс встречаемости терминов с учетом нового документа.

Таким образом, тезаурус будет достроен с учётом добавленного документа. Отметим, что в результате работы алгоритма на новых статистических данных часть прежних связей может быть отсеяна по порогу. Потенциально их можно было бы удалять из тезауруса, но на практике имеет смысл сохранять их.

4 Апробация метода автоматического построения тезауруса

Для апробации метода был разработан прототип программных средств на языке Python. В качестве источника терминов был выбран документ «ГОСТ Р ИСО/МЭК 15408-1-2012 ИТ. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель». Документ содержит требования в области информационной безопасности, в нём определяются методология и критерии оценки, которые позволяют проводить независимую оценку ПО на соответствие функциональным требованиям безопасности и требованиям доверия. Документ содержит 56 страниц, из них 18 страниц приложений. Для анализа были взяты страницы с 3 по 52 – именно в них содержится основная информация, важная для выделения терминов.

В следующей таблице представлено подмножество выявленных терминов-кандидатов с их оценками C-Value, а также указанием того, попадает ли термин в результирующее множество терминов. Для отбора терминов использовалось пороговое значение 1.

Таблица 3. Примеры отобранных терминов-кандидатов.

Термин-кандидат	***Оценка C-Value***	***Принадлежность к результирующему множеству терминов***
Критерии для оценки	2.00	+
Независимая оценка математических свойств криптографии	2.58	+
Независимая оценка	1.00	–
Терминология ИСО	1.58	+
Способ действия	1.00	–
Определённый способ действия	2.00	+
Предпочтительная возможность	1.58	+
Национальный стандарт Российской Федерации	2.32	+
Национальный стандарт	1.00	–
Информационная технология	3.00	+
Технология	0.00	–
Средства обеспечения	4.00	+
Средства обеспечения безопасности	8.00	+
Критерии оценки	7.00	+
Критерии оценки безопасности информационных технологий	7.75	+
Information	0.00	–
Information technology	1.58	+
IT security	1.00	–
Evaluation criteria for IT security	2.58	+
Принципы	0.00	–
Принципы оценки	2.00	+
Принципы оценки безопасности ИТ	4.64	+

Гост р	76.08	+
Использование надлежащего порядка разработки программного обеспечения	2.81	+
Вид деятельности	2.00	+
Вид деятельности на стадии жизненного цикла некоторого продукта	3.17	+
Приведение ОО	1.00	–
Приведение ОО в состояние готовности	2.58	+
Проверка документации	1.00	–
Успешная проверка документации	2.00	+

О выделенных терминах можно сделать следующие выводы.

- В результирующее множество отобралось примерно 60% выделенных терминов-кандидатов.
- Термины, вложенные в другие термины, имеют оценку ниже, чем невложенные. Это связано с тем, что метрика C-Value учитывает случаи, в которых термин является частью другого. Тем не менее, некоторые вложенные термины всё же выделяются, например, «Принципы оценки». Это происходит из-за того, что они встречаются довольно часто сами по себе, а их оценка по формуле (1) не сильно уменьшается от вложенности.
- Разработанный метод учитывает термины как на русском, так и на английском языке. Примеры: «Information technology», «Evaluation criteria for IT security».
- Пропускаются некоторые нужные термины. Например, пропускается термин «ГОСТ Р ИСО/МЭК 15408-1-2012 ИТ». Это может быть связано как с предварительным отбором термином, так и с их фильтрацией по C-Value.
- Выделяются некоторые лишние термины. Например, «ГОСТ Р». Этот термин вложен в наименование документа «ГОСТ Р ИСО/МЭК 15408-1-2012 ИТ» и не встречается отдельно, однако имеет очень высокую оценку C-Value. Это происходит из-за того, что термин «ГОСТ Р ИСО/МЭК 15408-1-2012 ИТ» не был выделен.

В таблице 4 представлены примеры выделенных отношений между терминами с оценками взаимной информации PMI, а также указанием класса отношения. Для отбора отношений было вычислено пороговое значение 5.39.

Таблица 4. Примеры выявленных отношений.

Термин 1	***Термин 2***	***Оценка PMI***	***Класс отношения***
Перемещение	Перемещение незавершённого ОО	10.65	Гипероним
Перемещение	Примеры	6.48	Ассоциация
Инструментальные средства разработки	Инструментальные средства разработки включая программные средства тестирования	11.65	Гипероним
Инструментальные средства разработки включая программные средства тестирования	Программные средства	10.65	Гипоним
Интегральная схема смарт-карты	Программные средства	10.65	Ассоциация
Локальная вычислительная сеть	Программные средства	8.65	Ассоциация
Life-cycle	Определение модели жизненного цикла	10.07	Ассоциация
Уязвимость	Потенциальная уязвимость	9.65	Гипероним
Виртуальная частная сеть	Инфраструктура открытых ключей	11.65	Ассоциация
Интегральная схема	Интегральная схема смарт-карты	9.65	Гипероним

О выявленных отношениях можно сделать следующие выводы.

- Отношение ассоциации не несёт семантической нагрузки само по себе, а лишь показывает, как пара терминов соотносится друг с другом в тексте. Однако это отношение может быть классифицировано более точно в дальнейшем.
- Не учитываются переводы слов. Так, например, между терминами «Life-cycle» и «Определение модели жизненного цикла» стоит простая ассоциация. Отношение может быть классифицировано с применением межъязыковых словарей.

5 Заключение

В работе представлен метод автоматического построения тезауруса технического документа. Метод обобщён для набора документов и для случая пополнения набора документов новым документом. Проведена апробация метода, представлены примеры выявленных терминов и отношений построенного тезауруса.

Разработанный метод не зависит от предметной области, однако уже позволяет решать задачу информационного поиска релевантных абзацев технической документации, относящихся к некоторому требованию нормативной документации. Метод целесообразно дополнить более сложными, возможно, предметно-ориентированными методами для уточнения выявляемых отношений. Другие модификации метода будут направлены на повышение качества отбора терминов, например, более тщательный подбор лингвистических шаблонов, по которым производится сопоставление с текстом.

Литература

[1] Н.В. Лукашевич. Тезаурусы в задачах информационного поиска. — Издательство Московского университета, 2011. — С. 396.

[2] Bird S. Klein E. Loper E. Natural Language Processing with Python. — O'Reilly Media, Inc., 2009. — C. 512.

[3] Mima H. Frantzi K.T. Ananiadou S. Automatic Recognition of Multiword Terms: The Cvalue/ NCvalue Method // Conference Paper in International Journal on Digital Libraries. — 2000. — C. 115–130

[4] Church K. Hanks P. Word association norms, mutual information, and lexicography // Proceedings of ACL89. — 1989. — C. 76–83

[5] Noh S. Kim S. Jung C. A Lightweight Program Similarity Detection Model using XML and Levenshtein Distance // FECS. — 2006. — C. 3–9.

[6] Росстандарт. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель. — Стандартинформ, 2014. — С. 51

РАЗРАБОТКА ВИРТУАЛЬНОЙ МАШИНЫ ДЛЯ ИСПОЛНЕНИЯ L*-ПРОГРАММ Р-ПОЛНОГО ЛОГИЧЕСКОГО ЯЗЫКА В ДЕЦЕНТРАЛИЗОВАННЫХ СРЕДАХ

Долгов Н.Ю.

Новосибирский государственный университет, ул. Пирогова, д. 1, г. Новосибирск, 630090, Россия.

nikitabrn1999@gmail.com

Аннотация. *В децентрализованных средах, при выполнении смарт-контрактов, зачастую возникает проблема, связанная с возможностью невыполнения контракта, если его вычислительная сложность превышает определенный порог. В качестве решения этой проблемы мы предлагаем описывать смарт-контракты на р-полном объектно-ориентированном языке L* [1], тогда их сложность можно точно рассчитать, поскольку такие смарт-контракты будут гарантированно вычислимы за полиномиальное время. Язык L* можно использовать во многих областях, в частности, для описания алгоритмов искусственного интеллекта или смарт-контрактов, при исполнении которых зацикливание и зависание исключены, в отличие от других языков. В качестве основы для децентрализованной среды исполнения L*-программ было решено выбрать блокчейн. В работе приводится описание виртуальной машины, разработанной для исполнения программ L* в децентрализованных средах.*

Ключевые слова: блокчейн, смарт-контракты, объяснительный искусственный интеллект

1 Введение

До недавнего времени не было разработано ни одного р-полного языка программирования. Более того, для существующих Тьюринг полных языков программирования не существует средств, при помощи которых можно определить, будет ли программа выполнена за полиномиальное время или нет. При этом свойство полиномиальной сложности программ – это одно из ключевых свойств, т.к. позволяет исполнять программы быстро, причём их зацикливание и зависание исключены, в отличие от Тьюринг полных языков.

В научных работах [2,3] Гончаровым С.С., Свириденко Д.И. и Нечесовым А.В. был представлен логический язык L. Было доказано, что с помощью данного языка можно описывать любые алгоритмы полиномиальной вычислительной сложности. Разработанный позднее язык L* [1], являющийся объектно-ориентированным расширением языка L, сохраняет важнейшее это свойство. Помимо самого языка L*, в работе [1] было дано описание виртуальной машины для исполнения программ этого языка. С помощью языка программирования Java, опираясь на работы [1-3], был реализован прототип виртуальной машины для исполнения программ логического программирования L*.

Язык L* может использоваться для построения объяснительного искусственного интеллекта, поскольку используемые в обучении алгоритмы являются полиномиально вычислимыми. На данный момент большинство алгоритмов искусственного интеллекта и машинного обучения являются по сути черным ящиком, что затрудняет их использование во многих областях. С помощью языка L*, напротив, представляется возможным построить

искусственный интеллект следующего поколения, где результаты будут объяснимы понятным человеку языком, а в сочетании с большим набором данных и применением теории индуктивного логического программирования можно порождать новые гипотезы и быстро проверять их истинность.

В данной работе было решено сосредоточиться на аспектах исполнения L*-программ в децентрализованных системах на базе блокчейна, а также рассмотреть применение языка для описания смарт-контрактов, поскольку данное направление является достаточно перспективным. Полиномиальная вычислимость смарт-контрактов является существенным свойством, поскольку, во-первых, зачастую с пользователя, делающего запрос к смарт-контракту, взимается плата, зависящая от времени исполнения его транзакции, а во-вторых, транзакция может быть не выполнена, если превышен порог выполнения. Еще одним немаловажным аспектом является безопасность. Хранение исходного кода и результатов транзакций в децентрализованной среде позволит защитить их от взлома и злонамеренного изменения.

2 Компиляция и выполнение L*-программ

2.1 Язык L*

Приведем краткое описание языка L*.

В качестве базовой модели для обоих языков выступает наследственно-конечная списочная модель $HW(M)$ конечной сигнатуры σ [4], расширенной константой nil, списочными функциями языка LISP (head, cons и т.д.), двуместными предикатами $\in$ и $\subseteq$ – определяющих элемент списка и начальный сегмент списка соответственно, а также одноместный предикат U – который выделяет основное множество M.

В основе языка L лежит язык, состоящий из Δ_0-формул [5] сигнатуры σ модели $HW(M)$ и стандартных термов логики предикатов первого порядка. Помимо этого, язык L содержит две уникальные конструкции – это условный терм и p-итерационный терм [1]. Первая конструкция по смыслу схожа с if-then-else в языках программирования, а вторая – с циклом for. Было доказано [3], что этого достаточно для описания любых алгоритмов полиномиальной вычислительной сложности.

Язык L* расширяет язык L, при этом позволяя пользователю, как в других высокоуровневых языках программирования, описывать и использовать свои функции и классы. Функции описываются при помощи ключевого слова function:

$$[mod]\ function\ f(x_1,\ldots,x_n)\ \{p\}$$

где p – L*-подпрограмма, являющаяся телом функции, mod – модификатор доступа (public, protected, private). Классы описываются при помощи ключевого слова class, включают в себя описание атрибутов и методов, есть возможность наследования от одного родительского класса:

$$class\ ClassName\ [extends\ parentClass]\{p\}.$$

Данных языковых конструкций достаточно, чтобы писать полноценные программы.

2.2 Компиляция L* программ

Перед тем как исполнять L* программу, ее предварительно необходимо скомпилировать. Программа считывается из файла и для нее строится абстрактное синтаксическое дерево средствами библиотеки JavaCC [6] на основе описанной грамматики языка L*. На этом же этапе осуществляется проверка грамматических ошибок в программе. В случае их нахождения, пользователь будет уведомлен о найденных ошибках и процесс компиляции завершится.

В случае, если грамматических ошибок выявлено не было, дальнейший этап компиляции L* программы предполагает удаление из кода частей отвечающих за описание классов и функций и их преобразование в другой формат. Остальной код остается в неизменном виде.

В работе [1] процесс компиляции описывается как удаление частей кода отвечающих за описание классов и функций и последующее их добавление в списки со следующей структурой:

- $< funcName, [mod], < x1, \ldots, xk >, p* >$
- $< className, < x1, \ldots, xk >, < f1, [mod], < x1, \ldots, xk1 >, p1* >, \ldots, < fm, \ldots >>$

где $x_1, \ldots, x_k$– имена свойств класса, $f_1 \ldots f_m$ – названия методов класса. Отметим, что в полученной скомпилированной программе больше не содержится никаких конструкций, описывающих классы или функции.

```
function factorial(x) {
   result := 1;

   iteration<x>(i := 1, i <= x ==> do {
      result := result * i;
      return i + 1;
   });
   return result;
}

function get_even(x) {
   return cond((x / 2) * 2 = x ==> x, x + 1);
}

n := 2 ^ 3;
print(factorial(n));
print(get_even(3));
print(get_even(6));
```

Рис. 1. Пример L* программы.

При реализации компилятора было решено в качестве представления скомпилированных функций и классов использовать объекты описанные на языке программирования Java – это позволит более удобно обрабатывать их на этапе исполнения программы.

```
public interface ICompiledClass {
    String getName();
    ICompiledClass getBase();

    IMember accessMember(final String name, final EnumSet<AccessType> access);
    ICompiledMethod accessMethod(final String name, final EnumSet<AccessType> access);
    ICompiledMethod getConstructor(final EnumSet<AccessType> access);
}
```

Рис. 2. Фрагмент Java кода, описывающий интерфейс для хранения информации о скомпилированных L*-классах.

2.3 Виртуальная машина и выполнение L* программ

Виртуальная машина выполняет скомпилированные L* программы. Лента состоит и последовательных ячеек. Каждая ячейка содержит список означенных переменных и список инструкций, который необходимо исполнить, а также информацию о том, в какой функции/методе находятся данные инструкции. В первую ячейку кладется вся скомпилированная программа, т.е. программа, из которой удалены все определения классов и функций. На каждом такте виртуальная машина исполняет текущую инструкцию в последнем блоке ленты.

В случае, если встречается “нестандартная” конструкция [1], например вызов L* определимой функции (подпрограммы), то исполнение прерывается на этом моменте. Такой конструкцией может являться вызов функции/метода/конструктора. В этом случае эта конструкция заменяется на временную переменную, куда будет записан результат после исполнения вызываемой подпрограммы. Тело вызываемой подпрограммы ищется в списках скомпилированных функций/классов и кладется на ленту. При этом аргументы функции означиваются передаваемыми параметрами. Отмечу, что стандартные функции (например, print, cons, len и т.д.) описаны на самом языке Java внутри виртуальной машины и исполняются за один такт, не требуя расширение ленты. Список таких функций может быть расширен в дальнейшем. В случае, если встречается “нестандартная” конструкция return, или закончились инструкции в блоке, то текущая ячейка удаляется с ленты, и последняя временная переменная, если она была, инициализируется значением описанным в конструкции return или значением nil соответственно. После удалении ячейки с ленты, предыдущая ячейка снова становится текущей и исполнение продолжается с той же инструкции, где оно прервалось. На рисунке 3 показан пример выполнения простой программы языка L* на виртуальной машине.

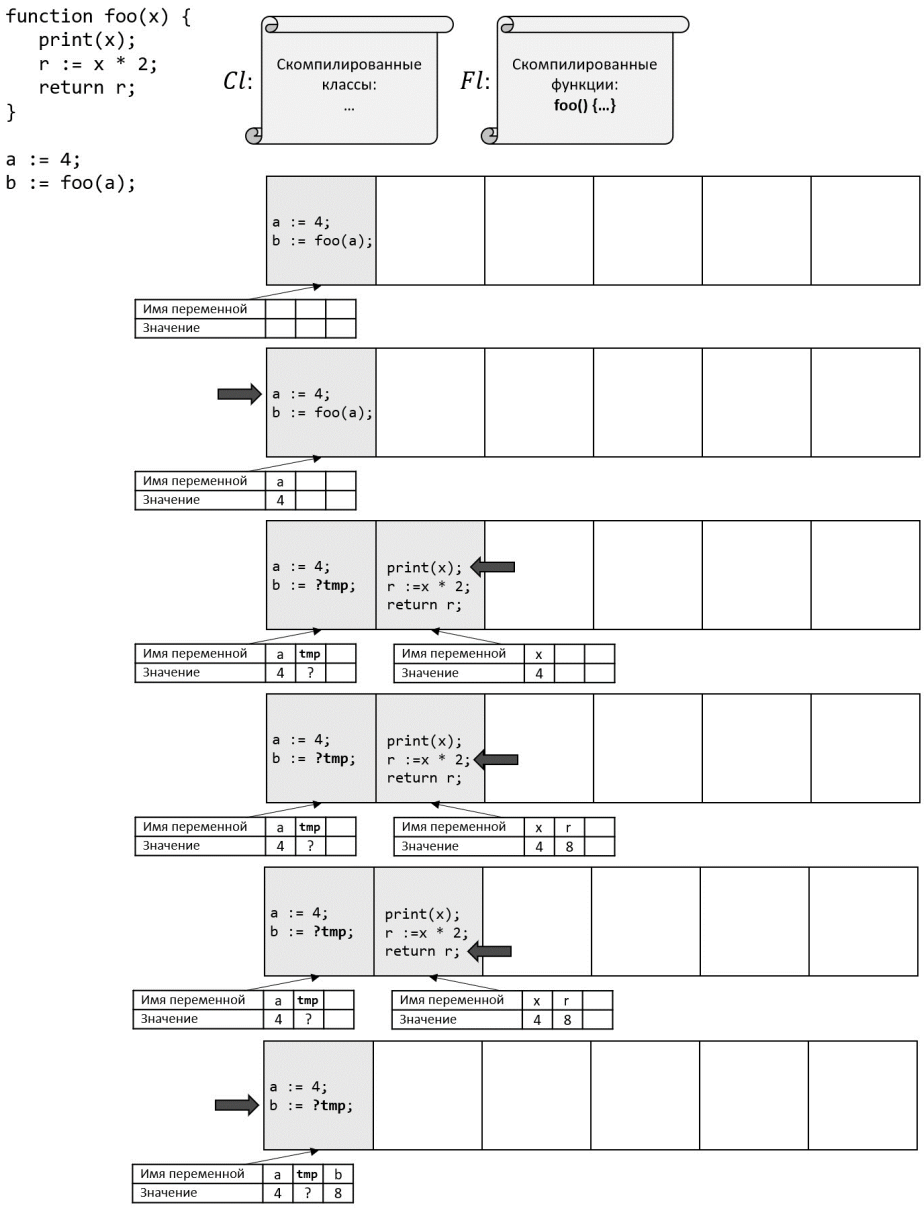


Рис. 3. Пример выполнения L*-программы на виртуальной машине.

Чтобы гарантировать полиномиальную вычислимость и исключить зацикливание программы, требуется запретить явную рекурсию а также неявную, посредством цепочки вызовов, например, foo() → bar() → foo(). Поддержка первого запрета выполняется тривиально, поскольку ячейка содержит информацию о том, в какой функции мы сейчас находимся, и в случае попытки рекурсивного вызова функцией самой себя, выполнение программы завершится с ошибкой. Чтобы запретить неявную рекурсию было введено следующее правило: вызывать L*-определимую функцию в коде можно только тогда, когда ее определение предшествует вызову. Соответственно, если предположить, что функция bar определена раньше функции foo, то мы сможем из функции foo вызвать bar а при попытке вызвать из bar функцию foo выполнение программы завершится с ошибкой.

В распределенной среде на основе блокчейна, с которой мы планируем интегрировать виртуальную машину, программы (смарт-контракты) представляют из себя классы, написанные на языке L*, и содержат атрибуты и методы. Аналогичное представление смарт-контрактов уже используется, например, смарт контракты в Ethereum написаны на объектно-ориентированном языке Solidity [7].

При загрузке смарт-контракта в блокчейн его состояние, т.е. набор атрибутов, инициализируется указанными пользователем начальными значениями посредством вызова конструктора. В блокчейн сохраняется как сам код смарт-контракта там и начальное состояние контракта. Пользователи осуществляют транзакции посредством вызова некоторого метода смарт-контракта, который модифицирует его внутреннее состояние. Результат транзакции в виде списка новых значений атрибутов записывается в блокчейн. Каждый блок в блокчейн-цепи может содержать информацию о нескольких осуществленных транзакциях. Для защиты данных в блокчейне от подмены будет использоваться механизм proof of work, используемый во многих популярных блокчейн-платформах.

Чтобы продемонстрировать возможность интеграции виртуальной машины L* со средой на основе блокчейна, было решено упростить задачу и использовать в качестве блокчейна его эмуляцию в виде простого односвязного списка. Были описаны интерфейсы, посредством которых можно взаимодействовать как с настоящим блокчейном, так и его эмуляцией. Был описан простейший пример смарт-контракта, изображенный на рисунке 4.

```
class TestContract {
   private balances;

   public __constructor__() {
      this.balances := [1000, 0, 0, 0, 0, 0];
   }

   public function transferMoney(from, to, money) {
     this.balances := cond(nth(this.balances, from) >= money ==> do {
                              balances := this.balances;
                              balances := set(balances, from, nth(balances, from)  - money);
                              balances := set(balances, to, nth(balances, to) + money);
                              return balances;
                           }, this.balances);
   }

   public function printBalance(client) {
      print(nth(this.balances, client));
   }
}
```

Рис. 4. Пример тестового смарт-контракта.

Данный смарт-контракт в существенно упрощенном виде демонстрирует пример осуществления операций с некой валютой. В данном случае, предположим, что у нас есть 6 счетов, и счет пользователя идентифицируется его порядковым номером от 0 до 5. Изначально контракт инициализирует счет пользователя №0 с 1000 единицами валюты, а пользователей №1-№5 с 0 единиц валюты. Контракт позволяет переводить валюту от одного пользователя к другому, если у отправителя достаточно средств на счете. Тогда, загрузив смарт контракт в блокчейн и осуществив следующий набор транзакций:

- пользователь №0 переводит 300 единиц валюты пользователю №1,
- пользователь №1 переводит 200 единиц валюты пользователю №3,

в связный список будет записано 4 блока: блок с кодом смарт-контракта, блок с начальным состоянием контракта, по одному блоку на результат каждой из двух произведенных транзакций. Схематично, состояние блокчейн-цепочки показано на рисунке 5.

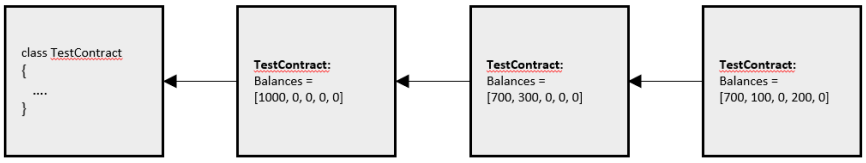


Рис. 5. Цепочка блоков в блокчейне.

Поскольку данный подход показал, что язык L* подходит для описания смарт-контрактов, дальнейшие работы будут вестись в направлении интеграции с настоящей децентрализованной блокчейн системой, например Ethereum.

3 Заключение

В результате проделанной работы был разработан прототип виртуальной машины для исполнения L* программ. Данная разработка перспективна, поскольку позволяет исполнять любые пользовательские программы и смарт-контракты за гарантированно полиномиальное время, что позволит применять язык L* в различных областях. В частности, использование языка L* для написания смарт-контрактов решает проблему их невыполнения из-за превышения порога на количество операций или время выполнения. В дальнейшем планируется продолжить развитие в сторону интеграции виртуальной машины L* с децентрализованными средами на основе блокчейна.

Литература

[1] Goncharov, S.S.; A. Nechesov, A.V. Semantic programming for AI and Robotics, 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Yekaterinburg, Russian Federation, 2022, pp. 810-815, DOI: 10.1109/SIBIRCON56155.2022.10017077.

[2] Goncharov, S.S.; Sviridenko, D.I. Logical Language of Description of Polynomial Computing. Doklady Mathematics, 2019, 485(1), 11-14, DOI:10.31857/S0869-5652485111-14.

[3] Goncharov, S.S.; Nechesov, A.V. Solution of the problem P = L. MDPI Mathematics, 2021, 10(1) 113, DOI:10.3390/math10010113.

[4] Goncharov, S. Nechesov, A. Polynomial analogue of Gandy's fixed point theorem. MDPI Mathematics 2021, 9(17) 2102, DOI:10.3390/math9172102.

[5] Ершов Ю. Л., Палютин Е. А. Математическая логика, 6-е изд., М., Физматлит. – 2011.

[6] Java Compiler Compiler (JavaCC). – URL: https://javacc.github.io/javacc/

[7] Solidity - Solidity 0.8.20 documentation. – URL: https://docs.soliditylang.org/en/v0.8.20/

Продолжения в семантике естественного языка

О.А. Доманов[1]

[1]Институт философии и права СО РАН, ул. Николаева, д. 8, г. Новосибирск, 630090, Россия.

domanov@philosophy.nsc.ru

Аннотация. *Продолжение (англ. continuation) в компьютерных науках представляет собой продолжение программы с некоторого места. В вычислительной семантике естественного языка оно позволяют обеспечить композициональность в случаях, когда она недостижима доступными средствами. В статье демонстрируется использование продолжений при интерпретации естественного языка на примере фрагмента английского языка. Для этого определяется формальный синтаксис фрагмента и строятся функции интерпретации, отображающие выражения языка в язык теории типов. Таким образом, теория типов выступает в качестве онтологического или семантического языка, на котором проводится интерпретация. Формализация проводится в функциональном языке Agda, основанном на теории типов П. Мартин-Лёфа.*

Ключевые слова: вычислительная семантика, естественный язык, теория типов, монада, Agda

В компьютерных науках продолжением (англ. continuation) называется часть программы, продолжающая с определённой точки. В этом смысле, продолжение реализует «оставшуюся часть программы». Оно содержит всю информацию (кроме значений переменных), необходимую для продолжения программы с данной точки. Например, для короткой программы $3 * (2 + 5)$ продолжение с точки, где расположено число 5, может выглядеть как $3 * (2 + [\ ])$, где $[\ ]$ изображает «дырку», в которую следует поместить результат предыдущего вычисления и затем продолжить. В так называемом программировании в стиле передачи продолжений (CPS, Continuation-passing style) функции, вместо того, чтобы возвращать значения, передают их продолжениям для завершения программы. Как оказывается, в семантике естественного языка уже Монтегю [9] использует этот стиль для представления кванторов и собственных имён, не используя само понятие продолжения. В последние годы в работах Баркера, Шена и др. [4; 3] продолжения используются для представления разных языковых явлений, таких как квантификация, учёт контекстов, анафоры и пр. При этом используется язык грамматики Монтегю, то есть λ-исчисления с двумя типами e и t. В данной работе используется теоретико-типовая грамматика [12] вместе с теоретико-категорным понятием монады. Эта грамматика обладает большей выразимостью, по сравнению с грамматикой Монтегю. Что же касается понятия монады, то оно соответствует общему понятию вычисления [8]. Семантика при этом понимается как вычислительная система, вычисляющая смысл. Языком, на котором записывается смысл, выступает теория типов (в варианте языка Agda или единой теории типов UTT, Unified Theory of dependent Types [7]). Поэтому формализация проводится в следующем порядке. Сначала формализуется синтаксис, позволяющий записать в языке Агда выражения естественного языка, а затем строится функция интерпретации как функция из этих выражений в выражения Агды.

Для формализации выбран фрагмент английского языка, позволяющий продемонстрировать использование продолжений. В качестве же языка формализации, как уже сказано, используется функциональный язык Агда [10; 1]. В статье излагаются лишь основные

идеи и подходы, полную формализацию можно найти по адресу https://github.com/odomanov/ttsemantics/tree/v4/Agda/MontagueTT/MontagueTTcontinuation.agda.

1 Формализация синтаксиса и семантики

Начнём с определения монады продолжения. Стандартная библиотека Агды уже содержит эту монаду, но для наглядности определим её самостоятельно.

```
open import Category.Monad

Cont : ∀{a b}(R : Set a) → Set (a ⊔ b) → Set (a ⊔ b)
Cont R A = (A → R) → R

MonadCont : ∀{a b} (R : Set a) → RawMonad (Cont {a} {b} R)
MonadCont R = record { return = λ x k → k x
                     ; _>>=_  = λ m f k → m (λ x → f x k)
                     }
```

Тип `Cont R A` представляет собой функцию, которая при применении к функции продолжения типа `(A → R)` даёт значение типа `R`. Таким образом, имея значение этого типа и «продолжение до `R`» мы можем получить значение `R`. Далее мы будем рассматривать продолжения до предложения, которое само будет интерпретироваться как `Set`, поэтому в качестве `R` везде фигурирует `Set`. Стиль CPS состоит в том, что мы переопределяем все используемые функции так, чтобы они отдавали значения типа `Cont`, что позволяет их далее продолжить тем или иным способом. Такой способ построения семантики позволит нам сохранить принцип композициональности в случаях, когда это оказывается невозможным для семантик, основанных на языке Монтегю.

Мы видим, что монада `MonadCont` содержит две функции, `return : A → Cont R A`, равную $\lambda xk.k(x)$, и `_>>=_ : Cont R A → (A → Cont R B) → Cont R B`, определяющую композицию (подчерки в имени функции обозначают места для её аргументов). Функция композиции равна

$$\lambda mfk.m\big(\lambda x.f(x,k)\big).$$

Если неформально изобразить её действие на продолжения, то получим что-то подобное

$$m[\ ]_A\ >>=\ f_A[\ ] = m[f[\ ]].$$

Это можно понимать как продолжение, которое сначала продолжает способом f, а затем — способом m.

Можно проверить (см. указанный выше файл), что монадные законы для `MonadCont` выполняются.

Перейдём к построению грамматики или синтаксиса. Она будет очень упрощённой и не позволит формализовать многие языковые явления. Например, в ней практически отсутствует отрицание, которое в теории типов вообще требует особого внимания (см., например, [2]). Однако её достаточно для демонстрации работы продолжений в семантике.

Для определения синтаксиса в общем виде зададим лексическую структуру языка:

```
record LexStructure : Set₁ where
  field
    nameCN namePN nameVA nameVI nameVT : Set
    argPN  : namePN  → nameCN
    argVI  : nameVI  → nameCN
    argVT  : nameVT  → nameCN × nameCN
    argVA  : nameVA  → nameCN
```

Эта структура содержит имена для нарицательных имён `CN`, собственных имён `PN`, переходных (`VT`) и непереходных (`VI`) глаголов, а также глаголов пропозициональных установок `VA`, таких как «верить, что…». Ниже почти все из них определяются как типы, зависящие от `CN`, то есть от типа нарицательных имён (англ. common nouns). Функции `arg`X описывают эту зависимость. А именно, имена `CN` будут позднее интерпретироваться как типы (т. е. множества), поэтому, например, для собственных имён `PN` функция `argPN` определяет множества, к которым они относятся, для глаголов `VI`— множества, к которым применимы глаголы, и т. д.

Грамматика определяет синтаксические категории на основе лексической структуры. Большинство из них зависят от `CN`, что означает, что проверка согласования слов производится

на уровне синтаксиса, что исключает правильные синтаксически, но неправильные семантически выражения. Это ограничение нашей грамматики, но оно может быть позднее снято при необходимости.

Синтаксические категории и их конструкторы выглядят следующим образом:

```
data CN : Set where
  cn-n : nameCN → CN

data VI : CN → Set where
  vi-n : (n : nameVI) → VI (cn-n (argVI n))

data VA : CN → Set where
  va-n : (n : nameVA) → VA (cn-n (argVA n))

data VT : CN → CN → Set where
  vt-n : (n : nameVT) → VT (cn-n (proj₁ (argVT n))) (cn-n (proj₂ (argVT n))
)

data PN : CN → Set where
  pn-n : (n : namePN) → PN (cn-n (argPN n))

data DET : Set where
  a/an every no the : DET

data VP : (cn : CN) → Set where
  vp-vi : ∀{cn} → VI cn → VP cn
  vp-vt : ∀{cn1 cn2} → VT cn1 cn2 → NP cn2 → VP cn1

data NP : (cn : CN) → Set where
  np-pn  : (n : namePN) → NP (cn-n (argPN n))
  np-det : DET → (cn : CN) → NP cn

data S : Set where
  s-nvp  : ∀ {cn} → NP cn → VP cn → S
  s-vt   : ∀ {cn1 cn2} → NP cn1 → VT cn1 cn2 → NP cn2 → S
  s-vtp  : ∀ {cn1 cn2} → NP cn2 → VT cn1 cn2 → NP cn1 → S
  s-va   : ∀ {cn1} → NP cn1 → VA cn1 → S → S
```

`DET` обозначает категорию определителей из которых мы рассмотрим артикли `a/an` и `the`, слово `every` и отрицание `no`. Глагольная группа `VP` имеет два конструктора, соответствующих конструированию из непереходного и переходного глаголов. Именная группа `NP` может быть либо собственным именем, либо строиться с помощью определителя. Предложения `S` конструируются с помощью четырёх конструкторов:

- `s-nvp` конструирует предложения вида NP VP, например, «Джон бежит»;
- `s-vt` конструирует предложения вида NP VP NP, например, «Петя любит Машу»;
- `s-vtp` аналогично предыдущему, но в пассивном залоге;
- `s-va` конструирует предложения с глаголами установок вида «NP верит, что S».

Таким образом, мы определили типы всех возможных выражений нашего языка и их конструкторы. Для построения семантики определим тип моделей, содержащих интерпретацию лексической структуры:

```
record Model (nam : LexStructure) : Setω where
  open LexStructure nam
  field
    valCN  : nameCN → Set
    valPN  : (n : namePN)  → valCN (argPN n)
    valVI  : (n : nameVI)  → valCN (argVI n) → Set
    valVT  : (n : nameVT)  → valCN (proj₁ (argVT n))
                           → valCN (proj₂ (argVT n)) → Set
    valVA  : (n : nameVA)  → (h : valCN (argVA n))
                           → ∀{a}(A : Set a)(P : A → Set) → Set a
```

Эта структура определяет, как должны интерпретироваться имена `name`*X*. Имена `CN` (т. е. `nameCN`) интерпретируются как типы, `PN`— как элементы типов `valCN (argPN n)`, глаголы `VI` и `VT`— как типы, зависящие от значений, определённых функциями `argVI` и `argVT`. Глаголы `VA` мы рассмотрим подробнее ниже.

Для интерпретации мы определим не одну функцию, а по одной функции для каждой синтаксической категории. Будем обозначать их ⟦`cn_`⟧, ⟦`pn_`⟧ и т. д. для категорий `CN`, `PN` и т. д. Эти функции переводят выражения естественного языка (уже представленные в описанном выше формализме) в выражения языка Агда. Последний, таким образом, выступает как семантически «уже понятный» язык.

Рассмотрим последовательно, как выглядят функции интерпретации для всех категорий. Выражения категории `CN` интерпретируются как типы или множества. Они берутся из модели `Model`:

```
⟦cn_⟧ : CN → Set
⟦cn cn-n n ⟧ = valCN n
```

Такая интерпретация отличается от грамматики Монтегю, где `CN` представляются предикатами или функциями вида $e \to t$ (см. подробнее о сравнении этих двух подходов в [5]).

Интерпретации собственных имён также берутся непосредственно из модели:

```
⟦pn_⟧ : {cn : CN} → PN cn → ⟦cn cn ⟧
⟦pn pn-n n ⟧   = valPN n
```

Все следующие категории интерпретируются как продолжения. Мы рассматриваем только продолжения до `Set`, то есть до предложений. Для этого функции вида `A → B` мы заменяем на продолжения вида `A → Cont Set B`, т. е. заменяем их значения на функции `(B → Set) → Set`. Ниже они, как правило, записываются как функции вида `λ k →` X .

Именные группы `NP` являются 0-местными функциями, поэтому они интерпретируются как `Cont Set A` для некоторых `A`:

```
⟦np_⟧ : {cn : CN} → NP cn → Cont Set ⟦cn cn ⟧
⟦np np-pn pn ⟧ = return ⟦pn pn-n pn ⟧
⟦np np-det d cn ⟧ = ⟦det d ⟧ cn
```

Таким образом, в случае именной группы, состоящей только из собственного имени, интерпретация даётся функцией `return`, а в случае группы с определителем (например, a girl, every boy и т. д.) — значением определителя, которое само является продолжением (см. ниже).

Значения глаголов `VI` и `VT` также даются просто функцией `return`:

```
⟦vi_⟧ : {cn : CN} → VI cn → ⟦cn cn ⟧ → Cont Set Set
⟦vi vi-n n ⟧ x = return (valVI n x)

⟦vt_⟧ : ∀ {cn cn1} → VT cn cn1 → ⟦cn cn ⟧ → ⟦cn cn1 ⟧ → Cont Set Set
⟦vt vt-n n ⟧ x y = return (valVT n x y)
```

Глагольная группа имеет два конструктора. Значение для первого совпадает со значением для `VI`, а для второго, состоящего из переходного глагола, применённого к именной группе, вычисляется как монадная композиция значений глагола и группы:

```
⟦vp_⟧ : {cn : CN} → VP cn → ⟦cn cn ⟧ → Cont Set Set
⟦vp vp-vi vi ⟧ = ⟦vi vi ⟧
⟦vp vp-vt vt np ⟧ x = do y ← ⟦np np ⟧
                         ⟦vt vt ⟧ x y
```

Интерпретация определителей структурно совпадает с определениями Монтегю, хотя он сам не употреблял понятие продолжения. Аналогично его грамматике неопределённый артикль соответствует экзистенциальному, а слово every — универсальному квантору:

```
⟦det_⟧ : DET → (cn : CN) → Cont Set ⟦cn cn ⟧
⟦det a/an ⟧  cn = λ k → Σ[ x ∈ ⟦cn cn ⟧ ] k x
⟦det every ⟧ cn = λ k → ∀(x : ⟦cn cn ⟧) → k x
⟦det no ⟧    cn = λ k → ∀(x : ⟦cn cn ⟧) → ¬ k x
⟦det the ⟧   cn = λ k → Σ[ Aₚ ∈ Pointed ⟦cn cn ⟧ ] k (theₚ Aₚ)
```

Для интерпретации «the» здесь используется тип с выделенным элементом, обозначенным здесь $\texttt{the}_\texttt{p}$:

```
record Pointed (A : Set) : Set where
  field
    the_p : A
```

Таким образом, интерпретация ⟦det the ⟧ означает: «Существует тип с выделенным элементом и этот выделенный элемент удовлетворяет k».

При интерпретации предложений будем предполагать, что это может происходит разными способами, поэтому значением предложения будет список значений. Для краткости, рассмотрим пока только предложения, конструируемые с помощью s-nvp и s-vt:

```
⟦s_⟧ : S → List (Cont Set Set)
⟦s s-nvp np vp ⟧ = (do x ← ⟦np np ⟧
                       ⟦vp vp ⟧ x)
                  ∷ []
⟦s s-vt np1 vt np2 ⟧ = (do x ← ⟦np np1 ⟧
                           y ← ⟦np np2 ⟧
                           ⟦vt vt ⟧ x y)
                       ∷ (do y ← ⟦np np2 ⟧
                           x ← ⟦np np1 ⟧
                           ⟦vt vt ⟧ x y)
                       ∷ []
```

Первое из них не имеет множественной интерпретации и вычисляется как монадная композиция значений для NP и VP. Второе может иметь две возможные интерпретации, которые, как видно, различаются порядком вычисления значений для именных групп. Как мы увидим ниже, в некоторых случаях они позволяют различить интерпретации *de re* и *de dicto*.

У нас осталась ещё интерпретация глаголов VA, но мы отложим её на время и обратимся к примерам.

1.1 Примеры

Прежде всего, зададим лексическую структуру фрагмента языка, т. е. типы name$X$ и функции arg$X$.

```
data nameCN  : Set where Human Unicorn : nameCN
data namePN  : Set where Alex Mary John Ralph Ortcutt : namePN
data nameVI  : Set where run is-spy : nameVI
data nameVA  : Set where believe strive : nameVA
data nameVT  : Set where love find : nameVT

argPN : namePN → nameCN
argPN Alex = Human
argPN Mary = Human
argPN John = Human
argPN Ralph = Human
argPN Ortcutt = Human

argVI : nameVI → nameCN
argVI run = Human
argVI is-spy = Human

argVA : nameVA → nameCN
argVA believe = Human
argVA strive = Human

argVT : nameVT → nameCN × nameCN
argVT love = Human , Human
argVT find = Human , Unicorn
```

Затем определим некоторые предикаты, на которых будем проводить интерпретацию. Условимся, что звёздочкой в начале будем обозначать то, что относится к онтологии — объекты, функции на них и пр.

```
postulate
  *Human *Unicorn : Set
  *Alex *Mary *John *Ralph *Ortcutt : *Human
  _*run    : *Human → Set
  _*is-spy : *Human → Set
  _*find_  : *Human → *Unicorn → Set
  _*love_  : *Human → *Human → Set
```

Далее нам нужно задать модель, то есть интерпретацию базовой лексики:

```
valCN : nameCN → Set
valCN Human = *Human
valCN Unicorn = *Unicorn

valPN : (n : namePN) → valCN (argPN n)
valPN Alex = *Alex
valPN Mary = *Mary
valPN John = *John
valPN Ralph = *Ralph
valPN Ortcutt = *Ortcutt

valVI : (n : nameVI) → valCN (argVI n) → Set
valVI run = _*run
valVI is-spy = _*is-spy

valVT : (n : nameVT) → valCN (proj₁ (argVT n))
      → valCN (proj₂ (argVT n)) → Set
valVT love = _*love_
valVT find = _*find_
```

После задания интерпретации базовой лексики перейдём к сложным выражениям.

Рассмотрим предложение «Mary runs»:

```
s1 = s-nvp (np-pn Mary) (vp-vi (vi-n run))
```

Определим функцию для извлечения значения из предложения (которое, как мы помним, представляет собой список):

```
_[_] : (s : S) → Fin (length ⟦s s ⟧) → Set
s [ n ] = (lookup ⟦s s ⟧ n) id
```

Здесь `lookup`— стандартная функция извлечения элемента номер `n` из списка. Таким образом, `s [ # n ]` означает n-ое значение для предложения `s`. Тогда Агда вычисляет интерпретацию `s1` следующим образом:

```
s1 [ # 0 ] ≡ *Mary *run
```

(здесь `≡` означает пропозициональное, т. е. доказываемое равенство, в отличие от равенства по определению).

Аналогично для предложений «A human runs» и «Every human runs»:

```
s4 = s-nvp (np-det a/an (cn-n Human)) (vp-vi (vi-n run))
s4 [ # 0 ] ≡ Σ *Human _*run

s5 = s-nvp (np-det every (cn-n Human)) (vp-vi (vi-n run))
s5 [ # 0 ] ≡ ∀(x : *Human) → x *run
```

Как и можно было ожидать, первое интерпретируется как экзистенциальная, а второе — как универсальная пропозиция.

Для «The human runs» имеем:

```
s6 = s-nvp (np-det the (cn-n Human)) (vp-vi (vi-n run))
s6 [ # 0 ] ≡ Σ[ Aₚ ∈ Pointed *Human ] (theₚ Aₚ) *run
```

Если мы постулируем существование соответствующих доказательств, то можем получить и доказательство пропозиции s6, которое равно:

```
proof ≡ Hₚ , *Mary-runs
  where
  Hₚ : Pointed *Human
  Hₚ = record { theₚ = *Mary }

  postulate *Mary-runs : *Mary *run
```

Рассмотрим более сложное двусмысленное предложение «Every human loves a human»:

```
s13 = s-vt (np-det every (cn-n Human)) (vt-n love)
           (np-det a/an (cn-n Human))

s13 [ # 0 ] ≡ (∀(x : *Human) → Σ[ y ∈ *Human ] (x *love y))
s13 [ # 1 ] ≡ (Σ[ y ∈ *Human ] ∀(x : *Human) → (x *love y))
```

Мы получили *de re* и *de dicto* интерпретации, которые, как мы видим, различаются порядком вычисления продолжений в монаде. Продолжения позволяют нам, во-первых, регулировать этот порядок, а во-вторых, корректно проводить квантификацию в контекстах.

2 Глаголы пропозициональных установок

Глаголы VA требуют особого рассмотрения. Прежде всего, они должны быть интерпретированы так, чтобы допускать квантификацию в контекст пропозиции, на которую они направлены. Рассмотрим, например, глагол believe. Определим его следующим образом:

```
record believeR {H : Set}(h : H){a}(A : Set a)(P : A → Set) : Set a where
  field
    s1 : A
    s2 : P s1
```

Это тип, который имеет следующий смысл: «h верит относительно элемента A, что P» (ср. [11, p. 180]). Агда позволяет ввести для этого более наглядное обозначение, которым я и буду далее пользоваться:

```
believeR h A (λ x → P) = h believe[ x ∈ A ] P
```

Добавим аналогичные определения для striveR и определим интерпретации для модели:

```
valVA : (n : nameVA)(h : valCN (argVA n))
      → ∀{a}(A : Set a)(P : A → Set) → Set a
valVA believe = believeR
valVA strive = striveR
```

Это позволяет нам добавить функцию интерпретации для VA:

```
_⟦va_⟧ : ∀{cn1} → ⟦cn cn1 ⟧ → VA cn1 → (A : Set) → Cont Set A
(h ⟦va (va-n n) ⟧) A = λ k → valVA n h A (λ x → k x)
```

Выражение h ⟦va (va-n believe) ⟧ A означает «h верит относительно элемента A, что…». Интерпретация предложения ⟦s s-va np va S ⟧ должна быть построена для разных S, но мы сейчас ограничимся только предложениями вида s-nvp и s-vt и определим:

```
⟦s s-va np1 va (s-nvp {cn2} np2 vp2) ⟧ =
      (do x ← ⟦np np2 ⟧
          y ← ⟦np np1 ⟧
          (y ⟦va va ⟧) ⟦cn cn2 ⟧
          ⟦vp vp2 ⟧ x)
    ∷ (do y ← ⟦np np1 ⟧
          x ← (y ⟦va va ⟧) ⟦cn cn2 ⟧
          ⟦vp vp2 ⟧ x)
    ∷ []
⟦s s-va np1 va (s-vt {cn2} {cn3} np2 vt np3) ⟧ =
     (do z ← ⟦np np3 ⟧
         x ← ⟦np np1 ⟧
```

```
        y ← ⟦np np2 ⟧
        (x ⟦va va ⟧) ⟦cn cn3 ⟧
        ⟦vt vt ⟧ y z)
  ∷ (do x ← ⟦np np1 ⟧
        y ← ⟦np np2 ⟧
        (x ⟦va va ⟧) ⟦cn cn3 ⟧
        z ← ⟦np np3 ⟧
        ⟦vt vt ⟧ y z)
  ∷ []
```

Таким образом, мы полагаем, что предложения такого вида имеют по две возможных интерпретации. Посмотрим на примере, чем они различаются. Предложение Куайна [11] «Ralph believes someone is a spy (Ральф верит, что кто-то шпион)» выглядит следующим образом:

```
s19 = s-va (np-pn Ralph) (va-n believe)
           (s-nvp (np-det a/an (cn-n Human)) (vp-vi (vi-n is-spy)))
```

Оно имеет две интерпретации:

```
s19 [ # 0 ] ≡ Σ[ x ∈ *Human ] (*Ralph believe[ _ ∈ _ ] x *is-spy)
s19 [ # 1 ] ≡ *Ralph believe[ x ∈ *Human ] x *is-spy
```

Первую из них словами можно передать как «Существует человек, о котором Ральф верит, что он шпион», а вторую как «Ральф верит, что существуют шпионы». Как мы видим, продолжения позволяют получать доступ к контекстам, таким как система убеждений Ральфа.

Рассмотрим ещё один пример. Для формализации двусмысленного предложения Монтегю «John seeks a unicorn» («Джон ищет (какого-то) единорога») представим его как «John strives that he finds a unicorn» («Джон добивается, чтобы он нашёл (какого-то) единорога») (ср. [11]):

```
s30 = s-va (np-pn John) (va-n strive)
           (s-vt (np-pn John) (vt-n find) (np-det a/an (cn-n Unicorn)))

s30 [ # 0 ] ≡ Σ[ x ∈ *Unicorn ] (*John strive[ _ ∈ _ ] *John *find x)
s30 [ # 1 ] ≡ *John strive[ _ ∈ _ ] (Σ[ x ∈ *Unicorn ] *John *find x)
```

Первая интерпретация утверждает, что Джон ищет какого-то конкретного единорога (из «нашего» контекста), тогда как вторая — что он ищет всё равно какого единорога (из контекста Джона).

3 Заключение

Таким образом, продолжения позволяют нам регулировать порядок семантических вычислений, имея доступ к области действия (scope) операторов квантификации и пропозициональных глаголов. Основным преимуществом продолжений является то, что они позволяют это сделать композициональным образом. Нужно заметить, однако, что, хотя мы имеем возможность регулировать порядок вычислений, это не решает проблему непрозрачности контекстов. Если Ральф знает некоего Орткута в разных ситуациях, не подозревая о том, что это один и тот же человек, причём в одних из них считает его шпионом, а в других — нет (пример Куайна из [11]), то предложение «Ральф верит, что Орткут шпион»:

```
s21 = s-va (np-pn Ralph) (va-n believe)
           (s-nvp (np-pn Ortcutt) (vp-vi (vi-n is-spy)))

s21 [ # 0 ] ≡ (*Ralph believe[ _ ∈ _ ] *Ortcutt *is-spy)
```

не имеет определённого доказательства, т. е. значения истинности. Наших инструментов недостаточно для выражения заключённой здесь двусмысленности. Хотя Агда допускает, как мы видели, доступ к внутренним контекстам, она не позволяет его регулировать в достаточной степени, например, устанавливать сложные связи между переменными вне и внутри этих контекстов. Вместе с тем, уже Фреге указывает, что непрозрачные контексты полагают границы, при пересечении которых референция терминов может изменяться (см. подробнее [6, p. 185 sq.]). Это одно из ограничений, связанных с тем, что Агда создавалась в ориентации преимущественно на математику, а не семантику естественного языка. Она, тем не менее, может, как мы видим, найти своё применение в моделировании некоторых языковых явлений.

Литература

[1] Agda Documentation. URL: https://agda.readthedocs.io/.

[2] *Agudelo-Agudelo J. C.*, *Sicard-Ramírez A.* Type Theory with Opposite Types: A Paraconsistent Type Theory // Logic Journal of the IGPL. 2021. Vol. 30, no. 5. P. 777–806.

[3] *Barker C.* Continuations and the Nature of Quantification // Natural Language Semantics. 2002. Vol. 10, no. 3. P. 211–242.

[4] *Barker C.*, *Shan C.-c.* Continuations and Natural Language. Oxford : OUP, 2014.

[5] *Chatzikyriakidis S.*, *Luo Z.* On the Interpretation of Common Nouns: Types Versus Predicates // Modern Perspectives in Type-Theoretical Semantics. Springer International Publishing, 2017. P. 43–70.

[6] *Kaplan D.* Quantifying In // Synthese. 1968. Vol. 19, no. 1/2. P. 178–214. JSTOR: 20114638.

[7] *Luo Z.* Computation and Reasoning : A Type Theory for Computer Science. OUP, 1994.

[8] *Moggi E.* Notions of Computation and Monads // Information and Computation. 1991. Vol. 93, no. 1.

[9] *Montague R.* The Proper Treatment of Quantification in Ordinary English // Formal Philosophy : Selected Papers of Richard Montague. New Haven, London : Yale University Press, 1974. P. 247–270.

[10] *Norell U.* Dependently Typed Programming in Agda // Advanced Functional Programming: 6th International School, AFP 2008, Heijen, The Netherlands, May 2008, Revised Lectures. Berlin, Heidelberg : Springer-Verlag, 2009. P. 230–266.

[11] *Quine W. V. O.* Quantifiers and Propositional Attitudes // Journal of Philosophy. 1956. Vol. 53, no. 5. P. 177–187.

[12] *Ranta A.* Type-Theoretical Grammar. Clarendon Press, 1994. 226 p.

Знания-Онтологии-Теории (ЗОНТ-2023)

Автоматизированные методы проведения собеседований

Зайцев А.О., Пальчунов Д.Е., Трегубов А.С.

Факультет Информационных Технологий, Новосибирский государственный университет, ул. Пирогова, д. 1, г. Новосибирск, 630090, Россия.

a.zaitsev3@g.nsu.ru, d.palchunov@g.nsu.ru, tematre@gmail.com

Аннотация: Статья посвящена разработке автоматизированных методов ведения диалога с пользователем для проведения собеседований. Целью данной работы является разработка модуля управления процессом и содержанием первичного собеседования. Предложенный метод определения знаний, навыков кандидатов, а также оценки их компетенций основан на взаимодействие с онтологической моделью предметной области, и при этом не требует участия технических специалистов. В ходе работы, благодаря использованию шаблонов проектирования, была заложена гибкая архитектура позволяющая внедрение различных предметных областей, типов заданий, а также использование разных платформ для интеграции. В дальнейшем планируется развитие данной работы за счет наполнения онтологической модели и использования различных видов заданий.

Ключевые слова: диалог с пользователем, первичное собеседование, онтологическая модель, порождение знаний, ассоциативные правила.

1. Введение

Каждый, кто хоть раз проводил отбор кандидатов, знает, как много времени и сил уходит впустую при проведении первичного этапа собеседования. К тому же, человек, проводящий собеседование, может необъективно оценить знания и отсечь потенциально хорошего специалиста. Таким образом, все эти факторы приносят компании убытки и говорят о низкой эффективности HR-процессов.

Поэтому для решения данных проблем было предложено создать платформу, позволяющую проведение автоматизированного первичного тестирования знаний, навыков кандидатов, а также оценки их компетенций. Система собирает ответы кандидатов, анализирует их корректность, правильность и точность в сравнении с эталонным ответом, выдавая итоговую оценку, состоящую из: общей оценки знаний, навыков; оценки компетенции; психологической оценки. Также платформа позволяет самостоятельно управлять процессом и содержанием собеседования, автоматизировать сбор, исправление и категоризацию заданий для собеседования. Участие технических специалистов предполагается только для выполнения первичной настройки и одобрения/отклонения заявки кандидата.

Целью данной работы является разработка автоматизированного метода ведения диалога с пользователем, направленного на определение компетенций совместно с установлением знаний кандидата в той или иной областях.

В ходе решения данной задачи знания о предметной области были представлены в виде онтологического графа. Разработаны ассоциативные правила, позволяющие на основе онтологической модели пользователя делать дополнительные выводы о его знаниях в различных областях. Заложена гибкая архитектура, независящая от конкретных предметных областей, а также позволяющая внедрение разнообразных видов заданий и использование различных платформ для интеграции.

2. Обзор литературы

На сегодняшний день существует множество способов извлечения знаний о пользователе. Использование конкретного метода или комбинации методов напрямую зависит от поставленных интервьюеру целей и задач. Тем не менее, наиболее распространёнными являются следующие решения:

1. Анкетирование.
2. Наблюдение.
3. Интервью.
4. Свободный диалог.
5. Тестовое задание.

Очевидно, что, исходя из поставленной цели, необходимо рассматривать только автоматизированные методы – анкетирование, интервью и свободный диалог.

Самым популярным и простым классическим решением задачи является метод онлайн анкетирования. Однако, несмотря на широту использования, оно обладает рядом существенных недостатков, одним из которых является большое влияние качества составленной анкеты. Поэтому было предложено улучшить данный подход путем использования комбинации методов анкетирование и интервью. Такой выбор позволит не только построить модель пользователя, но также иметь возможность управлять процессом и содержанием диалога с ним.

Для структурирования знаний о предметной области, в связи с требованием машиночитабельности, было решено рассмотреть методы «Сущность-связь» и онтологические модели.

При выборе метода «Сущность-связь» разработчики будут вынуждены чаще прибегать к использованию вспомогательных сущностей, не имеющих прямого отношения к предметной области [1-4]. Это может привести не только к низкой эффективности построенной схемы, но и к ухудшению производительности платформы в целом

Исходя из этого, было решено использовать метод онтологического моделирования. Однако, важно заметить, что большая часть современных онтологических моделей описывает предметную область в двух уровнях [5-8]. Поэтому было предложено использовать подход [9,10], улучшающий онтологическую модель, путем использования четырех уровней описания предметной области:

1. Онтология. Содержит наборы и определения ключевых понятий, на языке которых описывается данная предметная область. Этот уровень определяет смысл понятий и отношения между понятиями. Например, «является» или «целое-часть».
2. Универсальные, общие утверждения. Представлен в виде знаний о предметной области: общие принципы, законы, постулаты, закономерности. Должны быть полностью достоверны на данный момент времени. Со временем эти знания могут изменяться даже в случае, когда смысл употребляемых терминов остается полностью неизменным.
3. Прецеденты. Являются эмпирическими знаниями о предметной области. Представлены в виде набора знаний, верных только для какой-то конкретной ситуации в конкретный момент времени.
4. Вероятностные или оценочные знания. Представлен в виде набора знаний, которые берутся из любых внешних источников или же порождаются на основе знаний, представленный в онтологической модели. Для генерации оценочных знаний предложен алгоритм построения гипотез на основе базы прецедентов.

Важным преимуществом, предложенной модели является возможность адаптации под любую необходимую предметную область. Итоговая структура модели представлена на рис. 1:

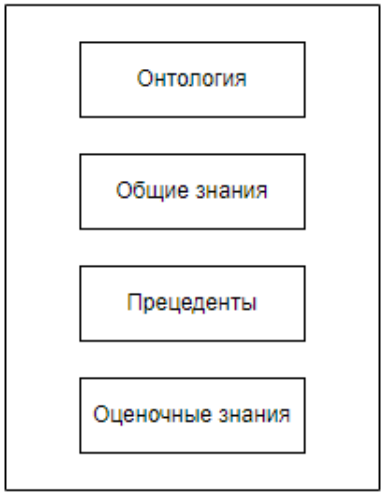


Рис. 1. Структура четырехуровневой модели предметной области.

В качестве инструмента для создания модели пользователя были использованы когнитивные карты [11, 12]. Каждая из таких карт формализует представления кандидата о некотором понятии в виде графа, представляющего собой соответствующий подграф семантической сети. Сопоставление знаний кандидата к некоторым понятием предметной области сводится к сравнению указанного подграфа семантической сети и графа, который определяет соответствующая когнитивная карта.

В конечном итоге модель интеллектуального помощника можно разделить на три основных уровня, за каждым из которых закреплена уникальная ответственность:

- Предметная область. Содержит четырехуровневую онтологическую модель, а также шаблоны правил логического вывода.
- Модель пользователя. Содержит общую информацию о модели пользователя, занимается ее построением и дополнением.
- Представление о диалоге. Содержит информацию о состоянии текущего диалога с пользователем.

3. Предлагаемое решение

Поскольку разрабатываемое решение является частью большой платформы, то для получения пользовательских ответов необходимо создать универсальный сервис, обеспечивающий простую интеграцию с любыми сторонними приложениями. Его разработку было решено выполнить на основе взаимодействия с Telegram.

В целях обеспечения гибкости и простоты расширяемости системы, необходимо избежать жёсткой привязанности полученного внешнего сообщения к обработчику. Для построения подобного взаимодействия был использован поведенческий шаблон проектирования «Цепочка обязанностей» (Chain of responsibility).

Принцип действия данного шаблона заключается в том, что возможные обработчики образуют своеобразный конвейер. Данные запроса передаются в качестве аргументов метода от одного объекта к другому до тех пор, пока не закончиться цепочка (рис. 2). Каждое звено при выполнении выбирает: либо закончить обработку запроса; либо передать запрос следующему объекту.

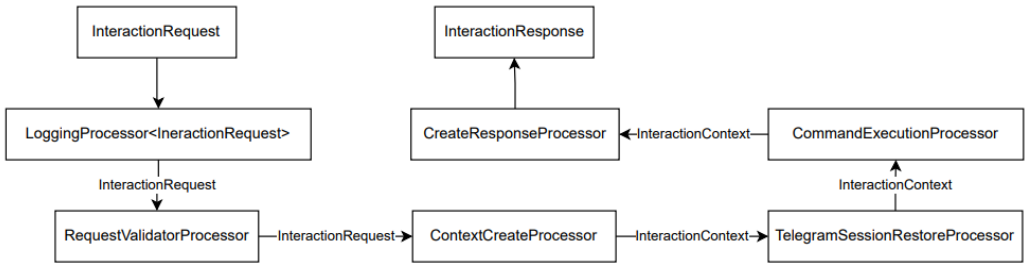


Рис. 2. Схема взаимодействия построенной цепочки.

Для восстановления текущего состояния собеседования было решено использовать шаблон проектирования архитектуры приложения «Модель-Представление-МодельПредставление» (рис. 3), в котором:

- Модель – описание фундаментальных данных для отображения элемента;
- Представление – графический интерфейс. Содержит информацию о том, как отобразить конкретную модель;
- МодельПредставление – обертка данных из модели. Может выполнять команды, которые меняют текущее состояние модели;

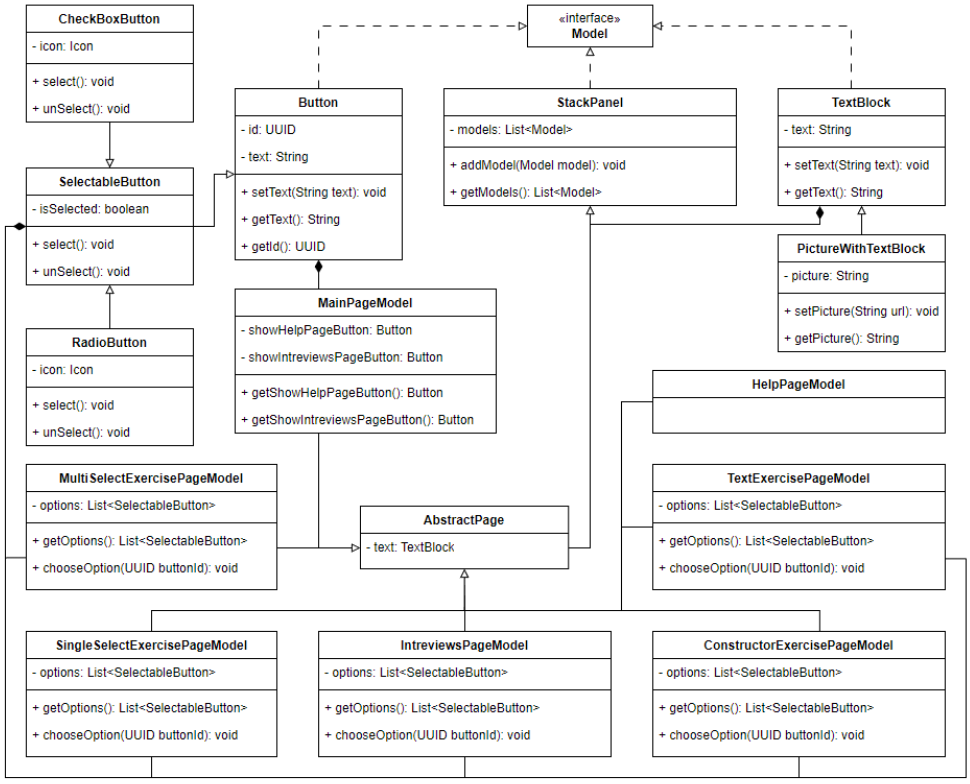


Рис. 3. Реализация шаблона проектирования «Модель-Представление-МодельПредставление».

После построения восстановления состояния собеседования необходимо реализовать систему построения ответа от сервера. Основная проблема подобного взаимодействия заключается в динамичности диалога с кандидатом: у каждого пользователя, использующего платформу, имеется свой сценарий прохождения собеседования, который не только заранее неизвестен, но и может изменяться по ходу продвижения интервью.

Для решения данной проблемы был использован поведенческий паттерн проектирования «Команда» (рис. 4). Он превращает запрос на выполнение действия и его параметры в отдельный объект-команду. Этот объект имеет связь с объектом-обработчиком. Использование подобной схемы взаимодействия позволяет уменьшить зависимость между классом-отправителем и классом-получателем.

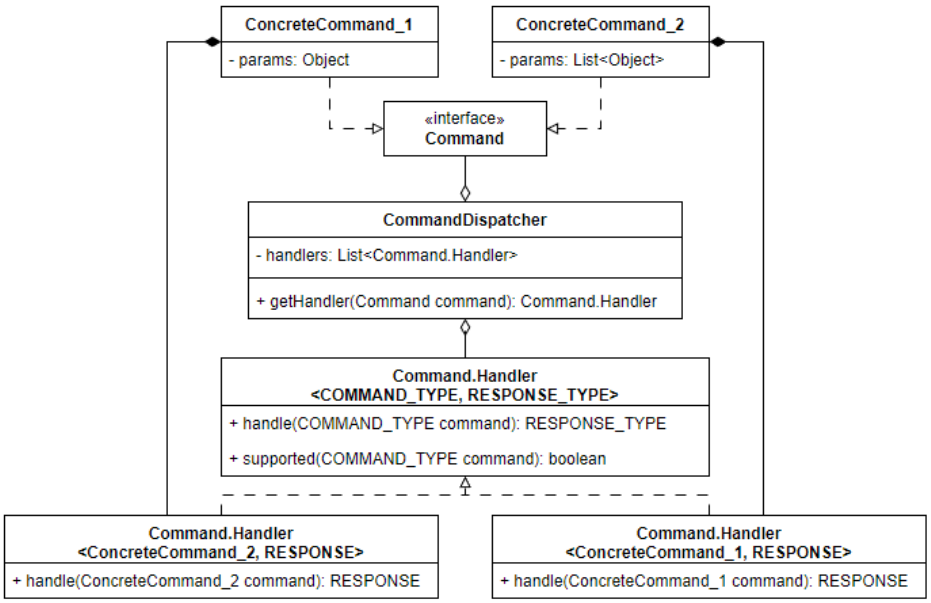


Рис. 4. Реализация паттерна «Команда».

Для осуществления: привязки команд к диалогу; восстановления команд, необходимых для выполнения; получения последней, выполняемой в диалоге с пользователем команды, было предложено воспользоваться реляционной базой данных. Однако, чтобы раз в несколько минут не вытаскивать команды из базы данных с целью обновления их атрибутов, команды были разделены на два семейства:

1. Базовые команды – семейство команд, которое преобразовывает модель в единый ответ сервера.
2. Команды помощники или инициализаторы – семейство команд, которые инициализируют модель для команд из первого семейства.

После реализации инфраструктуры взаимодействия со сторонним программным обеспечением, необходимо составить алгоритм выбора задания. Очевидно, что каждое собеседование обладает своим набором компетенций, требуемых для проверки, поэтому было решено использовать последовательный подход: переходить от одной компетенции к другой, только после завершения проверки первой из них. Выбор первого задания производится с помощью генератора случайного элемента коллекции. Для выбора последующих заданий используются данные о связанных узлах графа: необходимо выбрать задание, имеющее наибольшее количество связанных вершин, не содержащихся в модели пользователя.

Критерием перехода к следующей компетенции является выполнение одного из двух условий:

1. Если кандидат ответил на минимальное количество заданий и процент его знаний (то есть количество вершин, помеченных как «знает», деленное на общее количество вершин) не изменился больше чем на заранее определенный коэффициент ε. Это условие позволяет перестать выбирать задания, относящие к определенной компетенции, в случаях: если кандидат имеет превосходные знания в данной категории или если кандидат не имеет ни малейшего представления о данной категории;
2. Если процент покрытия компетенции (то есть количество вершин, хранящихся в модели пользователя, деленное на общее количество вершин в компетенции) больше заранее определенного значения φ;

Инициализация модели пользователя выполняется в два этапа: создание дополнительного свойства «знает» (рис. 5); наполнение модели правилами для машин логического вывода (рис. 6).

```
36      private OWLDataProperty addHasToKnowProperty(OWLOntology clearOntology) {
37          OWLDataFactory dataFactory = owlOntologyManager.getOWLDataFactory();
38          OWLDataProperty hasToKnowProperty = dataFactory
39                  .getOWLDataProperty(IRI.create(BASE_IRI + "#hasToKnow"));
40          OWLDatatype booleanDatatype = dataFactory.getBooleanOWLDatatype();
41          OWLDataPropertyRangeAxiom rangeAxiom = dataFactory.getOWLDataPropertyRangeAxiom(hasToKnowProperty,
42                  booleanDatatype);
43          owlOntologyManager.addAxiom(clearOntology, rangeAxiom);
44          return hasToKnowProperty;
45      }
```

Рис. 5. Листинг создания дополнительного свойства «знает».

Для осуществления гибкой конфигурации аксиом используется реляционная база данных: для каждого свойства, описывающего связи между объектами хранятся строки вида <Node, Boolean, Boolean>. Первое значение типа Boolean означает вывод, сделанный на основе ответа пользователя, второе – вывод, который должен быть получен на основе машин логического вывода.

```
47      private void fillOntologyAxiom(OWLOntology clearOntology, OWLDataProperty hasToKnowProperty) {
48          OWLDataFactory dataFactory = owlOntologyManager.getOWLDataFactory();
49          for (OWLObjectProperty objectProperty : baseOntology.getObjectPropertiesInSignature()) {
50              for (var property : ontologyService.findByOwlObjectProperty(objectProperty)) {
51                  OWLObjectSomeValuesFrom someValuesFrom = dataFactory
52                          .getOWLObjectSomeValuesFrom(objectProperty, dataFactory.getOWLThing());
53                  OWLClassExpression condition = dataFactory
54                          .getOWLObjectIntersectionOf(
55                                  dataFactory.getOWLDataHasValue(hasToKnowProperty,
56                                          dataFactory.getOWLLiteral(property.getIfValue())),
57                                  someValuesFrom);
58                  OWLSubClassOfAxiom subClassAxiom = dataFactory
59                          .getOWLSubClassOfAxiom(condition,
60                                  dataFactory.getOWLDataHasValue(hasToKnowProperty,
61                                          dataFactory.getOWLLiteral(property.getThenValue())));
62                  owlOntologyManager.addAxiom(clearOntology, subClassAxiom);
63              }
64          }
65      }
```

Рис. 6. Листинг наполнения модели правилами логического вывода.

Получив ответ, используя связь с заданиями, узлы модели пользователя помечаются как «знает» или «не знает». Также, используя машины логического вывода, извлекаются дополнительные знания о связанных вершинах в графе, что позволяет расширить модель пользователя.

Затем вызывается алгоритм выбор следующего вопроса, и, возможно, выбор следующей компетенции. Точкой выхода из подобного алгоритма является завершение обхода всех компетенций.

В качестве модели предметной области было решено использовать онтологическую модель знаний об языках программирования Java, Python и C# [13], совместно с моделью знаний об объектно-ориентированном программировании. Связь между заданиями и онтологической моделью осуществляться за счет реляционной базы данных.

4. Результаты

Проделанная работа позволяет завершить сборку всей системы, а также дает возможность произвести полноценное тестирование (рис. 7).

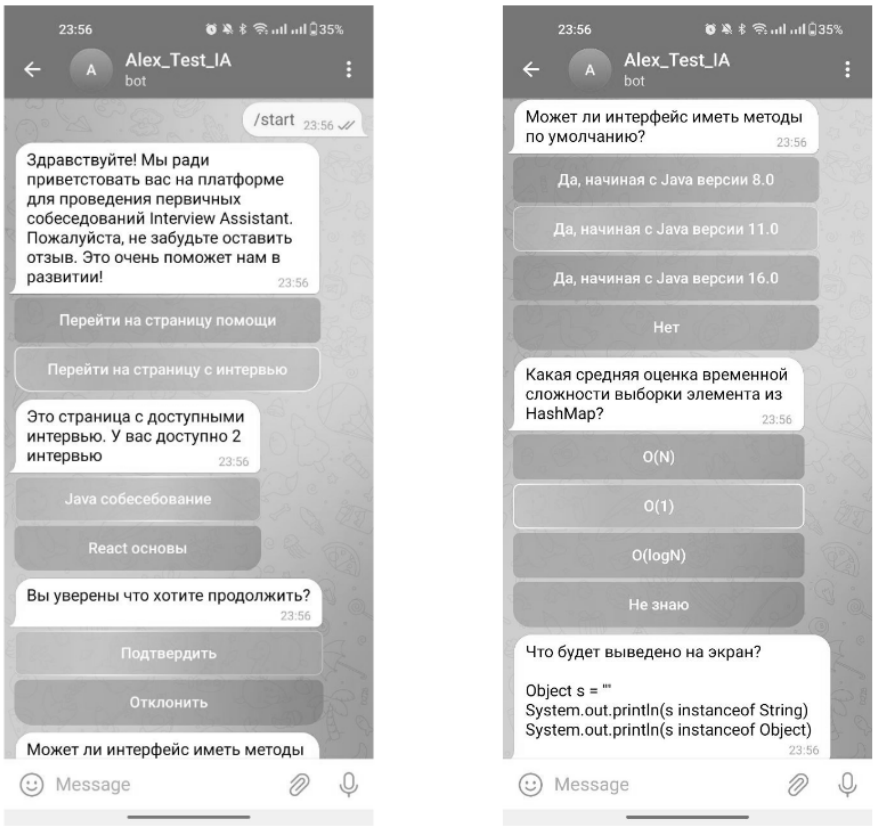


Рис. 7. Пример работы системы.

Учитывая важность задачи создания платформы для автоматизации проведения первичного собеседования, дальнейшее развитие этой работы имеет смысл. Например, можно расширить онтологическую модель путем добавления новых предметных областей. Также можно повысить уверенность построенных знаний с помощью внедрения новых аксиом для машин логического вывода.

5. Заключение

Автоматизация процесса проведения собеседования является важной и актуальной, на текущий момент, задачей. Создание платформы, проводящей тестирование знаний, навыков кандидатов, а также оценки их компетенций, позволит не только сократить среднее время собеседования, но и ускорить процесс закрытия вакансий. В совокупности, подобные факторы приведут к значительному сокращению временных затрат квалифицированных специалистов. Ключевой особенностью, подобной технологии является возможность использования для различных профессиональных сфер.

В ходе работы, благодаря использованию различных шаблонов проектирования, была заложена гибкая архитектура, позволяющая внедрение новых типов заданий, а также использование различных платформ для интеграции. Была построена четырехуровневая онтологическая модель знаний об языках программирования Java, Python, C# и объектно-ориентированном программировании. Разработаны ассоциативные правила, позволяющие делать выводы о знаниях пользователей. Таким образом, предлагаемое программное решение не зависит от конкретных областей знаний и полностью открыто для дальнейшего расширения.

Результаты данной работы могут служить отправной точкой для последующих исследований. В дальнейшем предполагается расширения построенного модуля, во-первых, за счет наполнения онтологической модели предметной области, а во-вторых, за счет добавления новых типов заданий и различных областей знаний. Параллельно с этим планируется проверка работы интеллектуального помощника на реальных пользователях, в закрытых группах размерами от пяти до десяти человек.

Литература

[1] R. T. Yarlagadda and H. H. Syed, ―Data models in information technology, International Journal of Innovations in Engineering Research and Technology, vol.3, no.2, Feb. 2016.

[2] Japheth R. B., Ogheneovo E. E. On a Conceptual Data Modeling Tool via Data Abstraction Using the Entity Relationship Model //Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 8 No 1. Pp 61-68

[3] A. Elfaki, A. Aljaedi and Y. Duan, "Mapping ERD to Knowledge Graph," 2019 IEEE World Congress on Services (SERVICES), Milan, Italy, 2019, pp. 110-114, doi: 10.1109/SERVICES.2019.00038.

[4] Al-Fedaghi S. Dissipating with Relations: Implication for the Entity-Relationship Model //arXiv preprint arXiv:2008.00135. – 2020.

[5] Ahmed Z., Gerhard D. Role of ontology in semantic web development //arXiv preprint arXiv:1008.1723. – 2010.

[6] ElAssy O. et al. Towards Ontological Conversation Interpretation: A Method for Ontology Creation from Medical Guidelines //arXiv preprint arXiv:2111.15338. – 2021.

[7] Neal C. et al. Ontology-Based Anomaly Detection for Air Traffic Control Systems //arXiv preprint arXiv:2207.00637. – 2022.

[8] Chandra R., Agarwal S., Singh N. Semantic sensor network ontology based decision support system for forest fire management //Ecological Informatics. – 2022. – Т. 72. – С. 101821.

[9] Найданов Ч. А., Пальчунов Д. Е., Сазонова П. А. Разработка автоматизированных методов предупреждения рисков возникновения критических состояний, основанных на анализе знаний, извлечённых из историй болезней пациентов // Сибирский научный медицинский журнал. 2016. №1.

[10] Долгушева Е. В., Пальчунов Д. Е. Теоретико-модельные методы порождения знаний о предпочтениях абонентов мобильных сетей //Вестник Новосибирского государственного университета. Серия: Информационные технологии. – 2016. – Т. 14. – №. 2. – С. 5-16.

[11] Смирнова Е. В., Добрица Е. К., Демиденко Н. О. Использование онтологий в образовательных процессах //Проблемы современной науки и образования. – 2017. – №. 22 (104). – С. 70-74.

[12] Белоус В. В., Карпенко А. П., Смирнова Е. В. Оценка понятийных знаний обучающегося на основе иерархической ролевой онтологии //Наука и образование: научное издание МГТУ им. НЭ Баумана. – 2014. – №. 9. – С. 181-193.

[13] Nongkhai L. N., Wang J., Mendori T. Developing an Ontology of Multiple Programming Languages from the Perspective of Computational Thinking Education //International Association for Development of the Information Society. – 2022.

Знания-Онтологии-Теории (ЗОНТ-2023)

Истинность знаний, эмоции, искусственный интеллект

Иванус Александр Иванович

ФГОБУ ВО «Финансовый университет при Правительстве Российской Федерации», Ленинградский проспект, д. 49/2, 125167, Москва, Россия

9162249075@mail.ru

Аннотация. *Мозг человека по своей природе не может мыслить по правилам математики, он мыслит по тем правилам, которые ему определила эволюция. Но тем не менее именно «нематематический» мозг создал математику. Для чего? Математический язык гарантирует максимальную интуитивную истинность выводимых умозаключений, которые в принципе не должны иметь альтернатив и противоречий за счёт соответствующего выбора системы аксиом, доказательства теорем и проч. В противоположность математическому наш естественный вербальный язык содержит альтернативы практически любому умозаключению. Онтология знаний на вербальном языке – это своего рода информационный «бульон», состоящий из альтернативных и противоречивых, но истинных данных о предметной области. Из двух альтернативных утверждений: «стакан наполовину пуст» и «наполовину полон» – оба истинные. Из этого вербального «бульона» мозг формирует устойчивые структуры в виде семантических ядер истинности знаний. Данные семантические ядра и являются смысловыми «единицами» мышления. Именно подобные структуры мозг использует в своём осуществлении «нематематического» процесса генерации новых знаний, реализуя принцип «истинность из неопределённости», как и «порядок их хаоса». Данный подход исследован и предложен в качестве рабочей гипотезы о возможности генерации новых знаний вне мозга человека.*

Ключевые слова: истинность информации, семантическое ядро, искусственный интеллект, новое знание, эмоции.

Парадоксы робототехника Ханса Моравека (1988):

- *два противоположных утверждения могут быть одновременно истинными. Например: «Все люди лгут» и «Некоторые люди говорят правду»;*
- *вопреки распространённому мнению когнитивные процессы высокого уровня требуют относительно небольших вычислений, их легче реализовать, в то время как низкоуровневые сенсомоторные операции требуют огромных вычислительных ресурсов.*

1 Введение

Несмотря на впечатляющие успехи программных продуктов, заносчиво именующих сами себя искусственным интеллектом (ИИ), тем не менее накопилось весьма значительное количество обстоятельств и фактов, которые приглашают к некоторому переосмыслению и корректировке выбранного нейросетевого тренда развития ИИ. В первую очередь речь идёт о том, что ИИ

успешно решает те задачи, которые уже решенычеловеком. Это – игра в шахматы, автопилотирование, распознавание образов и проч.

Реально же непосредственно сформулировать задачу ИИ пока не может. Потому что каждая такая задача относится к классу задач генерации нового знания (ГНЗ), что в свою очередь вызвано потребностями эволюционного процесса. А относительно же наличия у ИИ эволюции говорить пока не приходится.

Именно решение задачи ГНЗ выделяет человека из окружающего мира, именно ей мы обязаны не только всеми благами цивилизации, но и наличием самой цивилизацией. Задача ГНЗ не родилась на пустом месте, она появилась в результате эволюции, главным критерием которой является выживание. Поэту связка «выживание – новое знание» представляется имеющей чрезвычайно важную конструктивную основу.

Рассматривая данную связку, мы видим, что задача ГНЗ не есть математическая задача. Мозг человека не работает по законам математики. Он работает по каким-то другим законам, которые мы до конца ещё не осознали. А что вместо математики мозговые процессы используют? Какой такой внутри мозга имеется «нематематический» алгоритм, что он умеет генерировать НЗ, и при этом достаточно успешно (с точки зрения опять той же самой эволюции)?

Есть один аспект этой проблемы, существование которого всегда признаётся, но дальше констатации он не имеет продолжения. Этот аспект – истинность знания вообще и нового знания, в частности. Именно истинность считается его одним из главных атрибутов знания.

Достижение истинности знания всегда было прерогативой науки, собственно, это и есть основная задача, ради которой она существует. Здесь информационная сторона дела отходит на второй план по сравнению с истинностной. В обеспечении истинности знания непосредственно само её доказательство составляет наибольшую долю трудоёмкости.

Здесь мы отмечаем весьма важный момент. Истинность знания об некотором объекте окружающего мира не может быть определена просто, «в лоб». Она может быть получена в результате сравнения утверждения об изучаемом объекте с утверждением о каком-то другом объекте, заранее известном как истинным. Истинность, как и теплота, которая исходит от более нагретого тела к менее нагретому, переходит от более истинного утверждения к менее истинному. Этот вывод не вполне очевиден, но он подтверждается при более детальном рассмотрении этой проблемы.

В соответствии с концепцией К. Гёделя источник истинности знаний должен находиться в некоторой внешней «смысло-объясняющей» надстройке, находящейся за пределами исследуемой системы [1].

Внутри же самой системы истинность определить невозможно (это то же самое, что «вариться в собственном соку», «вытащить себя за волосы» и т. д.).

2 Теоретические проблемы

Что такое истинность? Этот вопрос всегда стоял на повестке всех учёных вовсе времена. По нашему мнению, интуитивно понятие истинности ассоциируется с врождённым человеческим чувством, определяемым как

понятие собирательное (или свёртка), объединяющее близкие по смыслу составляющие [2]:

Истинность = правдоподобность + непротиворечивость + убеждённость + очевидность + подлинность + объективность + надёжность + достоверность + проверенность + уверенность + убедительность + правильность + несомненность + точность + реальность + доверие + верифицируемость + подтверждаемость + ….

Приведём некоторые известные определения истинности, которые учитывают наличие того факта, что исследуемый объект содержит сложные многоаспектные альтернативные отношения и процессы типа: целостность-раздробленность, коммуникация-изоляция, согласие-антагонизм, противоречие-согласованность и т.д.

1. Семантическая истинность по А.Тарскому. Для обеспечения истинности знания придумана специальная наука, которая по своей сути является генератором истинности. Эта наука – математика. Именно математика по своей сути, отвлекаясь от реального мира, а оперируя только вымышленными объектами – точками, прямыми, плоскостями, функциями и проч., формулирует из некоторого набора исходных истинных утверждений (аксиом) более сложные утверждения (теоремы), истинность которых специальным образом доказывается. И математика, как хранилище истинных утверждений, может служить и служит эталоном истинности для других, уже связанных с реальными объектами, утверждений. Данный метод связан с именем А.Тарского, который установил, что умозаключение считается истинным тогда и только тогда, когда оно формально внутренне непротиворечиво [3]. А математика как раз и обеспечивает эту непротиворечивость. Ещё этот подход называется семантической истинностью, так как А.Тарский показал, что доказательство непротиворечивости высказываний основано на их семантическом анализе.
2. Когерентная истинность. НЗ должно быть согласовано с другими знаниями, относительно которых известно, что они все являются истинными [4]. В качестве таких проверенных эталонов истинности служит множество общесистемных законов принципов (сохранение энергии, вещества, наименьшего действия, максимума энтропии и проч.), истинность которых подтверждена многолетней практикой, экспериментами, исследованиями. Противоречие этим законам и принципам при оценке истинности недопустимо.
3. Конвенциональная истинность. Значительный вклад в обеспечение истинности знания может внести также мнение коллег и экспертов. Они, как носители истинных знаний, должны подтвердить или не подтвердить истинность производимого НЗ [5]. Их мнение есть по сути дела база данных истинных знаний, несмотря на возможный разброс в деталях и тонкостях. По этому принципу единства носителей истинности формируются диссертационные советы и другие экспертные сообщества. Решения относительно истинности принимаются, как правило, голосованием. Этот способ, конечно, во многом субъективен, однако,

часто применяется на практике в силу своей простоты и чисто «человеческой» понятности.

4. Но самым известным, корректным и проверенным в течение последних 2400 лет считается метод, связанный с именем Аристотеля [6]. По Аристотелю «истина есть адекватное отображение действительности в сознании» или «практика – критерий истины».

Существует еще ряд методов обеспечения истинности, которые применяются на практике. Однако, рассмотренные методы имеют то преимущество, что дают возможность подойти к истинности не только качественно, но и количественно.

Для формирования количественной меры истинности предложен подход, основанный на известном принципе максимума энтропии [2].

Использование вариационных принципов термодинамики и, в частности, принципа максимума энтропии Л. Больцмана давно зарекомендовали себя в исследованиях таких сложных нефизических систем, как экономические, относящиеся ближе к гуманитарным в силу превалирующего влияния в них поведения человека. Решающим фактором здесь является то, что каждый человек волей или неволей вносит, причём каждый на своём уровне социальной лестницы, в систему генерации НЗ свою лепту, что делает эту системы близкой по совокупному поведению всех этих микровоздействий с термодинамической системой. Тем более что в соответствии с исследованиями А.П. Левича [7] *«…конструкция энтропии получена вне каких-либо статистических или вероятностных предпосылок. Величина обобщённой энтропии может быть строго рассчитана для состояний любых систем, эксплицируемых математическими структурами. Она может быть вычислена для состояний, описываемых множествами с любым – большим или малым – количеством элементов. Вероятностные интерпретации могут возникать при рассмотрении определённых типов систем, но они совершенно не обязательны для расчётов энтропии. Для некоторых математических структур, например, для множеств с разбиениями, формула для обобщённой энтропии полностью совпадает с формулами для энтропии идеального газа Л. Больцмана или для энтропии каналов связи К. Шеннона».*

Опыт использования принципа Л. Больцмана говорит о том, что такой подход оправдан и позволяет получать для практики полезные результаты. Необходимо только выполнять присущие этому принципу ограничения и предписания.

Введём следующие понятия:

- минимальный истинный элемент знания (МИЭЗ). Это то минимальное по форме истинное утверждение, дальнейшее уменьшение (сокращение) которого приводит к лишению его смыслового содержания.
- семантическое ядро истинности нового знания (СЯИНЗ) – это утверждение, содержащее в своей основе некоторое количество *x* МИЭЗ. Поэтому истинность СЯИНЗ определяется всей совокупностью входящих в него МИЭЗ.

Связь между СЯИЗ и МИЭЗ записывается в виде равенства:

$$СЯИЗ = SUMM (МИЭЗi, i =1, x). \quad (1)$$

Следуя принципу максимума энтропии [7] задачу формирования структуры истинного СЯИНЗ можно сформулировать так: из заданного

состояния замкнутая на множестве единиц истинности СЯИНЗ стремится перейти в другое состояние, для которого суммарная энтропия истинности на множестве всех единиц истинности стремится к максимуму, и в самой общей постановке записывается в виде равенства [8]:

$$H = -\int_{\Omega} f(x) \ln f(x) dx = max, \quad (2)$$

при наличии ограничений:

$$\int_{\Omega} E(x) f(x) dx = E,$$

$$\int_{\Omega} f(x) dx = 1,$$

где

x – количество единиц истинности (равное количеству истинных МЭИЗ), заданных на множестве **Ω**, и входящих в состав СЯИЗ;

f(x) – искомое частотное распределение единиц истинности в количестве *x*, входящих в СЯИЗ;

H – энтропия истинности СЯИЗ;

E(x) – ресурс системы, затрачиваемый на генерацию единиц истинности в количестве x;

E – общий суммарный ресурс системы,

Опуская промежуточные расчёты, приведенные в [2], представим окончательное решение системы (2) в виде:

$$f(x) = \frac{1}{\varphi E} \frac{1}{x^{\frac{1}{\varphi E}+1}} = \frac{\alpha}{x^{1+\alpha}}, \quad (3)$$

где величина $\alpha = \frac{1}{\varphi E}$,

φ – параметр интенсивности производства МЭИЗ.

Отсюда, заменяя $x_{max} = Q$, получаем:

$$H_{max}(Q) \approx 4{,}59\ Q^{-3{,}59}\ . \quad (4)$$

Мы получили чрезвычайно удобную формулу зависимости максимального значения энтропии *H* от единственной переменной, а именно – от максимального количества единиц истинных МЭИЗ – *Q*.

Данное решение имеет принципиальную особенность, отличающую его от классического понимания энтропии, а именно, в классической формуле энтропия системы увеличивается с увеличением количества входящих в неё элементов, а здесь наоборот – с увеличением количества истинных элементов максимум энтропия истинности монотонно уменьшается.

Отсюда следует ещё один интересный вывод: истинность целого меньше суммы истинностей частей.

Таким образом, если оценивать истинность НЗ, то это НЗ нужно «пропустить» через своего рода все четыре упомянутых фильтра, как это показано на рис. 1.

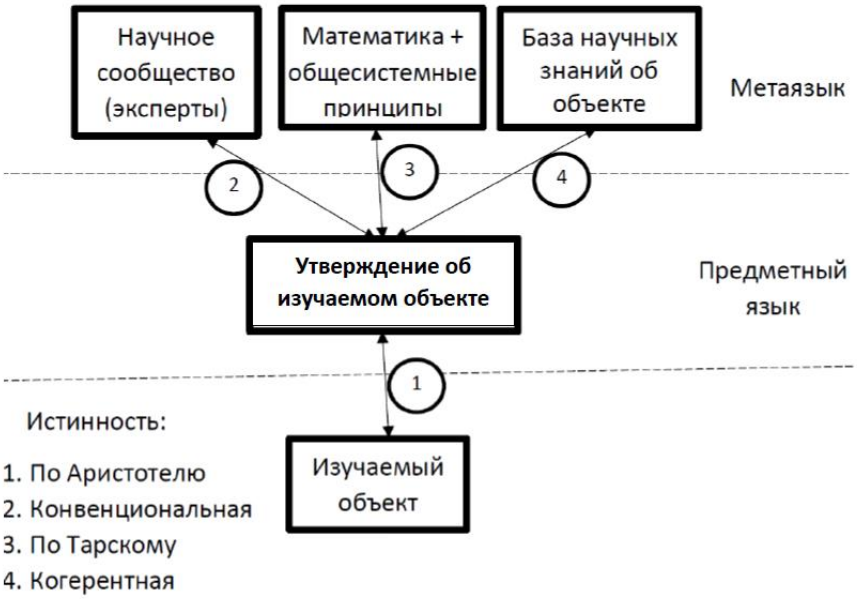


Рис. 1. Схема обеспечения истинности НЗ об изучаемом объекте.

Поскольку истинность, как мы определили, есть врождённое интуитивное чувство, то именно для чувственной стороны истинности должно быть место в нашей схеме. Известно, что эмоциональный подъём знаком всем, кто в процессе достижения какого-либо умственного результата (например, решение трудной задачи). То есть эмоция следует за моментом достижения истинного знания или совсем кратко – эмоция есть индикатор момента появления истинности. Но проблема в том, что очень часто эмоция может сопровождать ошибочный, неверный результат. Наличие эмоции как бы «сбивает с толку» исследователя. Но тем не менее чувственная сторона сопровождает всякое НЗ, даже если оно является результатом добросовестного заблуждения. Поэтому среди множества действительно истинных знаний существует большая доля знаний малоистинных. Причём количество малоистинных знаний гораздо больше, чем истинных. И эти малоистинные знания играют очень важную роль в познании, так как они могут служить той питательной средой, которая стимулирует поиск истинных знаний.

Данный подход предполагает, что в поиске истинных НЗ нельзя полностью отказаться от эмоциональной составляющей. Но именно так пытаются представить задачу многие исследователи, занимающиеся искусственным интеллектом. Сознательно пренебрегая эмоциональной составляющей, они пытаются построить чисто математическую модель работы мозга, в которой заведомо нет этих малоистинных знаний. Но если там нет малоистинных составляющих (они или не учитываются или отбрасываются), то в какой среде тогда искать истинность новых знаний?

Отсюда следует: задача создания искусственного интеллекта должна решаться с использованием в качестве аргументной базы не только истинные математические знания, но и обязательно все малоистинные.

Искусственный интеллект на основе только математической модели без чувственной модели вряд ли возможен. Потому что тогда в рамках только лишь математической модели нет смысла искать истинность. В математике истинность

уже доказана. Истинность надо искать там, где её или совсем нет или её мало, т. е. в там, где не обеспечена непротиворечивость по К. Гёделю одновременно двух истинных высказываний: истинно **A** и истинно $\bar{\mathbf{A}}$,

Здесь имеет место своего рода логическая (вербальная) бифуркация, как точка раздвоения общественного сознания: одна часть общества считает истинным А, а другая – $\bar{A}$. Это продолжается до тех пор, пока не появится некоторая метатеория (метаконцепция, метагипотеза), которая обеспечит условие непротиворечивости высказываний, и тем самым определит, которое из них, А или $\bar{A}$, действительно истинно. Если в общественной жизни такие метатеории появляются не часто, то и раздвоение общественного сознания может устойчиво сохраняться в течение долгого времени.

Представленный подход даёт нам инструмент, который позволяет целенаправленно строить альтернативы уже имеющимся умозаключениям. Этот процесс, по сути, есть не что иное, как генерация НЗ.

Если на уровне эмоций и чувств эта тема ближе всего расположена к искусству, поэзии, политике и т. д., то уже для вербальных социально-экономических моделей и других этот механизм генерации НЗ может быть вполне успешно реализован [9].

Данный подход имеет особенно важное значение для такой области, как прогностика. Ведь будущее всегда плохо предсказуемо, в нём с увеличением горизонта прогноза относительное количество цифровой информации уменьшается, а увеличивается относительное количество вербальной информации. Вот здесь как раз методы генерации вербальных альтернатив могут с успехом обеспечить перебор возможных вариантов будущего.

Поскольку математические знания не связаны с реальностью, то получается, что и истинности в понимании Аристотеля в математике нет.

Истинности по Аристотелю нет, а истинности по А. Тарскому для математических знаний есть. Тогда возникает вопрос: как совместить эти две вещи? Опять приходит на помощь К. Гедель. Если в системе сосуществуют две альтернативные сущности, значит должна быть мета-система, которая «выберет» одну из них, а другую отвергнет.

Когда в мозге человека появляется то, что мы называем истинностью знания, то наша модель должна содержать описание как этой истинности, так и этого знания. Данное утверждение является принципиальным. Отсюда начинается первый шаг в направлении процесса, описываемого метафорической формулой «истинность из неопределённости», которая, как сказано выше, аналогична и идейно близка известной формуле «порядок из хаоса».

3 Заключение

Истинность имеет эмоционально-чувственную основу и, таким образом, круг исследований в области ИИ стремится к расширению и исследования принимают междисциплинарный характер.

Если методами ИИ решается задача генерации НЗ, то там необходимо тогда «вводить» модель эмоций в качестве некоторой компоненты генератора малоистинных знаний.

Литература

[1] Гёдель К.Ф. Совместимость аксиомы выбора и обобщённой континуум-гипотезы с аксиомами теории множеств. Успехи математических наук, 1948, 3:1 (23), стр. 96–149.

[2] Иванус А.И. Искусственная генерация новых знаний: Моделирование процессов мышления для получения новых знаний вне мозга человека. Качественный скачок в развитии искусственного интеллекта (Учебное пособие). – М: ЛЕНАНД. 2022. –200с. (Наука об искусственном. №45). https://urss.ru/cgi-bin/db.pl?lang=Ru&blang=ru&page=Book&id=288209

[3] Тарский А. Понятие истины в языках дедуктивных наук // Философия и логика Львовско–Варшавской школы. М.: РОСПЭН, 1999. с.14–178.

[4] Ляшов В.В. Когерентная теория истинности и истинностно-значная семантика // Логико-философские исследования. М., 1991. Вып. 2.

[5] Кун Т.С. Структура научных революций. Сост. В.Ю. Кузнецов. М.: ООО «Издательство АСТ», 2003. —605 с.

[6] Аристотель. Сочинения. В 4 т. (Серия «Философское наследие»). М.: Мысль, 1975—1983

[7] Левич А.П. Энтропия как обобщение понятия количества элементов для конечных множеств // Философские исследования. 2001. №1. С. 59–72.

[8] Заличев Н.Н. Энтропия информации и сущность жизни. М.: Радиоэлектроника, 1995. 192 с.

[9] Иванус А.И. Универсальный алгоритм генерации новых знаний и управленческих решений на основе моделирования когнитивных функций мозга человека. Рыночная стоимость оценена независимым оценщиком в размере 12,5 млн. руб. (1. Свидетельство о регистрации объекта интеллектуальной собственности № 20-871: Реестр Фонда интеллектуальной собственности. Москва. 16.03.2020. 2. Сертификат стоимости объекта интеллектуальной собственности №20-19. Москва. 15.07.2020.).

Знания-Онтологии-Теории (ЗОНТ-2023)

Концепция и архитектура базы знаний систем линейных алгебраических уравнений

В. П. Ильин

Институт вычислительной математики и математической геофизики СО РАН

Рассматриваются мотивация разработки, концепция, архитектура и технологические аспекты базы знаний систем линейных алгебраических уравнений (СЛАУ), включающей библиотеки алгоритмов, коллекции и генераторы матриц, классифицированную библиографию и глоссарий, документацию и примеры заданий, архивы расчетных данных, средства принятия решений и машинного обучения, а также инструменты поддержки активного жизненного цикла системы.
Ключевые слова: классификация матриц, прямые и итерационные методы, распараллеливание алгоритмов, архивы данных, машинное обучение.

1 Введение

Решение систем линейных алгебраических уравнений (СЛАУ) представляет актуальную проблему вычислительной математики и суперкомпьютерного моделирования по двум основным причинам. Первое - всеобщая распространенность данной расчетной стадии, поскольку при исследовании фундаментальных и прикладных проблем, описываемых системами дифференциальных и/или интегральных уравнений, а также сетевыми и оптимизационными постановками, включая нестационарные и нелинейные задачи, практически неизбежно приходится решать СЛАУ высоких порядков. Второй аспект – это большая ресурсоемкость данного вычислительного этапа, поскольку объемы арифметических операций и требуемой памяти растут нелинейно с увеличением размерности задачи, когда число неизвестных доходит до 10^9 - 10^{12}. При этом доля машинного времени может достигать 80 - 90%, а экономические вопросы крупномасштабных машинных экспериментов становятся весьма насущными, см. [1], [2].

Несмотря на свою внешнюю простоту, СЛАУ представляет собой очень содержательный математический объект, определяемый главным образом свойствами своей матрицы, которая может быть вещественной и комплексной, симметричной и несимметричной, вырожденной и невырожденной, плотной и разреженной, и т.д. Еще более богатым является класс методов решения СЛАУ, которые делятся на прямые и итерационные. Среди последних уже более полувека основными являются предобусловленные алгоритмы в подпространствах Крылова. Программные реализации для широких классов алгебраических систем и их решателей представлены в многочисленных общедоступных или коммерческих библиотеках, количество и качество которых растет одновременно с бурным развитием компьютерных платформ.

Особую востребованность здесь приобретают распараллеливание алгоритмов и их отображение на архитектуры многопроцессорных вычислительных систем (МВС) с распределенной и иерархической общей памятью. Математическог и программное обеспечение по решению СЛАУ является одной из продвинутых областей вычислительных наук, а теоретические и экспериментальные исследования в этих направлениях активно продолжаются учеными многих стран, о чем свидетельствует стабильных рост научных публикаций и конференций [3]-[5].

Глобализация математического моделировния постоянно приводит к появлению новых типов СЛАУ, зачастую плохо обусловленных, неустойчивых и экстремально ресурсоемких, требующих специальных подходов к их решению с гарантированной точностью. Несмотря на значительный прогресс в данной области, решение алгебраических систем отнюдь не является рутинной областью, а требует глубоких знаний и практического опыта, что неизбежно приводит к довольно узкой специализации математика-программиста, фактически недоступной для широкого круга пользователей с различной профессиональной подготовкой.

В этих условиях, когда исходные данные в реальных условиях могут оказываться неточными, неполными и противоречивыми, машинный эксперимент становится на грани между наукой и искусством, что требует в конечном итоге интеллектуализации вычислительной деятельности, направленной на качественное повышение производительности труда как разработчиков, так и конечных пользователелей математического инструментария. Это принуждает к более широкому взгляду на создание программного продукта нового поколения как экосистемы или интегрированного вычислительного окружения (ИВО), содержащего не только реализации алгоритмов, но и весь необходимый багаж знаний. В применении к рассматриваемой нами предметной области это включает, например, коллекции и генераторы матриц для характерных методических и практических задач, примеры формирования исходных данных, архивы расчетных результатов, анализатор свойств матриц и СЛАУ, классифицированная библиография и глоссарий, гибкие средства расширения задач и применяемых алгоритмов, а также адаптации к эволюции компьютерных платформ, использование внешних программных продуктов, инструменты машинного обучения, в том числе выбор оптимального метода в конкретной ситуации, и т.д., см. [6]-[23] и цитируемые там работы,

В п.2 данной статьи мы на содержательном уровне представляем предметную область СЛАУ. Пункт 3 содержит описание общей структуры и функциональности базы знаний (БАЗ), а в заключении обсуждаются дальнейшие планы исследований.

2 Описание предметной области СЛАУ

Рассматриваемая проблема может быть в общем случае записана в комплексной форме:

$$Au = f, \quad A = \{a_{i,j}\} \in \mathbb{C}^{N_r,N_c}, \quad u = \{u_j\} \in \mathbb{C}^{N_c}, \quad f = \{f_i\} \in \mathbb{C}^{N_r}, \qquad (1)$$

где N_r и N_c - число строк и столбцов матрицы A соответственно. Наиболее частые на практике СЛАУ, которые мы и будем рассматривать в дальнейшем, имеют квадратные вещественные матрицы $A \in \mathbb{R}^{N,N}$. Классификацию исследуемой предметной области мы рассмотрим кратко по следующим характеристикам, имея целью только обозначить многообразие объектов и операций над ними: свойства алгебраических систем, методы их решения, распараллеливание алгоритмов, программные технологии реализации.

2.1 Классификация и свойства СЛАУ

Актуальные на практике алгебраические системы мохно характеризовать по таким признакам, как их происхождение, структурные и спектральные свойства, а также способы представления. Наибольший интерес представляют "большие"СЛАУ с порядками $N \approx 10^9$ - 10^{12} и более, при этом средними на текущий период развития ЭВМ можно считать $N \approx 10^5$ - 10^8.

Практически все матрицы делятся на два типа – разреженные и плотные. В первом случае число ненулевых элементов мало, т.е. $nz \ll N^2$, и, как правило, их количество n_r в одной строке не зависит от N. Такие системы возникают при сеточных дискретизациях многомерных краевых задач, когда величина n_r составляет единицы или несколько десятков, причем она растет с увеличением порядка аппроксимации. Плотные матрицы возникают при численном решении интегральных уравнений, и в этих случаях величина nz равна или примерно пропорциональна N^2.

Наиболее богатый и практически важный класс составляют большие разреженные СЛАУ, имеющие разнообразные блочные структуры: ленточные, окаймленные, седловые и др., которые зависят, естественно, от упорядоченности векторных компонент. Для однородных (единообразных) сеточных СЛАУ важное значение имеет конфигурация шаблона, то есть совокупности ненулевых элементов в строке матрицы. При этом можно определить граф матрицы, в котором ее ненулевой элемент $a_{i,j}$ соответствует ребру между i-й и j-й вершинами.

Важнейшими характеристиками матриц являются также их спектральные свойства, определяемые собственными и сингулярными числами, а также соответствующими векторами, в зависимости от которых СЛАУ может иметь или не иметь единственного решения. Актуальной количественной спецификацией невырожденной матрицы служит ее число обусловленности в одной из гельдеровских норм

$$cond_r(A) = ||A||_p||A^{-1}||_p, \quad p = 1, 2, \infty, \qquad (2)$$

определяющее свойство устойчивости численного решения СЛАУ к погрешностям исходных данных задачи или к арифметическим ошибкам округления. С последним вопросом тесно связан способ представления векторно-матричных компонент, где общепринятыми стандартами являются простая

или двойная точность с 32 и 64 битами в машинном слове. Для экстремально плохо обусловленных СЛАУ ($cond(A) \approx 10^{15}$) может потребоваться повышенная точность, что значительно увеличивает вычислительную сложность, и ее необходимость должна быть обоснована дополнительным априорным или апостериорным анализом.

С технологической точки зрения существенным является формат хранения матрицы. Наиболее сложный случай – когда СЛАУ генерируется из аппроксимации уравнений на неструктурироаннной сетке, при этом для каждого узла номера его соседей можно указать только перечислением. В таких случаях вынужденно применяются разреженно-сжатые форматы (например, CSR - Compressed Sparse Row), позволяющие хранить только ненулеые матричные элементы, но существенно замедляющие доступ к их значениям.

2.2 Краткий системный анализ методов решения СЛАУ

Существует два различных подхода к решению алгебраических систе - прямые и итерационные алгоритмы. Первые основаны на методах исключения Гаусса (LU-разложения), требующих выполнения $O(N^\gamma), \gamma \leq 3$ арифметических действий и значительных объемов памяти, делающих нецелесообразным их применение для больших матриц. Существующие стратегии оптимизации упорядоченности векторных компонент, выбора ведущих матричных элементов и уточнения численных решений, делают эти методы надежным (робастным) средством решения средних по размеру СЛАУ (имеющиеся исключения только подтверждают это правило).

Для больших разреженных систем безальтернативным является применение итерационных процессов вида

$$B(u^{n+1} - u^n) = r^n \equiv f - Au^n, \quad n = 0, 1, ..., \tag{3}$$

где число итераций определяется по условию

$$||r^n|| \leq \varepsilon ||f||, \quad \varepsilon \ll 1 \tag{4}$$

и зависит от количества обусловленности матричного произведения $B^{-1}A$, где B – предобуславливающая матрица, которая должна экономично обращаться и хорошо аппроксимировать исходную матрицу A.

Современные итерационные подходы основаны на итерационных методах в подпространствах Крылова, реализуемых с помощью рекуррентных формул

$$\begin{gathered} u^{n+1} = u^n + \alpha_n p^n, \quad r^{n+1} = r^n - \alpha_n A p^n, \\ p^{n+1} = \beta_{1,n}^{-1} r^{n+1} + \sum_{k=0}^{m} \beta_{n,k} p^{n-k}, n = 0, 1, \ldots, \end{gathered} \tag{5}$$

где $\alpha_n, \beta_{n,k}$ и p^n – итерационные параметры и направляющие векторы, вычисляемые из различных ортогональных, вариационных и проекционных условий. Здесь существенно различаются алгоритмы с короткими ($m = 0$) и

длинными ($m = n$) рекурсиями, возникающими при решении симметричных (эрмитовых в комплексном случае) и несимметричных СЛАУ и определяющими соответственно методы сопряженных или полусопряженных направлений (примером последних служит популярный GMRES – обобщенный метод минимальных невязок). Наличие длинных рекурсий при большом числе итераций сильно увеличивает ресурсоемкость алгоритма, и тогда искусственно ограничивают значение m в (5). Это, в свою очередь, замедляет скорость сходимости итераций, что приводит к необходимости дополнительных исследований для поиска "золотой середины".

Имеется универсальный подход с трансформацией Гаусса, связанный с умножением матрицы A на транспонированную к ней $A^\top$. Это приводит к симметризации СЛАУ, но ухудшает ее обусловленность и требует дополнительного матричного умножения на каждой итерации. Существуют также методы решения несимметричных систем, основанные на биортогонализации векторов (примеры - алгоритмы типа BiCGStab или IDR) или бидиагонализации матриц Голуба-Кахана.

Другие интересные развития крыловских процессов связаны с дополнением подпространств на основе использования апостериорной информации. Такие разработки активно проводятся при многократном решении СЛАУ с различными последовательно определяемыми правыми частями, что является актуальными в нелинейных, нестационарных и оптимизационных задачах математического моделирования.

Для предобуславливания СЛАУ имеются три основные стратегии: приближенные треугольные разложения матриц (ILU - Incomplete LU-factorization), многосеточные алгоритмы (MG - Multi-Grid), асимптоматически оптимальные по порядку, т.е. число операций пропорционально количеству неизвестных, и методы декомпозиции областей (DDM – Domain Decomposition Methods), которые являются главным инструментом распараллеливания алгоритмов на МВС с распределенной и общей иерархической памятью. Все итерационные алгоритмы наиболее эффективно реализуются с учётом их «геометрической предыстории», использующей информационные связи между соседними узлами сетки и соответствующие структурные свойства матриц. Однако зачастую задача конструирования решателя формулируется наоборот, в «алгебраическом ключе», когда алгоритм требуется реализовать только с использованием, например, CSR-формата матрицы. На его основе изучить блочную структуру матрицы в принципе можно, но решение этой проблемы требует серьёзного вовлечения искусственного интеллекта, поскольку в идеале вычислительный модуль должен функционировать как «чёрный ящик», без вмешательства человека.

2.3 Стратегии и тактики распараллеливания алгоритмов

Основными количественными показателями качества распараллеливания являются ускорение вычислений и эффективность использования процессоров

$$S_p = T_1/T_p, E_p = S_p/P, \quad T_p = T_p^a + T_p^c \tag{6}$$

где T_p^a и T_p^c суть времена выполнения арифметических операций и коммуникаций при решении задачи на p процессорах. Эти характеристики алгоритмов для конкретных конфигураций МВС можно оценить только очень грубо, и главный инструмент исследований здесь – это экспериментальные исследования со сравнительным анализом быстродействия различных алгоритмов для характерных типов СЛАУ. Фактически оптимизация вычислительных моделей заключается в минимизации обменов, так как передачи данных не только замедляют работу компьютера, но и значительно увеличивают энергозатраты, что существенно сказывается на эксплуатационных расходах.

Одним из основных средств распараллеливания на современных кластерных архитектурах МВС является гибридное программирование путём передачи сообщений между соединёнными шинами, одновременно работающими многоядерными вычислительными узлами с помощью библиотеки MPI (Message Passing Interface), а также проведением многопотоковых расчётов на каждом узле с помощью системы Open MP на общей памяти. Кроме того, можно осуществлять векторизацию операций, используя систему команд AVX, учитывающей свойства иерархической структуры памяти CPU (Central Processor Unit). Один из общих принципов ускорения заключается в совмещении арифметических действий с информационными обменами, что требует тонкого управления вычислительным процессом. Здесь могут быть использованы эффективные вычислительные инструментарии, такие как библиотека матрично-векторных операций Sparse Blas со встроенными параллельными режимами исполнения. Существует поговорка, что «наиболее быстрые методы плохо распараллеливаются», которая может быть права в том, что если мы на каком-то вычислительном этапе сократим число арифметических операций, то это приведёт к увеличению относительной доли времени коммуникаций. Заключительный вывод данных качественных рассуждений состоит в том, что оптимизация распараллеливания – это поиск алгоритмов со сбалансированными обменами вычислений и обменов данными.

2.4 Общая структура и функциональность базы знаний СЛАУ

В целом БАЗ для алгебраических систем можно определить как интеллектуальное вычислительно-информационное окружение, призванное обеспечить высокую производительность программного кода, а также его разработчиков и пользователей. Главную целевую функциональность такой инструментальной экосистемы можно определить следующим образом: пусть библиотека содержит представительный набор решателей для широкого класса алгебраических систем, а также архив расчётных данных с анализом и рекомендациями по предыдущим экспериментам и набор методических руководств Требуется для решения заданной СЛАУ или их набора подобрать конкретный код, обеспечивающий максимальную производительность на применяемой МВС.

Чисто теоретически поставить, а тем более решить такую задачу чрезвычайно сложно. Однако с привлечением большого объема экспериментальных данных в данном случае можно поставить проблему так называемой суррогатной оптимизации с привлечением эмпирических и статистических подходов. Другими словами, здесь применима идеология машинного обучения, в данном случае программно-информационной системы, представляющей собой базу математических знаний. Такая предметная область, как решение СЛАУ, является удобным объектом для построения концепции и методологии БАЗ, поскольку при всей кажущейся простоте алгебраической системы она обладает и актуальностью, и глубоким теоретическим содержанием, и включает технологические проблемы, затрагивающие профессиональные интересы широкого круга пользователей.

Современный уровень научно-технического прогресса порождает глубокие специализации в огромном количестве областей. Инженеру-авиаконструктору, например, невозможно, да и нецелесообразно разбираться в тонкостях решения СЛАУ. То же самое можно сказать и о программисте, который пишет код для моделирования управления самолетом, когда могут возникнуть нештатные алгебраические ситуации. И тут на помощь человеку должен прийти искусственный разум (в форме БАЗ), в котором заложены соответствующие знания и средства помощи в принятии решения.

Под базой знаний СЛАУ можно понимать хранилище всех возможных знаний и инструментов, которые могут оказаться полезными для решения рассматриваемого круга задач. С этой точки зрения в БАЗ включаются следующие разделы: библиотека алгебраических решателей, коллекции и генераторы матриц, архивы расчетных данных, анализатор СЛАУ, классифицированная библиография и глоссарий, пользовательская документация с примерами вычислительных сеансов, маркетинговая информация, различные инструментарии системной поддержки.

Для обеспечения длительного жизненного цикла с успешной широкой востребованностью такого большого проекта он должен удовлетворять следующим техническим требованиям: гибкое расширение состава алгоритмов и типов решаемых задач, адаптация к эволюции компьютерных платформ, переиспользование внешних программных продуктов, согласованное участие различных групп разработчиков, высокая производительность кода, интеллектуальные внешние и внутренние интерфейсы.

Состав решаемых СЛАУ может классифицироваться не только по первичным матричным характеристикам, но и по крупноблочным структурам, связанным с такими актуальными приложениями, как задачи электромагнетизма, прочности, гидрогазодинамики, многофазной фильтрации и т. д.

Набор алгебраических решателей включает как оригинальные реализации, так и возможности использования программ из других библиотек (например, EIGEN, PETSc, HYPRE и др.). Библиография по методам решения СЛАУ насчитывает тысячи наименований с многочисленными теоретическими разделами, в том числе следующие: CHE – чебышевские алгоритмы, KRY – методы в подпространствах Крылова, AMG – алгебраические мно-

госеточные методы, DDM – методы декомпозиции областей, REC – многократное решение СЛАУ, LLU – алгоритмы неполной факторизации, ADI – неявные методы переменных направлений, PRO – проекционные подходы, TRE – решатели на графах, PAR – параллельные алгоритмы, AI – искусственный интеллект, SAD – седловые СЛАУ и т. д. Данная компонента БАЗ совместно с глоссарием является основой онтологии СЛАУ, на которой должны формироваться инструменты машинного обучения разрабатываемой интеллектуальной системы (рука не поднимается написать модное слово "нейросети"), а также средства принятия решений.

3 Заключение. Обсуждение результатов и планов дальнейших исследований

В работе предложены концепция, целевой функционал и общая структура базы знаний СЛАУ. Проведен содержательный анализ рассматриваемой предметной области, состав функционального и системного наполнения БАЗ, технические требования к ее разработке, а также принципы реализации технологической поддержки машинного обучения и системы принятия решений. Планы дальнейших исследований включают построение онтологии и когнитивных инструментариев для создания интеллектуальной базы знаний СЛАУ.

3.1 Список литература

1. Dongarra J., Grigori, L., Higham, N. J. Numerical algorithms for high performance computational science, 2020. V. 378, iss. 2166 https://doi.org/10.1098/rsta.2019.0066
2. Il'in V.P. Iterative Preconditioned Methods in Krylov Spaces:Trends of the XXI Century. Computational Mathematics and Mathematical Physics, 2021, Vol. 61, No. 11, pp. 1750–1775. © Pleiades Publishing, Ltd., 2021. DOI: 10.1134/S0965542521110099
3. Y. Saad Y. Iterative Methods for Sparse Linear Systems, 2nd ed. (SIAM, 2003)
4. Dongarra J. List of freely available software for linear algebra on the web (2006) http://netlib.org/utk/people/JackDongarra/la-sw.html
5. Il'in V. Parallel intelligent computing in algebraic problems, Sokolinsky, L., Zymbler, M. (eds.) Parallel Computational Technologies. PCT 2021. Communications in Computer and Information Science. 2021. V. 1437. Springer, Cham. doi.org/10.1007/978-3-030-81691-9_8
6. Ильин В.П. Математическое моделирование Часть 1.Непрерывные и дискретные модели, Новосибирск, изд.СО РАН, 2017
7. Forrester A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modeling: A Practical Guide. Wiley, New York (2008)
8. Kleppe A. Software language engineering: Creating domain-specific language using metamodels. NY: Addison-Wesley, 2008
9. Il'in V. The integrated computational environment for optimization of complex systems, Proceedings of the 15-th International Asian School-Seminar "Optimization

Problems of Complex Systems (OPCS 2019). P. 65–67 doi.org/10.1109/opcs.2019.888015.
10. Il'in V. Artificial intelligence problems in mathematical modeling, Voevodin V., Sobolev S. (eds.) RuSCDays 2019. CCIS. 2019. V. 1129. P. 505–516. Springer, Cham. doi.org/10.1007/978-3-030-36592-9_41
11. Mikony S.V.: Formalization of the cognitive process using the basis of models. Ontol. Des. 8(1(27)), P. 35–48 (2018) (in Russian)
12. LeCun Y., Bengio Y., Hinton G. Deep learning, Nature, 2015. V. 521. P. 436–444 doi.org/10.1038/nature14539
13. Weinan, E.: Machine learning and computational mathematics // Commun. Comput. Phys. 2020. V. 28. P. 1639–1670 https://doi.org/10.4208/cicp.oa-2020-0185.
14. Borgest N.M. Key terms of ontology of designing: review, analysis, generalizations [In Russian]. Ontology of designing. 2013. V. 3(9). P. 9–31
15. Luccioni A., Bengio Y. On the Morality of Artificial Intelligence. arXiv:1912.11945 [cs.CY] /doi.org/10.48550/arXiv.1912.11945
16. Antonov A., Dongarra J., Voevodin V. AlgoWiki Project as an Extension of the Top500 Methodology, Supercomput. Frontiers and Innova. 2018. V. 5, N. 1. P. 4–10
17. Feoktistov A., Kostromin, R., Sidorov, I. A., Gorsky, S. A. Development of distributed subject-oriented applications for cloud computing through the integration of conceptual and modular programming, Proc. of the 41st Intern. Conf. on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2018). P. 234–239
18. Загорулько Ю.А.; Загорулько Г. Б. Онтологический подход к разработке системы поддержки принятия решений на нефтегазодобывающем предприятии, Вестн. НГУ. Сер.: Информационные технологии. 2012. Т. 10, вып. 1. С. 121–128
19. Zagorulko Y., Zagorulko G. Architecture of extensible tools for development of intelligent decision support systems, New Trends in Software Methodologies, Tools and Techniques. Proc. of the 10-th SoMeT_11. Hamido Fujita (Eds.). Amsterdam: IOS Press. 2011. P. 457–466
20. Il'in V.P., Skopin I. N. About performance and intellectuality of supercomputer modeling, Programming and Computer Software. 2016. V. 42, iss. 1. P. 5–16
21. Glinskiy B., Kulikov I., Sapetina, A., Zagorulko, Y., Zagorulko, G. The creation of intelligent support methods for solving mathematical physics problems on supercomputers, Communications in Computer and Information Science. 2019. V. 1129. P. 427–438
22. Gorodnichev M., Lebedev D. Semantic tools for development of high-level interactive applications for supercomputers, J. Supercomp. 2021. V. 77, iss. 10. P. 11866–11880 doi: 10.1007/s11227-021-03731-6
23. Бурнаев Е.В., Бернштейн А.В., Оселедец И.В. и др. Фундаментальные исследования и разработки в области прикладного искусственного интеллекта, Докл. РАН. Математика, информатика, процессы управления. 2022. Т. 508. С. 19–27

Алгоритм кластеризации временных рядов и его применение для анализа гликемических кривых

Кладов Д.Е.[1,2], Бериков В.Б.[1,3], Климонтов В.В.[1]

[1]Научно-исследовательский институт клинической и экспериментальной лимфологии – филиал ФГБНУ Федеральный исследовательский центр Институт цитологии и генетики СО РАН, ул. Тимакова, д.2, г.Новосибирск, Россия.
[2]Новосибирский государственный университет, ул. Пирогова, д.2, г.Новосибирск, Россия.
[3]Институт Математики СО РАН, пр. Коптюга, д. 4, г. Новосибирск, Россия.

d.kladov@g.nsu.ru, berikov@math.nsc.ru, klimontov@mail.ru

Аннотация. *В работе решается задача кластеризации временных рядов с помощью алгоритма иерархического кластерного анализа, а также задача проверки статистической значимости полученных решений. Проверяется основная гипотеза о том, что в данных нет кластерной структуры. Проверка осуществляется с помощью метода Монте-Карло, который основан на многократной кластеризации искусственно сгенерированной выборки, вычислении индекса качества кластерной структуры и сравнении с индексом качества кластеризации на реальных данных. При генерации временных рядов учитывалась автоковариация, усреднённая по всему набору данных. В качестве выборки был использован реальный набор данных, предоставленный ИЦиГ СО РАН, состоящий из временных рядов, полученных в результате непрерывного мониторинга уровня глюкозы у 385 взрослых пациентов с сахарным диабетом 1 типа. В результате применения алгоритма кластеризации удалось: разделить выборку на кластеры и доказать статистическую значимость полученных решений, сжать данные, и определить паттерны флуктуаций уровня глюкозы.*

Ключевые слова: кластеризация, индекс качества силуэт, временной ряд, гликемический ряд, метод Монте-Карло, статистическая значимость

1 Введение

Во многих областях науки часто встречаются наборы данных, разделённые на подгруппы схожих между собой объектов. В большинстве случаев такое разбиение не является тривиальным или однозначным и представляет интерес для исследователей. Задачу разбиения данных на однородные группы называют задачей кластеризации [1] (синонимы: автоматическая классификация, таксономия).

Неформально задачу кластеризации временных рядов (ВР) можно сформулировать следующим образом: требуется разбить множество временных рядов на группы так, чтобы в одной группе ряды были максимально похожи друг на друга, а ряды из разных групп – не похожи. Под похожестью имеется в виду близость по заданной метрике расстояния. Задача кластерного анализа находит своё применение во многих областях науки: в биологии, медицине, психологии и др. Кластерный анализ применяется в задачах искусственного интеллекта и машинного обучения. В результате кластеризации выборки можно:

- обнаружить скрытые (внутренние) закономерности, которые было бы сложно выявить без разбиения на группы;
- исследовать механизмы, приведшие к этим закономерностям;

- более качественно исследовать генеральную совокупность объектов и выдвигать гипотезы на основе анализа полученных данных;
- обнаружить нетипичные объекты, которые не относятся ни к какому кластеру;
- сжать данные.

При кластеризации временных рядов возникает проблема оценивания качества полученных решений. Существующие методы, основанные, например, на вычислении индексов качества (cluster validity indices) не полностью учитывают специфику решаемой задачи. Наблюдаемые ряды являются реализациями некоторого случайного процесса, обладающего корреляционными свойствами. Эти свойства необходимо учитывать при оценивании качества разбиения на кластеры. В предлагаемой работе формулируется метод проверки статистической значимости результатов кластеризации, основанный на применении статистического моделирования (метода Монте-Карло), с учетом корреляционных свойств ряда. Метод применен в реальной задаче анализа гликемических кривых у больных сахарным диабетом 1 типа.

2 Кластеризация временных рядов

2.1 Постановка задачи

Пусть задано некоторое множество временных рядов. Формально, временной ряд – это реализация некоторого случайного процесса во времени [2], неформально – это некий n-мерный вектор (как правило, вещественный), у которого каждая компонента (за исключением первой) зависит от одной или нескольких предыдущих компонент. Пусть каждый заданный временной ряд – вещественный, тогда можно определить:

1. пространство признаков временных рядов: $X = (x_{ij})$ – j-ый признак i-го ряда, $i = 1, \dots, n, j = 1, \dots, k$, где k – количество признаков, которые можно выделить из временного ряда,n – количество временных рядов (например, x_{i1} – выборочное среднее, x_{i2} – выборочная дисперсия, x_{i3} – минимум/максимум временного ряда i и т.д.). Таким образом, $X \in M_{n \times k}$.
2. $D = \left(d_{pq}\right)_{p,q=1,\dots,n}$ – матрица попарных расстояний, d_{pq} – расстояние между рядами p и q.

Имея 1 или 2, требуется:

- разбить множество временных рядов на группы схожих между собой объектов путём максимизации (минимизации) целевого критерия:

$$N = \bigcup_{i=1}^{s} N_i, N_i \cap N_j = \emptyset, i \neq j$$

$$F(N_1, \dots, N_s) \to max$$

N – объём обучающей выборки, N_i – объекты из кластера i, s – количество кластеров;

- оценить статистическую достоверность кластерной структуры, полученной в результате применения алгоритма кластеризации.

2.2 Проверка статистической значимости результатов кластеризации временных рядов

Пусть задано некоторое множество временных рядов и множество меток кластеров, полученных в результате разбиения исходного множества ВР на k кластеров с помощью какого-либо алгоритма кластеризации. Требуется определить, является ли данное разбиение на кластеры статистически достоверным. Для этого проверяется основная гипотеза о том, что в данных нет кластерной структуры. Пусть имеется:

- X_0 – исходное множество временных рядов, $X_1, \dots, X_t - t$ наборов множеств временных рядов, сгенерированных при условии выполнения основной гипотезы;
- требуемое количество кластеров k;

- количественная мера того, насколько разбиение на кластеры "хорошее", например, некий индекс качества: $S(X_i, k) = s_{i,k}$ – значение индекса на множестве временных рядов X_i при разбиении его на kкластеров;
- уровень значимости α.

Имея указанную информацию, требуется оценить статистическую достоверность полученной кластерной структуры. Если $s_{0,k} > s_{i,k}, i = 1, \dots t$, то кластеризация принимается достоверной, в противном случае – нет. Выполнение данного неравенства является сильным требованием. Иногда (при больших t) оно может не выполняться из-за случайного превышения значением индекса качества для искусственных данных значения индекса качества для реальных данных. Поэтому требование можно ослабить до выполнения следующего неравенства: $s_{0,k} > q_\delta(s_{1,k}, \dots, s_{t,k})$, где $q_\delta(X)$ – квантиль уровня δ множества X.

3 Методы оценивания качества кластеризации

На сегодняшний день существует не так много методов оценивания качества кластеризации, которые бы кардинально различались в своих подходах. Иногда качество кластеризации может определить эксперт, основываясь, например, на визуализации решений или несложном их анализе. Чаще всего качество полученных решений оценивается с помощью индексов качества [3, 4, 5]. Индексов качества существует достаточно много, в данной работе будет подробно рассмотрен только один из них – silhouette score [6], поскольку он непосредственно используется в данном исследовании.

3.1 Индекс качества silhouette score

Для объекта i из кластера C_I определим среднее расстояние до объектов из кластера C_I:

$$a(i) = \frac{1}{|C_I - 1|} \sum_{j \in C_I, j \neq i} d(i,j)$$

и среднее расстояния до объектов из ближайшего кластера C_J:

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j).$$

Тогда индекс силуэт для объекта i определяется следующим образом:

$$s(i) = \frac{b(i) - a(i)}{max\{b(i),\ a(i)\}},$$

для всего набора данных:

$$S = \frac{1}{N} \sum_{i=1}^{N} s(i),$$

N – объем выборки, $d(i,j)$ – некоторое расстояние. Из определения индекса silhouette score видно, что $S \in [-1, 1]$.

3.2 Метод Монте-Карло для проверки статистических гипотез

Для анализа временных рядов в [7] был предложен метод оценивания статистической значимости полученной кластеризации. Данный метод был основан на методах Монте-Карло [8]. Далее будем полагать, что большему значению индекса качества соответствует лучший вариант кластеризации.

Идея метода заключается в следующем: сначала выдвигается основная гипотеза об отсутствии кластеризации. Таким образом, предполагается, что данные формируют один кластер и их нельзя разделить на группы (точнее, алгоритм иерархической кластеризации всегда сможет разбить множество объектов на кластеры, но такая кластеризация не будет представлять никакого интереса для исследователей, и, следовательно, не имеет смысла). В этом случае все объекты можно описать каким-нибудь одним типичным объектом кластера, например, центроидом или медоидом. В [9, 10] можно найти описание основных гипотез, применяемых при проверке статистической значимости кластеризации. Альтернативную гипотезу можно сформулировать как отрицание основной гипотезы – данные можно разбить на

различные кластеры и такое разбиение не является артефактом. Далее генерируются псевдовыборки, соответствующие основной гипотезе. Сгенерированные данные должны быть качественно похожи на реальные данные. На следующем этапе нужно многократно провести кластеризацию на искусственно сгенерированной выборке и определить, для какого количества выделенных кластеров индекс качества на реальных данных окажется больше квантиля уровня $1-\alpha$ для искусственных данных. В этом случае кластеризация принимается достоверной, так как достаточно оснований для того, чтобы отвергнуть основную гипотезу на уровне значимости α.

Метод Монте-Карло для проверки гипотезы о ложной кластеризации

Вход: набор данных, α– уровень значимости, s – количество кластеров, метод вычисления индекса качества кластеризации.

Выход: *true*, если кластеризация достоверна, *false* – иначе.

Шаги:

1. определить значение индекса качества кластеризации для реальных данных при разбиении на s кластеров;
2. сгенерировать k временных рядов длины n согласно основной гипотезе;
3. определить значение индекса качества кластеризации для искусственных данных для s кластеров;
4. повторить шаги 2-3 t раз;
5. вычислить квантиль уровня $1-\alpha$ для искусственных данных.

Если значение индекса качества кластеризации для реальных данных окажется больше квантиля уровня $1-\alpha$ для искусственных данных, то вернуть *true*, в противном случае вернуть *false*.

Описанный метод никак не учитывает то, что объекты кластеризации – это временные ряды. Поскольку у временных рядов, как правило, есть зависимость между соседними компонентами, то это означает, что у многих временных рядов есть автоковариация. Таким образом, идея улучшения метода состоит в том, чтобы учитывать автоковариацию кластеризуемых временных рядов для первых нескольких лагов. Автоковариация для каждого лага усреднялась по всему набору данных. Поскольку автоковариация учитывается при генерации объектов, сгенерированные временные ряды становятся более похожими на временные ряды из реального набора данных, что делает использование метода более обоснованным и подходящим для реальных данных.

На рисунке 1 представлена иллюстрация предложенного метода: показаны примеры распределения индексов качества для генерируемых временных рядов.

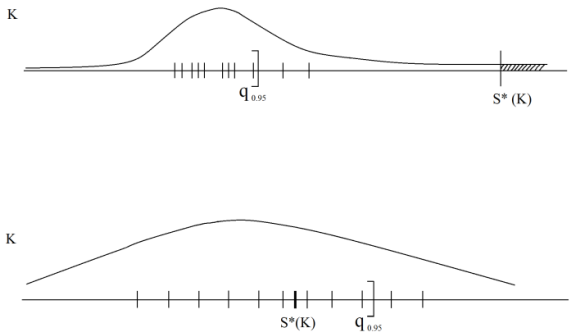


Рис.1. Метод Монте-Карло для проверки статистических гипотез.

Из рисунка видно, что если определить количество кластеров как величину k, и многократно провести кластеризацию на искусственных данных, то значения индекса качества некоторым образом распределятся на отрезке $[-1,1]$. Вероятность того, что индекс качества на искусственных данных может случайно достичь наблюдаемого или ещё большего значения, называют p-значением (p-value).

В данной работе непосредственно p-value не вычисляется, а выводы о достоверности кластеризации делаются на основе сравнения значения индекса качества на реальных данных со значением индекса качества на искусственно сгенерированных данных.

3.3 Генерация искусственного набора данных

Вероятностное распределение, из которого производилась генерация искусственных временных рядов, было многомерным нормальным распределением $N(E, C)$, где E – вектор выборочных средних, C – выборочная ковариационная матрица. Автоковариация использовалась следующим образом: сначала вычислялось выборочное среднее значение автоковариации для lag_1, lag_2 и lag_3 соответственно по всем временным рядам. Затем с учётом этого была вычислена симметричная матрица Тёплица (вектор (1, $mean_lag_1$, $mean_lag_2$, $mean_lag_3$, 0,…0) являлся первой строкой в матрице, где $mean_lag_i$ – усреднённый lag_i по набору данных), которая является ковариационной матрицей для исходного распределения. Наконец, имея ковариационную матрицу, можно сгенерировать многомерный гауссов вектор, который представляет собой синтетический временной ряд. Такой подход к генерации наилучшим образом соответствует исследуемой модели временных рядов, поскольку он учитывает автоковариацию исходных временных рядов.

3.4 Реальный набор данных

Был использован реальный набор данных, предоставленный ИЦиГ СО РАН, состоящий из временных рядов, которые были получены в результате непрерывного мониторинга глюкозы (НМГ) у 385 взрослых пациентов с сахарным диабетом 1 типа. Обследованы пациенты в возрасте от 18 до 70 лет, получавшие базис-болюсную инсулинотерапию. НМГ выполняли с помощью системы iPro™2, MMT-7741 (iPro2) CGM и программного обеспечения CareLink iPro™ (CareLink iPro, MMT-7340) (Medtronic, США). Средняя продолжительность НМГ составила 6,7 дня; диапазон составлял от 3 до 11 дней.

Временной ряд каждого пациента был разбит по часам на временные ряды, соответствующие ночным, ранним утренним и дневным наблюдениям: 0:00 – 5:59, 4:00 – 7:59 и 8:00 – 23:59 с размерностями 72, 48 и 216 соответственно.

В исходных гликемических рядах встречаются пропущенные значения, которые необходимо заполнить для того, чтобы можно было применить алгоритм иерархической кластеризации. Рассматриваются только такие временные ряды, в которых пропущено не более 3 измерений подряд или не более 10% от соответствующей длины ряда измерений всего. Ряды, не удовлетворяющие данным условиям, далее не рассматриваются, так как аппроксимация пропущенных значений в этом случае может быть не точна для дальнейшего анализа. Пропущенные значения были заполнены средним арифметическим между ближайшими справа и слева числовыми значениями. Если пропущены первые или последние значения временного ряда, то пропуски заполняются ближайшим числовым значением. На следующем этапе все временные ряды разделяются на две группы: гликемические ряды и гипогликемические ряды. Заключительным этапом предварительной обработки является стандартизация, она была применена для повышения производительности и качества алгоритма.

Информация о количестве анализируемых временных рядов, а также о количестве временных рядов, которые были удалены из рассмотрения, представлена в таблице 1. Приняты следующие обозначения: N_{init} – количество временных рядов до исключения неподходящих временных рядов; N_{final} – количество временных рядов после исключения неподходящих временных рядов; $Delta$ – количество временных рядов (в %), исключённых из рассмотрения; N_{gl} – количество гликемических рядов; N_{hg} – количество гипогликемических рядов.

Таблица 1. Количество рядов.

	Ночные вр.ряды	***Ранние утренние вр. ряды***	***Дневные вр. ряды***
N_{init}	2956	3036	2954
N_{final}	2782	2729	2156
$Delta$	5.9	10.1	27
N_{gl}	2464	2544	1630
N_{hg}	318	185	526

3.5 Выбор и настройка модели машинного обучения

Кластеризация проводилась с помощью алгоритма иерархического кластерного анализа. Пороговое расстояние, являющееся критерием остановки алгоритма, выбиралось эмпирически, основываясь на значении индекса silhouette score, а также с помощью визуализации решений и экспертной оценки. Метод Уорда [13] был выбран в качестве критерия, отвечающего за объединение кластеров. Чтобы реализовать этот метод, на каждом шаге находится пара кластеров, которая приводит к минимальному увеличению общей внутрикластерной дисперсии после объединения. Исходя из специфичности исходной задачи и данных, евклидово расстояние было выбрано в качестве метрики, отвечающей за расстояние между временными рядами.

Оценка модели в основном проводилась с помощью визуализации данных, оценки индекса «силуэт» и экспертной оценки. Проверка статистической достоверности полученных решений проводилась с помощью методов Монте-Карло, уровень значимости для проверки гипотезы о ложной кластеризации взят равным 0.05. При применении метода рассматривался квантиль уровня 0.95 для искусственных данных. Было установлено, что у всех временных рядов из обучающей выборки есть значимая автоковариация для первых трёх лагов.

4 Результаты

Приведем результаты только для ночных гипогликемических временных рядов, так как они наиболее важны для дальнейшего медицинского анализа (аналогичная работа была проведена и для всех остальных типов ВР).

Всего было кластеризовано 318 гипогликемических временных рядов. При пороговом расстоянии 36 выделено 7 кластеров, представленных на рисунке 2.

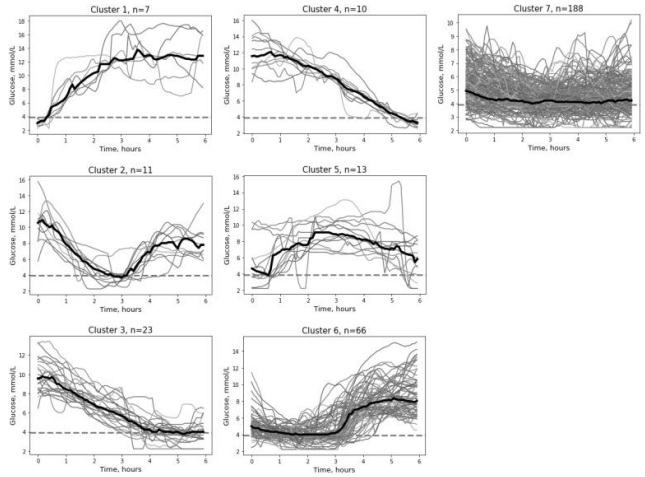


Рис.2. Кластеризация ночных гипогликемических кривых; чёрная кривая – медоид кластера, пунктирная линия – порог гипогликемии (3.9 ммоль/л).

Для большинства кластеров видно, что при исходном завышенном уровне глюкозы (>6 ммоль/л) гипогликемия наступает вследствие нисходящего тренда. В некоторых кластерах гипогликемия наступала сразу с начала ночи.

Типичные кривые, полученные в результате кластеризации, представлены на рисунке 3.

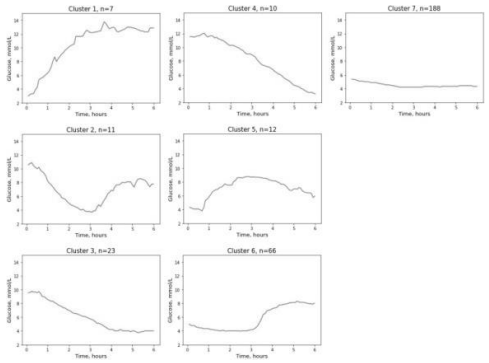


Рис.3. Медоиды ночных гипогликемических кластеров.

Из графиков видно, что нет похожих между собой медоидов, что в очередной раз подтверждает адекватность кластеризации ночных гипогликемических временных рядов.

В результате применения метода был получен график зависимости значений индекса качества от числа выделенных кластеров, представленный на рисунке 4. Параметры метода: количество временных рядов – 318, размерность – 72, количество повторений – 5000.

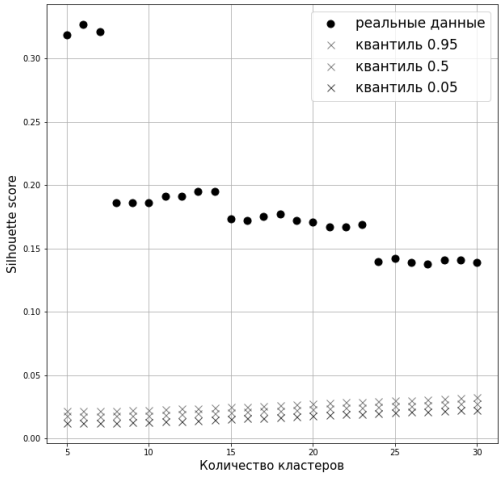


Рис.4. Зависимость индекса силуэт от числа выделенных кластеров.

На графике видно, что значения индекса качества на реальных данных превышает значение квантиля уровня 0.95 индексов качества для искусственных данных для любого числа выделенных кластеров. Из этого можно сделать вывод, что разработанный алгоритм доказал статистическую достоверность кластеризации на уровне значимости $\alpha < 0.05$.

5 Заключение

В работе рассматривался алгоритм иерархической кластеризации временных рядов. Был разработан метод проверки статистической значимости полученных решений, использующий автоковариацию временных рядов из реального набора данных. Разработана программная реализация метода. На основании этого метода был разработан алгоритм кластеризации временных рядов, включающий в себя предварительную обработку данных, иерархическую кластеризацию и проверку статистической значимости полученных решений. Разработанный алгоритм был протестирован на реальных данных, предоставленных ИЦиГ СО РАН. Проведена кластеризация гликемических кривых. Различия в кластерах касались начального и конечного уровня глюкозы, наличия или отсутствия нисходящего/восходящего тренда. Прослеживалась зависимость между исходным значением глюкозы, быстротой падения и временем возникновения гипогликемии. С помощью предложенного метода доказана статистическая достоверность кластеризации. Кроме того, были определены паттерны

флуктуаций уровня глюкозы у больных сахарным диабетом 1 типа в различные периоды суток. Полученные кластеры были проанализированы специалистами эндокринологами; сделаны выводы о корректировке плана лечения в соответствии с наблюдаемыми паттернами динамики.

В дальнейшем планируется разработка алгоритмов прогнозирования временных рядов с использованием их предварительной кластеризации.

Исследование выполнено за счет гранта Российского научного фонда № 20-15-00057-П.

Литература

[1] Ivakhnenko A. G. Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(1):364378, 1971.

[2] Kordik P., Snorek, M. Deterministic Crowding Helps to Evolve Non-correlated Active Neurons In: *Proceedings of*, Academy of Sciences, Glushkov Institute, p. 21-28. Kiev, Ukraine 2005, ISBN 966-02-3734-0

[3] Мандель И. Д. Кластерный анализ. — М.: Финансы и статистика, 1988. — 176 с.

[4] Мишулина О. А. Статистический анализ и обработка временных рядов. — М.: МИФИ, 2004. — С. 180, ISBN 5-7262-0536-7.

[5] Halkidi M., Batistakis I., Vazirgiannis M. On Clustering Validation Techniques // Journal of Intelligent Information Systems. — 2001. — Vol. 17, No. 2/3. — P. 107–145.

[6] Charrad M., Ghazzali N., Boiteau V., Niknafs A. NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. Journal of Statistical Software. — 2014. — Vol. 61, No. 6. —P. 1–36.

[7] Сивоголовко Е. В. Методы оценки качества четкой кластеризации. Компьютерные инструменты в образовании. — 2011. — № 4. — С. 14–31.

[8] Rousseeuw P. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of Computational and Applied Mathematics. — 1987. — Vol. 20. — P. 53–65.

[9] И.Л. Кирилюк, О. В. Сенько. Оценка качества кластеризации панельных данных с использованием методов Монте-Карло. Компьютерные исследования и моделирование. 2020. Т. 12, N 6. С. 1501–1513.

[10] Войтишек А. В. Лекции по численным методам Монте-Карло. – Новосибирск: ИПЦ НГУ, 2018. – 314 с.

[11] Giancarlo R., Utro F. Algorithmic paradigms for stability-based cluster validity and model selection statistical methods, with applications to microarray data analysis. Theoretical Computer Science. — 2012. — Vol. 428. — P. 58–79.

[12] Gordon A. D. Null Models in Cluster Validation // Gaul W., Pfeifer D. (eds) From Data to Knowledge. Studies in Classification, Data Analysis, and Knowledge Organization. — New York: Springer, 1996. — P. 32–44.

[13] Ward J.H. Hierarchical grouping to optimize an objective function // J. of the American Statistical Association, 1963. — 236 p.

К созданию системы определения авторитетности конференций на основе наукометрических данных

Козицын А.С., Шачнев Д.А.

НИИ Механики МГУ им. М.В. Ломоносова, Мичуринский пр., д. 1, г.Москва, 119192, Россия.

alexanderkz@mail.ru, mitya57@gmail.com

Аннотация. *Существующие механизмы оценки авторитетности журналов являются на настоящее время необходимым элементом как наукометрии, так тематического поиска. Существует несколько общепринятых метрик оценки журналов и программные системы для их вычисления. Аналогичные механизмы требуется создавать для определения авторитетности конференций. В работе приводится обзор существующих систем ранжирования конференций и показывается, что они обладают существенными недостатками. основными недостатками существующих рейтингов конференций являются: незначительный охват конференций в компьютерной тематике и практически полное отсутствие индексов конференций по другим тематическим направлениям; отсутствие систем ранжирования русскоязычных конференций; значительное количество ручной работы при построении рангов конференций. В статье приводится обзор основных критериев оценки конференций и описывается разрабатываемой в настоящей момент алгоритм определения авторитетности конференций по данным из наукометрических систем.*

Ключевые слова: граф соавторства, определение сообществ, ранжирование, конференции, наукометрия

1 Введение

Существующие механизмы оценки авторитетности журналов являются на настоящее время необходимым элементом как наукометрии[1], так тематического поиска[2]. При оценке значимости научных статей в наукометрических системах всегда учитывается индекс авторитетности журнала, который принял статью к публикации. Одним из наиболее распространенных показателей авторитетности журнала является импакт-фактор - отношение количества сделанных научным сообществом в текущем году ссылок цитирования на статьи этого журнала за предыдущие несколько лет. Существуют и другие варианты расчета авторитетности журнала [3,4], например, рассчитываемые по данным Scopus показатели SNIP и SJR - нормализованная по тематике цитируемость журнала и сумма цитирований с различным весом, в зависимости от престижа источника. Показатели авторитетности журнала также могут использоваться авторами при выборе журнала для размещения своих публикаций. Источниками оценок авторитетности журналов являются системы научного цитирования, такие как WoS, Scopus, РИНЦ, которые предоставляют необходимые данные по большинству значимых журналов. Также для оценок авторитетности журнала используются факт вхождения журнала в различные списки журналов и их принадлежность определенным группам авторитетности в этом списке. Наиболее известные в России группы: перечень ВАК РФ; ядро РИНЦ, Russian Science Citation Index (RSCI) с разделением на квантили K1-K4; WoS с квантилями Q1-Q4 и другие.

Аналогичные механизмы оценки авторитетности необходимо использовать при построении наукометрических оценок научных докладов и проведения информационного поиска по конференциям. Однако, для конференций отсутствует настолько развитая общепринятая

система вычисления рангов. На настоящий момент существует ряд систем, которые в той или иной степени позволяют проводить оценку авторитетности конференций.

2 Обзор ресурсов оценки авторитетности конференций

Сервис Microsoft Academic [5] является частью Microsoft Knowledge API и представляет Microsoft-партнера системы учета публикаций Google Scholar. Сервис предоставляет API для получения библиометрических индикаторов о конференциях и трудах по компьютерной науке от оригинала Microsoft Academic Seach. На текущий момент компания закрыла доступ к этому ресурсу для пользователей из России.

Рейтинг конференций в области компьютерных наук по данным AMiner [6] построен автоматически на основе данных из систем Google Scholar. Ранжирование происходит на основании метрики H5-Index. Данный рейтинг удобен тем, что конференции разделены на тематические подразделы. Основным недостатком является небольшое количество проанализированных конференций. Список содержит данные о 384 конференциях.

Рейтинг конференций CORE разработан ассоциацией компьютерных исследований и образования Австралии [7]. Доступны версии рейтинга за 2013,2014, 2017, 2018, 2020 и 2021 года. Рейтинг составляется вручную специальным комитетом. За основу каждого следующего рейтинга берутся данные предыдущей версий этого рейтинга и корректируются с использованием набора библиометрических показателей и экспертной оценки представителей экспертного комитета. Классификация конференций производится по уровням A+, A, B и C. В 2021 году комитетом рассматривалось 789 конференций по компьютерной тематике. При анализе учитываются данные о цитировании из Google Scolar (h-5), из Elsevier, данные о цитируемости наиболее значимых статей по метрике системы Aminer и данные Association for Computing Machinery (ACM), который содержит полнотекстовые коллекции всех публикаций, включая журналы, материалы конференций, технические отчеты, информационные бюллетени и книги [8].

Рейтинг LiveSHINE [9] является преемником рейтинга SHINE на основе Google-Scholar. Оригинальный рейтинг SHINE (Simple H-Index Estimator) был набором программных инструментов, предназначенных для расчета H-индекса конференций по информатике на основе данных Google Scholar. В настоящее время LiveSHINE основан на специальном подключаемом модуле (плагине) для браузера Google Chrome. Плагин позволяет администраторам просматривать базу данных конференций LiveSHINE и запускать запросы к Google Scholar для постепенного обновления количества цитирований.

Рейтинг конференций GII-GRIN-SCIE (GGS) разработан группой ученых из Италии и Испании [10]. Ранжирование конференции осуществляется по шкале A++, A+, A,B,B- и Other. Алгоритм расчета рейтинга конференций построен на данных трех рейтингов компьютерных конференций. В 2018 году было проиндексировано около 2.5 тыс конференций с использованием рейтингов: The CORE 2017; Microsoft Academic и LiveSHINE по данным 2018 года. Кроме того, для Microsoft Academic и LiveSHINE рассчитывается еще аналог импакт-фактора - отношение количества цитирований к количеству опубликованных статей. Для расчета данных в 2021 году использовались данные CORE 2021, Microsoft Academic 2021 и LiveSHINE 2018. Для учета цитирований в этом рейтинге при дальнейшей работе предполагается использовать данные Open Science (https://opencitations.net/) - независимой организации для публикации открытых библиографических данных и данных цитирования с использованием технологий Semantic Web (Linked Data). В проекте OpenCitations опубликовано формальное определение открытого цитирования и запущена система глобальных уникальных и постоянных идентификаторов (PID) для библиографических цитирований - идентификаторов открытого цитирования (OCI).

Рейтинг Computer Security Conference Ranking and Statistic (CSCRS) [11,12] составлен частным лицом Guofei Gu и не является официальным. В рейтинг входит около 50 конференций по компьютерным наукам. Для оценки конференции разработчиком рейтинга используются следующие параметры: коэффициент приемки (на странице есть статистика принято/подано докладов), качество и воздействие сборника конференций, качество членов комитета конференции; отношение количество участников конференции к количеству публикаций; место нахождения; история; отраслевое подключение.

Рейтинг CIF [13] разработан частным лицом и вычисляется для 25 конференций в области кибербезопасности. Рассчитывается как

$$CIF \equiv \frac{1}{\frac{d}{a}+\frac{d}{b}+\frac{d}{c}}, \quad (1)$$

где d - количество публикаций по результатам конференции, a - количество заявок, b - количество участников, c - среднее количество цитирований.

Веб-ресурс Conference Ranks [14] предоставляет интерфейс для поиска и просмотра рейтингов конференций из трех источников: рейтинги конференций ERA (2010) (старый вариант рейтинга CORE); рейтинг Microsoft Academic's(2014) и рейтинг Qualis (2012), построеный Бразильским министерством образования.

Веб-ресурс Networking Conferences Statistics [15] не составляет рейтинги конференций, но позволяет просматривать статистические данные двумстам ежегодно проводящимся конференциям по компьютерной тематике. Сайт собирает и предоставляет следующие данные: URL-адрес конференции; количество представленных докладов по годам; количество принятых докладов по годам; коэффициент приемки; количество параллельных дорожек; количество участников. Эти данные могут использоваться для оценки конференций в ручном режиме или построения автоматических рангов.

3 Критерии автоматической оценки конференций

Из приведенного обзора можно сделать вывод, что основными недостатками существующих рейтингов конференций являются: незначительный охват конференций в компьютерной тематике и практически полное отсутствие индексов конференций по другим тематическим направлениям; отсутствие систем ранжирования русскоязычных конференций; значительное количество ручной работы при построении рангов конференций.

В первую очередь, недостаточный уровень автоматизации в этом вопросе объясняется отсутствием хороших каталогов конференций. Кроме того, оказываемое на научное сообщество влияние докладов гораздо труднее оценить, чем влияние статей. Доклады проводятся в устной форме, и интенсивность его обсуждения нигде не фиксируется. Изложенные результаты доклада в развернутом виде часто публикуются в сборниках или в журнальных статьях, которые невозможно связать с исходной конференцией, но которые впоследствии используются слушателями для цитирования.

Альтернативным подходом является создание критериев оценки с использованием специфических для конференции показателей. При ручной оценке конференций можно использовать следующие характеристики: авторитетность докладчиков; количество докладчиков; соотношение количества докладчиков к количеству участников; авторитетность состава программного комитета; время рецензирования заявок; отношение количества принятых докладов к количеству поданных докладов; преемственность; количество и авторитетности участвующих в научной конференции научных коллективов; количество и цитируемость статей с результатами докладов; время существования конференции; место проведения конференции.

Авторитетность докладчиков является одним из наиболее важных критериев оценки значимости конференции. Для оценки авторитетности могут использоваться индивидуальные метрики, например, цитируемость, индекс Хирша и другие аналогичные показатели [16]. Несмотря на то, что индекс Хирша часто подвергается критике за недостаточную объективность[17], он продолжает оставаться стандартом в современной наукометрии. Кроме того, для уточнения расчетов авторитетности можно использовать дополнительные метрики, учитывающие взаимодействие различных ученых и графы соавторства [18].

Количество докладчиков используется для определения масштаба конференции. В приведенных выше ресурсах одним из критериев включение конференции в список высокорейтинговых конференций является преодоление минимального порога на количество докладов конференции. Например, для включения в список CSCRS конференции необходимо иметь не менее 25 докладов.

Соотношение количества докладчиков к количеству участников показывает уровень заинтересованности научного сообщества в конференции и, косвенно, уровень рецензирования докладов. Значение этого показателя равное 1 может свидетельствовать о том, что большинство участников заинтересовано только в представлении своих результатов. Низкое значение

показателя означает, что значительное количество участников пришли получить информацию и не считают себя готовыми представить свои результаты на конференции такого уровня.

Авторитетность состава программного комитета можно оценивать также, как и авторитетность докладчиков по различным наукометрическим метрикам. Высокий научный уровень членов программного комитета необходим для обеспечения хорошего уровня рецензирования докладов.

Время рецензирования заявок позволяет быстро отфильтровывать конференции «для срочных публикаций». У подобных конференций, как правило, срок подачи заявок заканчивается всего за несколько дней до начала конференции. Авторитетные конференции заканчивают прием докладов не менее чем за несколько месяцев до начала мероприятия.

Отношение количества принятых докладов к количеству поданных докладов показывает уровень отбора материалов для выступлений. Основной трудностью для использования этого критерия является сложность получения объективных данных о реальном количестве поданных заявок.

Преемственность можно оценивать как мощность пересечения множеств докладчиков конференций разных лет. Слишком низкий процент пересечения означает отсутствие постоянного контингента, слишком высокий – отсутствие притока новых участников и стагнацию.

Количество и авторитетности участвующих в научной конференции научных коллективов. Незначительное количество научных коллективов (или, как крайний случай, принадлежность всех докладчиков одному научному коллективу) свидетельствует о том, что на конференции не происходит обмена научной информацией. Такая конференция не решает основную задачу проведения подобных мероприятий. Вместе с тем, отсутствие пересечений научных коллективов у всех докладчиков может свидетельствовать о слишком широком профиле конференции и отсутствии общих научных интересов у присутствующих на конференции участников. Особенно, это характерно для конференций с близким к единице отношением количества докладчиков к количеству участников.

Количество и цитируемость статей с результатами докладов является хорошим показателем степени влияния конференции на научное сообщество. Однако, автоматизация расчета этого критерия является сложной технической задачей. Во многих случаях, авторы публикуют перед началом конференции тезисы или материалы конференции, а после проведения конференции по результатам обсуждения пишут полноценную статью в журнал или сборник. Название такой статьи, как правило, отличается от названия доклада. Определение степени связи такой статьи с исходным докладом на конференции требует семантического анализа и сравнения текста статьи и доклада.

Время существования конференции может также необходимо учитывать при определении ее авторитетности. Большое время существования может свидетельствовать о достаточно проработанной схеме ее организации и достаточной известности.

Место проведения конференции также может косвенно определять популярность и известность конференции. Однако, оценка этого показателя в автоматическом режиме является затруднительной.

При выборе используемых в каждом конкретном случае показателей необходимо учитывать доступные источники достоверных данных для проведения анализа. Данные по конференциям должны быть достаточно полные и верифицируемые. Ручной сбор данных и верификация непосредственно со страниц конференций, применяемый, в том числе, в рангах CORE, CSCRS, CIF не способен обеспечить большой охват анализируемых конференций.

4 Ранжировании конференций по наукометрическим данным

При разработке алгоритма ранжирования конференции авторы этого доклада используют данные наукометрической системы ИАС ИСТИНА[19]. В системе содержаться данные о докладах, сделанных в различное время сотрудниками тридцати зарегистрированных в системе организаций, в том числе, сотрудниками МГУ им. М.В. Ломоносова.

Поскольку данные о конференциях вносятся пользователями, то на первом этапе обработки данных осуществляется удаление дубликатов и определения связей между

конференциями, проводимых в разные года [20]. На текущий момент в системе зарегистрировано около 14 тысяч конференций с длительностью истории проведения от 1 года до 20 лет.

Еще одним важным элементом предварительной обработки данных является построение научных коллективов на основе анализа графа соавторства[21]. В разных работах предлагается использовать различные определения научного коллектива. В работах [22,23] научные коллективы определяются как полные клики с различными ограничениями на вес ребер графа соавторства, в работе [24] используется мера модулярности Q[25]

$$Q = \frac{1}{2m}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j), \qquad (2)$$

где n - количество вершин в графе, m - количество ребер в графе, A_{ij} -симметричная матрица весов ребер, k_i - количество ребер вершины i, c_i -номер кластера вершины i, $\delta(x, y) \equiv \begin{cases} 1, x = y \\ 0, x \neq y \end{cases}$.

Разбиение на научные коллективы (кластеры) считается хорошим, если для разбиения выполняется условие $Q > 0.7$. В работе [26] предлагается использовать G-aut алгоритм для построения разбиения графа соавторства на научные коллективы, в работе [27] описывается применение SNA моделей анализа социальных сетей. В данной работе для определения научных коллективов используется подход [28], определяющий научные коллективы как связные компоненты графа соавторства с минимальным порогом на веса ребер.

Одним из основных вопросов при классификации объектов является выбор наблюдаемых параметров для обучения. Параметры «время рецензирования заявок» и «отношение количества принятых докладов к количеству поданных докладов» в данных наукометрической системы ИАС ИСТИНА отсутствуют и не могут использоваться для ранжирования конференций с использованием этих данных. Параметры «количество докладчиков» и «количество участников» заполняются пользователями при создании конференции ИАС ИСТИНА, но обладают низкой достоверностью. С целью автоматизации процесса сбора этих данных проводились изыскания в области автоматического поиска и распознавания анонсов и программ конференций. Однако, на текущий момент, результаты апробации созданных алгоритмов признаны неудовлетворительными.

Параметры «авторитетность докладчиков», «авторитетность состава программного комитета» и «преемственность» рассчитываются на основании зарегистрированных в системе докладов сотрудников. Такой подход позволяет получить приблизительную оценку, которая может отличаться от точной оценки, рассчитанной по всем участникам конференций. Вариант приближенной оценки выбран вследствие того факта, что получение списка всех докладчиков для конференции требует не только автоматизации процесса сбора и обработки программ конференций, но и качественной работы алгоритма разрешения неоднозначности имен участников без ручной верификации результата.

Количество и авторитетности участвующих в научной конференции научных коллективов определяется также по зарегистрированным в системе докладам. Для каждого автора зарегистрированного доклада вычисляется научный коллектив и его авторитетность.

Для оценки количества и цитируемость статей с результатами докладов автоматически возможно выделять только сборники и тезисы докладов. Их названия, как правило, совпадают с названием доклада. Разделение материалов конференции на сборники и тезисы позволяет получить дополнительный параметр для оценки, однако, их цитируемость достаточно низкая, поскольку другие авторы предпочитают вместо тезисов включать в библиографический список статьи в журналах.

Время существования конференции оценивается на основе рассчитанных на предыдущем этапе связей между конференциями, проводимых в разные года [20].

Использование данных о месте проведения конференции признано на первом этапе работ не целесообразным, поскольку в большинстве случаев при описании места проведения конференции указывается только город.

Построение интегрированной оценки предполагается делать с использованием алгоритмов обучения на существующих ручных списках ранжирования CORE, GGS, CSCRS и на составленных вручную списках российских конференций, зарегистрированных в наукометрической системе ИАС ИСТИНА.

5 Заключение

Существующие на настоящий момент системы определения авторитетности не обеспечивают значительного охвата конференций, особенно, в русскоязычном сегменте. Представленный подход к автоматическому построению рангов позволит на основании выбранных критериев проводить автоматическое определение авторитетности значительного количества русскоязычных конференций. С увеличением количества зарегистрированных в системе ИАС ИСТИНА организаций полнота индекса будет увеличиваться.

Литература

[1] Садовничий В. А., Васенин В. А. Интеллектуальная система тематического исследования наукометрических данных: предпосылки создания и методология разработки. Часть 1. *Программная инженерия.* - 2018. - Т. 9, № 2. - С. 51-58.

[2] Козицын А. С. Алгоритмы тематического поиска данных в наукометрических системах // Программная инженерия. - 2022. - Т. 13, № 6. - С. 291-300.

[3] URL:https://library.bntu.by/laboratorija-nauchnyh-kommunikacij/indeksy-citirovaniya-i-bibliometricheskie-pokazateli/ (02.06.2023)

[4] URL:http://bonus.basnet.by/bibliometricheskie-pokazateli/drugie/ (02.06.2023)

[5] URL:https://academic.microsoft.com/ (02.06.2023)

[6] URL:https://aminer.org/ranks/conf (02.06.2023)

[7] URL: http://www.core.edu.au/conference-portal (02.06.2023)

[8] URL: https://library.hse.ru/mirror/pubs/share/471157926.pdf (02.06.2023)

[9] URL: http://liveshine.icomp.ufam.edu.br/ (02.06.2023)

[10] URL: https://www.informatics-europe.org/images/national-associations/GII-GRIN-SCIE_Conference_Rating_System.pdf (02.06.2023)

[11] URL: http://faculty.cs.tamu.edu/guofei/sec_conf_stat.htm (02.06.2023)

[12] URL: https://people.engr.tamu.edu/guofei/sec_conf_stat.htm (02.06.2023)

[13] URL: http://jianying.space/conference-ranking.html (02.06.2023)

[14] URL: http://www.conferenceranks.com/ (02.06.2023)

[15] URL: https://sites.cs.ucsb.edu/~almeroth/conf/stats/ (02.06.2023)

[16] Михайлов О. В. Цитируемость и библиометрические показатели российских ученых и научных журналов. *Проблемы деятельности ученого и научных коллективов.* - СПб. , 2017. - № 3 (33). - С. 152-170

[17] URL:https://sysblok.ru/society/skazhi-mne-kakoj-u-tebja-indeks-hirsha-i-ja-skazhu-kto-ty/ (02.06.2023)

[18]Козицын А. С., Афонин С. А., Шачнев Д. А. Алгоритм поиска по ключевым словам специалистов в заданной предметной области. *Современные информационные технологии и ИТ-образование.* - 2021. - Т. 17, № 1. - С. 34-47

[19]Интеллектуальная система тематического исследования научно-технической информации (ИСТИНА) / В. А. Васенин, С. А. Афонин, А. С. Козицын и др. *Обозрение прикладной и промышленной математики.* - 2012. - Т. 19, № 2. - С. 239-240.

[20]Козицын А. С. Алгоритмы поиска дубликатов научно-технических конференций и групп конференций в наукометрических системах. *Программная инженерия.* - 2023. - Т. 14, № 4. - С. 195-202.

[21]Заславский Д. Д., Афонин С. А. Поиск коллективов авторов по метаданным публикаций среди пользователей наукометрической системы с использованием алгоритма быстрого взвешенного поиска wand *Ломносовские чтения 2021. — Секция механики.* — Москва МГУ, 2021. — С. 91–91.

[22]Козицын А. С., Афонин С. А., Шачнев Д. А. Использование графа соавторства для тематического поиска конференций по наукометрическим данным. *Электронные библиотеки.* - 2023. - Т. 25, № 6. - С. 600-615.

[23]Афонин С. А., Гаспарянц А. Э. Автоматическое построение функции оценки качества в задаче разрешения неоднозначности имен авторов научных публикаций. *Программная инженерия.* - 2015. - № 10. - С. 31-37.

[24]Newman M.E., Girvan M. Finding and evaluating community structure in networks. *Physical Review* E. 2004. Vol. 69(2). P 026113. url:https://arxiv.org/pdf/cond-mat/0308217.pdf

[25]Malliaros F.D., Vazirgiannis M. Clustering and community detection in directed networks: A survey. *Physics Reports.* 2013. Vol. 533, Iss. 4. P. 95-142. url:https://arxiv.org/pdf/1308.0971.pdf

[26]Blondel V.D., Guillaume J-L., Lambiotte R., Lefebvre E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*. 2008. P 10008. url:https://arxiv.org/pdf/0803.0476.pdf

[27]Басараб М.А., Глинская Е.В., Иванов И.П., Колесников А.В., Кузовлев В. И., Исследование структуры графа научного соавторства методами анализа социальных сетей. *Вопросы кибербезопасности*, N1(19), 2017, 31-36 url:https://elibrary.ru/download/elibrary_28391583_94385659.pdf

[28]Козицын А. С., Афонин С. А. Разрешение неоднозначностей при определении авторов публикации с использование графов соавторства в больших коллекциях библиографических данных // *Программная инженерия*. - 2017. - Т. 8, № 12. - С. 556-562.

Метод визуализации, упрощающий понимание данных и результатов выравнивания сущностей

Колганова А. В.[1], Апанович З.В.[2]

[1]*Новосибирский государственный университет, ул Пирогова 2, Новосибирск, 630090, Россия*
[2]*Институт Систем Информатики СО РАН, пр. Лаврентьева 6, Новосибирск, Россия*

a.kolganova@g.nsu.ru, apanovich_09@mail.ru

Аннотация. *В данной работе представлен алгоритм визуализации, упрощающий понимание результатов работы алгоритмов выравнивания сущностей для разноязычных графов знаний, полученных на основе так называемых эмбеддингов, векторных представлений низкой размерности. Алгоритм совмещает обычную визуализацию эмбеддингов посредством проекции на плоскость с графовой визуализацией. Набор опций разработанного инструмента визуализации упрощает понимание не только результатов работы различных алгоритмов выравнивания сущностей, но и структуры тестового набора данных. Алгоритм позволяет визуализацию графов знаний в виде обобщенного графа типов, визуализацию сущностей заданного типа и их отношений, отношений, возможность переключения между режимами графовой визуализации и визуализации эмбеддингов, построенных различными алгоритмами выравнивания сущностей*

Ключевые слова: разноязычные графы знаний, алгоритмы выравнивания сущностей, эмбеддинги, визуализация

Введение

Графы знаний (ГЗ) используются все в большем количестве предметных областей, и все большее количество приложений использует этот тип представления для хранения данных без потери их семантики. Поэтому все чаще возникает необходимость сравнить или объединить информацию об одной и той же сущности, описанной в разных графах знаний. Частным случаем этой задачи является задача сравнения и/или объединения информации об одной и той же сущности, описанной в разноязычных графах знаний. Эта задача имеет много разных названий. Мы будем использовать название *кросс-языковая идентификация сущностей* или же *кросс-языковое выравнивание сущностей*. В современных реалиях задача кросс-языковой идентификации сущностей тесно связана с задачей импортозамещения, например поиском эквивалентных лекарств, запчастей или устройств для Интернета Вещей.

Графы знаний хранят факты в виде реляционных и литеральных триплет. Реляционные триплеты изображают отношение между двумя сущностями (объектами, имеющими уникальный Интернет-идентификатор, IRI) и имеют формат *tr_r* = (*субъектная сущность, отношение, объектная сущность*). Литеральные триплеты хранят информацию об атрибутах сущностей и имеют формат *tr_l* = (*субъектная сущность*, *атрибут, литеральное значение*).

При визуализации литеральные значения принято изображать прямоугольниками, а сущности или объекты реального мира – овалами. На рисунке 1 показаны фрагменты из англоязычной и русскоязычной версий DBpedia, описывающие советско-финский фильм «За спичками». Примером реляционной триплеты является триплета (*За спичками*, *режиссер*, *Гайдай, Леонид Иович*), а примером литеральной триплеты является триплета (*За спичками*, *длительность, 100 минут*). На рисунке красными линиями изображены отношения *owl:sameAs*, имеющиеся между сущностями в русскоязычном и англоязычном графах знаний. Можно видеть,

что англоязычной сущности *dbr:Borrowing_Matchsticks* в англоязычном графе знаний соответствует русскоязычная сущность *За спичками,* англоязычному предикату *dbo:director* – русскоязычный предикат *режиссер*, а англоязычной сущности *Leonid_Gaidai* – русскоязычная сущность *Гайдай, Леонид Иович.* Понятно, что англоязычный и русскоязычный списки сценаристов, режиссеров и актеров, сыгравших роли в этом фильме, должны бы совпадать. Однако в реальных графах знаний наблюдаются некоторые различия в описаниях эквивалентных сущностей. Прежде всего, следует обратить внимание на разное написание имен в русскоязычной и англоязычной версиях графов знаний. В русскоязычной версии указывается сначала фамилия, потом имя и отчество, а в англоязычной версии только имя и фамилия. Кроме этого, авторы сценария этого фильма в англоязычной версии графа знаний указаны как сущности, имеющие уникальный идентификатор. Соответственно, они изображены при помощи эллипсов. В русскоязычной версии графа знаний большинство авторов сценария указаны как обычные текстовые атрибуты, а не сущности. Поэтому даже если имена сущностей и литералов похожи (например, Владлен Бахнов, сценарист), они не могут быть признаны эквивалентами и связаны отношениями *owl:sameAs*. Также, длительность фильма в англоязычной версии указана в секундах, а в русскоязычной версии – в минутах. Понятно, что наиболее полное описание сущности можно получить объединением всех триплет, описывающих одну и ту же сущность. Но для решения этой задачи должно быть правильно установлено соответствие между сущностями.

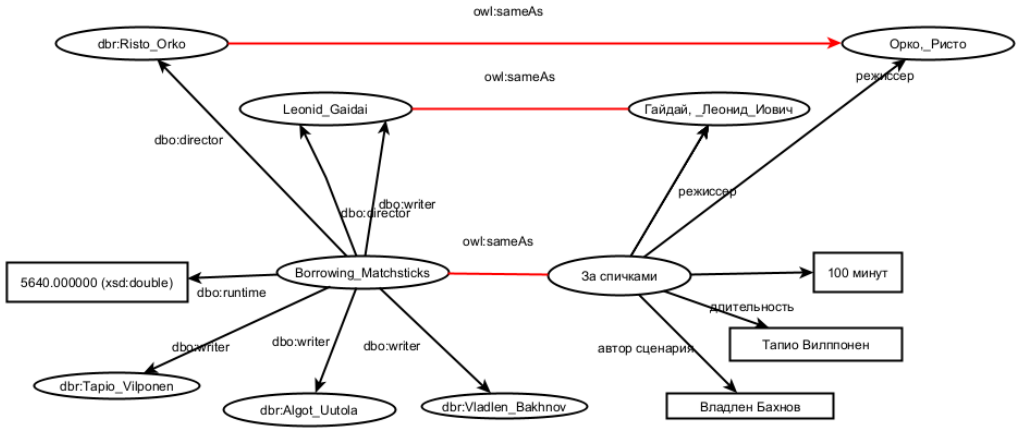


Рис. 1. Соответствие между англоязычными и русскоязычными сущностями.

В последние годы получили распространение методы установления соответствия между сущностями различных графов знаний, использующие так называемые «эмбеддинги» (embeddings), векторные представления заданной размерности для сущностей и отношений графов знаний. Достоинствами подхода на основе эмбеддингов являются высокая масштабируемость и небольшие усилия при подготовке обучающих выборок.

В работе [1] представлена библиотека OpenEA, содержащая около дюжины повторно реализованных алгоритмов выравнивания сущностей на основе различных стратегий, а также результаты экспериментов с этими алгоритмами на тестовой выборке, содержащей англо-немецкие, англо-французские и англо-китайские данные. Более поздние эксперименты с расширенным множеством алгоритмов выравнивания описаны в работе [2].

Понятно, что русскоязычному пользователю интересны, прежде всего, эксперименты, использующие русскоязычные данные. В работе [3] описан русско-английский набор данных для экспериментов с алгоритмами кросс-языкового выравнивания сущностей. В работе [4] показано, что качество выравнивания сущностей можно существенно улучшить, улучшая качество построения эмбеддингов для имен сущностей. Также были найдены наилучшие комбинации методов построения векторных эмбеддингов имен сущностей.

1 Совместная визуализация результатов выравнивания и структуры набора данных

Естественным продолжением этой работы является желание понять, как именно влияет структура самого графа знаний на качество выравнивания. Для решения этой задачи необходимо было использовать визуализацию как входных данных, так и результатов выравнивания

сущностей в графовом виде. Для экспериментов с визуализацией были выбраны такие методы как SEU [5] и RDGCN [6].

Метод RDGCN (Relation-aware Dual-Graph Convolutional Network) [6] использует для построения векторных представлений не только структуру исходных графов знаний (primal entity graph), но и вспомогательные графы, двойственные по отношению к исходным графам (dual relation graph), вершинами которых являются ребра исходных графов. Для осуществления взаимодействия между исходными графами знаний и двойственными реляционными графами используется механизм графовых сетей внимания (Graph Attention Networks, GAT). Результирующие векторные представления исходных графов затем подаются в графовые сверточные сети (Graph Convolutional networks, GCN), для извлечения информации о структуре окружений вершин.

Метод выравнивания сущностей под названием SEU [5] (Simple but Effective Unsupervised EA method) не использует нейронных сетей. Основная идея SEU состоит в сведении задачи EA к давно известной задаче назначения, для которой существует хорошо известный венгерский алгоритм решения.

RDGCN и SEU показали очень хорошие результаты на стандартных наборах тестовых данных и по этой причине были выбраны для экспериментов с визуализацией результатов выравнивания. Для оценки качества работы алгоритмов выравнивания сущностей в данной работе используется метрика hits@1, которую достаточно просто оценить визуально. По определению, метрика hits@k=n% означает, что для n процентов объектов из одного графа знаний эквивалентный объект из второго графа знаний находится среди ближайших k соседей в пространстве эмбеддингов. Метрика hits@1, равная 100%, соответствует алгоритму, который самостоятельно строит правильные отношения *owl:sameAs* между сущностями.

Стандартным способом визуализации результатов работы всевозможных методов, использующих «эмбеддинги» сущностей в графах знаний, является построение проекции этих эмбеддингов на плоскость при помощи метода t-sne [7]. На рисунке 2 показана визуализация результатов выравнивания сущностей русскоязычного и англоязычного графов знаний, построенных методом SEU. До определенного момента такие визуализации могут быть полезными. Например, именно благодаря такому типу визуализации, нам удалось установить влияние перевода имен сущностей на качество выравнивания сущностей и подобрать наиболее удачную комбинацию методов построения векторного представления имен сущностей. Однако, такая визуализация не позволяет оценить визуально влияние количества и качества предикатов, связанных с различными сущностями.

Одним из возможных подходов может быть разработка визуализации, использующей как стандартную визуализацию векторных представлений сущностей, так и визуализацию интересующих фрагментов тестовых данных в виде графа.

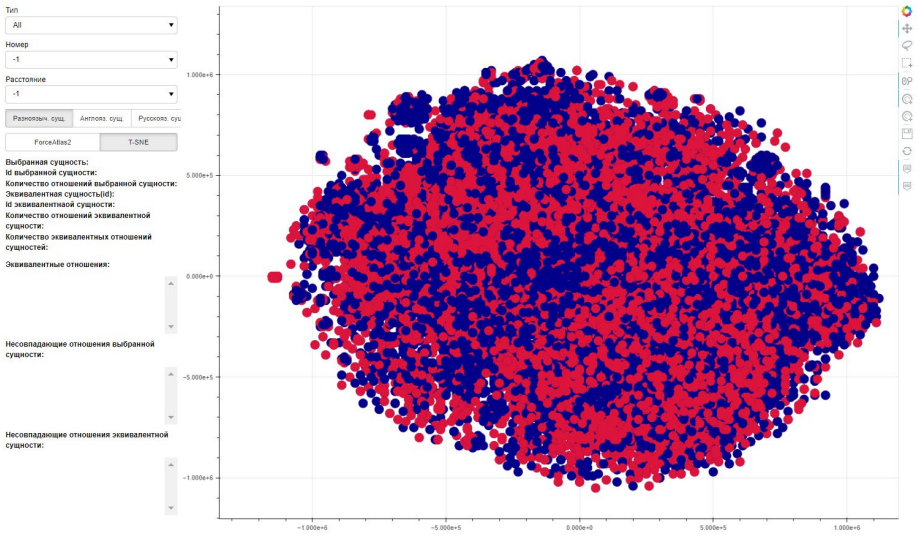


Рис. 2. Визуализация результатов выравнивания русскоязычного и англоязычного графов знаний, построенных методом SEU.

К сожалению, полная визуализация тестового набора данных, содержащего 30 000 сущностей из англоязычного и русскоязычного графов знаний, а также предикатов, связанных с этими сущностями, требует нескольких часов. Поэтому для визуального представления информации о графовой структуре текущего набора данных была построена визуализация, агрегирующая все пары сущностей, имеющих один и тот же тип, согласно онтологии DBpedia https://www.dbpedia.org/resources/ontology/. Для имеющегося тестового набора данных оказалось, что не все эквивалентные сущности из англоязычного и русскоязычного графов знаний имели один и тот же тип, поэтому была написана программа, подбиравшая для каждой пары сущностей наиболее близкий общий тип. [4].

На рисунке 3 показана визуализация тестового набора данных в «компактном» виде. Каждая вершина визуализации соответствует всем сущностям и русскоязычным и англоязычным, имеющим один и тот же тип в онтологии DBpedia. Размер каждой окружности пропорционален общему количеству сущностей указанного типа, а толщина ребер пропорциональна общему количеству связей между отдельными сущностями Для визуализации графовой структуры использован силовой алгоритм ForceAtlas2 [8]. Названия изображаемых типов сущностей «всплывают» при подведении мыши к каждой вершине графа.

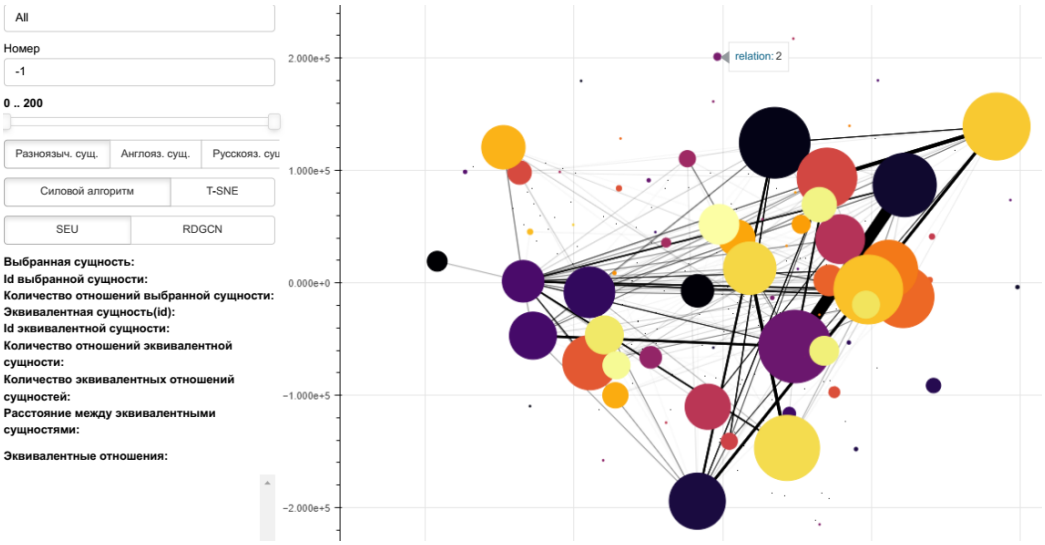


Рис. 3. Визуализация тестового набора данных в виде графа, вершинами которого являются типы сущностей, описанные в онтологии DBpedia.

Можно видеть, что разные типы сущностей представлены не равномерно, но разброс в размерах отражает размеры всего набора данных. Больше всего сущностей принадлежат классу *Film* (5510 сущностей), *Album* (3570), *SoccerPlayer* (2398), *Artist* (2334). Также легко заметить, что наибольшее количество связей наблюдается между такими типами сущностей как *Film* и *Artist*. При этом имеются и совсем крохотные множества сущностей одного типа, такие как *Monarch.*

В правой части изображения имеется панель управления визуализацией. Вверху расположено два выпадающих списка. По умолчанию, первый выпадающий список выдает пользователю все типы сущностей, имеющихся в двух наборах данных (Режим *All*). Пользователь может выбрать из выпадающего списка любой тип сущностей, имеющийся в двух наборах данных. Второй выпадающий список позволяет выбрать для визуализации любую конкретную сущность по ее идентификатору. По умолчанию никакая из конкретных сущностей не выбрана (режим -1). Ползунок, расположенный под всплывающими списками, позволяет выбирать все сущности из двух графов знаний, расположенные на расстоянии, заданном ползунком. Ниже расположены три группы кнопок. Самая верхняя группа, состоящая из трех кнопок, позволяет выбирать для визуализации сущности либо русскоязычного, либо англоязычного графа знаний, либо сущности из обоих графов. Вторая группа, состоящая из двух кнопок, позволяет выбирать режим визуализации: либо графовую визуализацию графов знаний (режим «Силовой алгоритм»), или же визуализацию векторных представлений сущностей (режим t-sne). Наконец, самая нижняя группа из двух кнопок позволяет выбрать результаты выравнивания сущностей одним из двух алгоритмов, либо RDGCN или SEU. В ближайшем будущем эти две кнопки будут заменены

одним выпадающим списком, содержащим результаты всех доступных результатов выравнивания сущностей.

Для более детального анализа результатов работы алгоритмов выравнивания сущностей программа визуализации предоставляет возможность выбрать только сущности заданного типа, а затем рассмотреть связи между этими сущностями. Такой просмотр возможен как в режиме проекции векторного представления на плоскость, так и в режиме визуализации графа.

В работе [4] была сделана попытка проанализировать связь качества выравнивания сущностей с такими атрибутами сущностей как имя сущности и количество отношений сущности. Анализ метрики *hits@k* показал, что для таких методов выравнивания как MultiKE, RDGCN и SEU имелись типы сущностей, такие как *Country, University, Painter*, для которых метрика *hits*@1 была близка 100 процентам. При этом были и типы сущностей такие как *Insect*, для которых *hits*@1 достигала всего 44 процента. На рисунке 4 показано распределение сущностей типа *Country*, полученное методом выравнивания RDGCN, а на рисунке 5 - распределение сущностей типа *Country*, полученное методом выравнивания SEU.

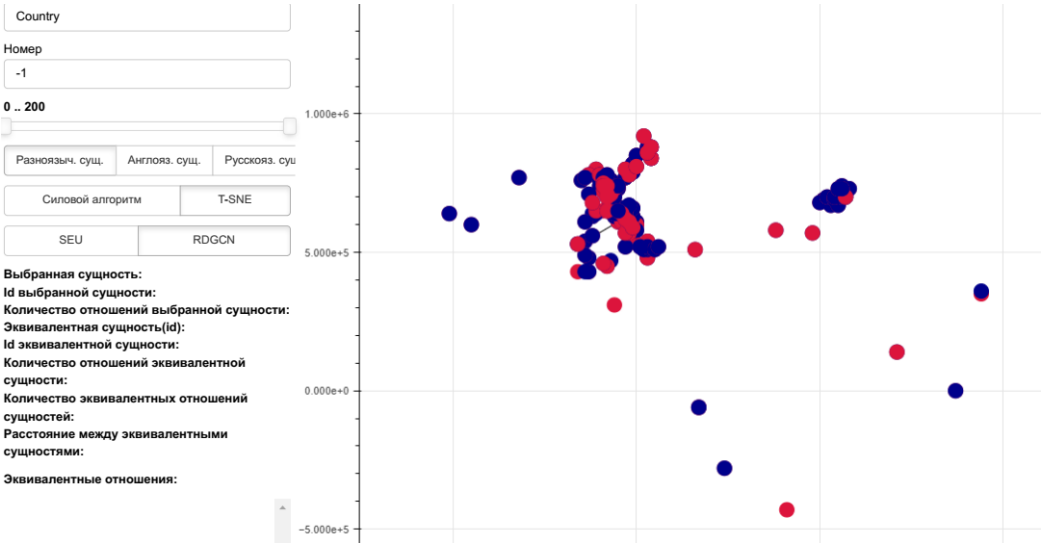


Рис. 4. Распределение сущностей типа Country, полученное методом выравнивания RDGCN

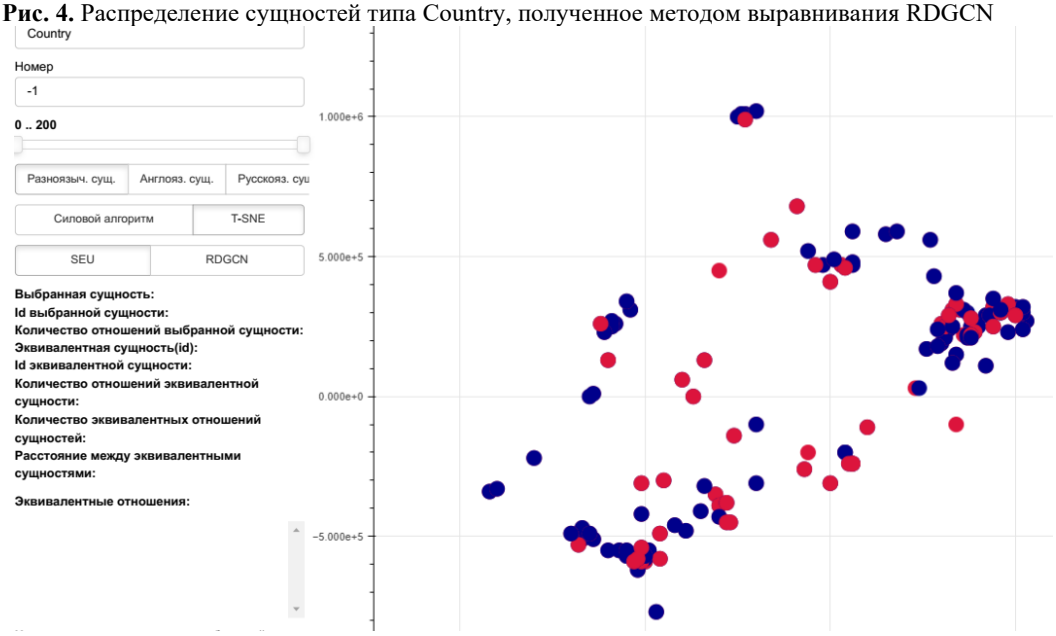


Рис. 5. Распределение сущностей типа *Country*, полученное методом выравнивания SEU.

Оба изображения демонстрируют значительный разброс сущностей, но при более детальном просмотре выясняется, что эквивалентные русскоязычные и англоязычные сущности имеют одну

и ту же координату, а расстояние между ними близко к нулю, что говорит об очень хорошем качестве выравнивания. Программа визуализации позволяет еще более детальный анализ результатов выравнивания. Для этого на панели слева имеется возможность выбора конкретной сущности заданного типа, например, сущности под номером Id231, описывающей русскоязычную сущность *Россия*. При выборе этой сущности программа визуализации покажет эквивалентную ей англоязычную сущность *Russia*, указанную в наборе данных под номером Id50. Также будут показаны все предикаты англоязычной и русскоязычной сущностей, количество эквивалентных предикатов для этих сущностей и расстояние между этими сущностями, соответствующее заданному алгоритму выравнивания, в данном случае, SEU. На панели слева также имеется слайдер, позволяющий выбирать все сущности заданного типа, расположенные в определенном диапазоне расстояний друг от друга. Такой тип визуализации показан на рисунке 6.

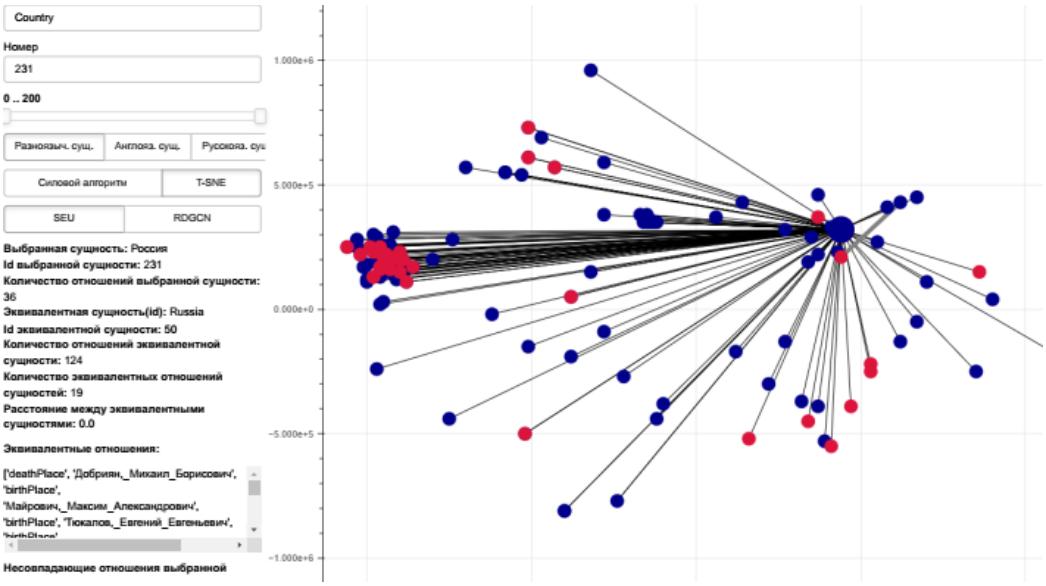


Рис. 6 Распределение соседей англоязычной и русскоязычной сущностей Россия и Russia**.**

На панели слева можно видеть, что расстояние между векторными представлениями этих сущностей равно нулю, и поэтому увеличенная синяя окружность в центре изображения представляет не только англоязычную, но и русскоязычную сущность. Просто две окружности имеют одну и ту же координату. Также на панели слева показаны все предикаты этих эквивалентных сущностей и количество одинаковых предикатов, равное девятнадцати. Недостатком визуализации векторных представлений сущностей является большое количество наложения эквивалентных сущностей, что усложняет понимание изображения. Именно для таких случаев предназначена визуализация данных в виде графа, которая показывает эквивалентные сущности и их соседей без наложений. Такое изображение показано на Рис. 7. Эквивалентные сущности *Россия* и *Russia* показаны увеличенными окружностями красного и синего цвета соответственно. Между каждой парой их эквивалентных соседей имеется серое утолщенное ребро, соответствующее отношению *owl:sameAs.* Это изображение дает интуитивную подсказку, почему между сущностями типа *Country* наблюдается такое хорошее выравнивание несколькими рассмотренными алгоритмами выравнивания. Судя по всему, это связано, во-первых, со стандартными способами перевода названий стран с одного языка на другой, а во-вторых с большим количеством эквивалентных сущностей-соседей. Более тонкие детали надо анализировать на уровне конкретного алгоритма.

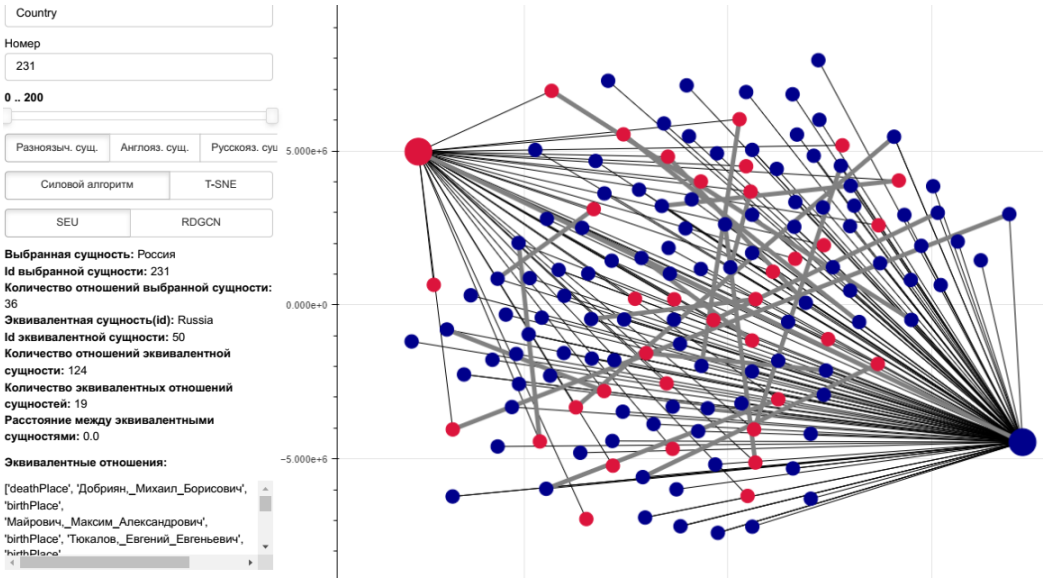


Рис. 7. Графовая визуализация эквивалентных сущностей Россия и Russia**.**

Для сравнения рассмотрим теперь визуализацию результатов выравнивания для сущностей типа Insect, на котором получены самые низкие оценки метрики hits@1. На рисунке 8 показаны результаты выравнивания сущностей типа Insect полученные методом SEU. Тестовый набор содержит всего 36 сущностей этого типа и расстояния между эквивалентными сущностями превышают 100. Выбирая визуализацию конкретных сущностей этого типа, можно увидеть, во-первых, что в англоязычном наборе данных названия этих сущностей приводятся на латыни Id 29424 Cryptocephalus поэтому при переводе названия русскоязычной сущности на английский не возникает текстового сходства между названиями сущностей Скрытоглавы – Hidden heads), а предикаты этих сущностей весьма немногочисленны. На рисунке 8 показана визуализация существующих отношений эквивалентных сущностей Id29424 Cryptocephalus и Id23379 Скрытоглавы_(род).

Таким образом, можно сделать вывод, что два различных метода выравнивания сущностей дают весьма сходные результаты на различных типах сущностей в основном из-за специфических особенностей тестового набора данных. Во-первых, хорошие результаты выравнивания наблюдаются в случае текстового совпадения названий сущностей и значительного количества эквивалентных предикатов у эквивалентных сущностей. Аналогично, плохие результаты возникают при значительном разбросе в названиях сущностей и наличии эквивалентных сущностей с низким количеством эквивалентных предикатов. Причины разброса в названиях сущностей хорошо диагностируются именно благодаря визуализации. Наиболее частой причиной такого несоответствия является то, что, например, англоязычный набор данных использует неанглоязычные названия сущностей определенного типа. Пример с сущностями типа Insect уже был продемонстрирован, но в процессе экспериментов были обнаружены многочисленные несоответствия этого типа. Таким образом можно утверждать, что предлагаемый метод визуализации полезен не только при анализе результатов работы алгоритмов выравнивания, но и при анализе тестовых наборов данных.

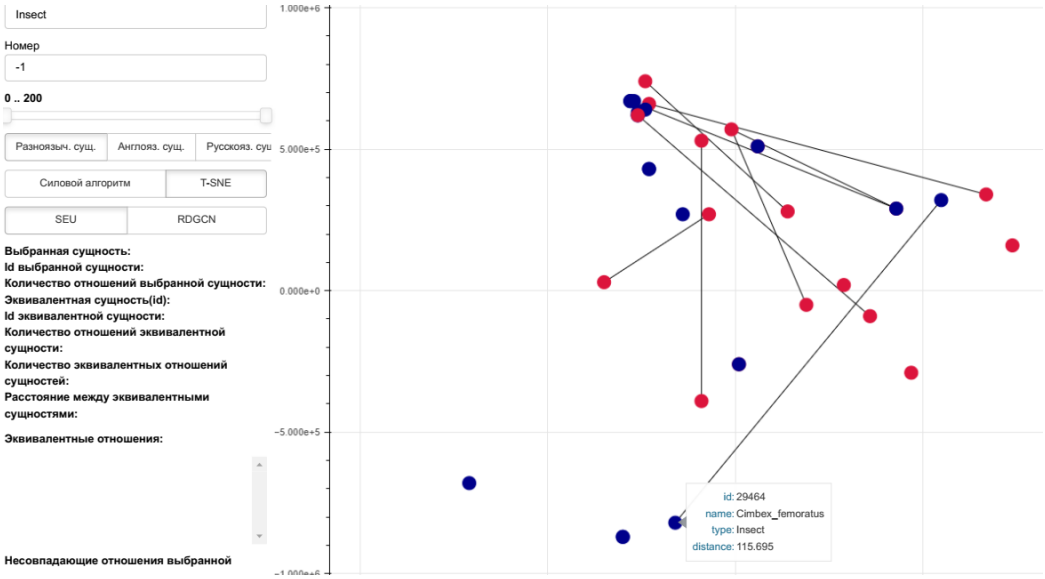


Рис. 8. Визуализация результатов выравнивания сущностей типа Insect методом SEU.

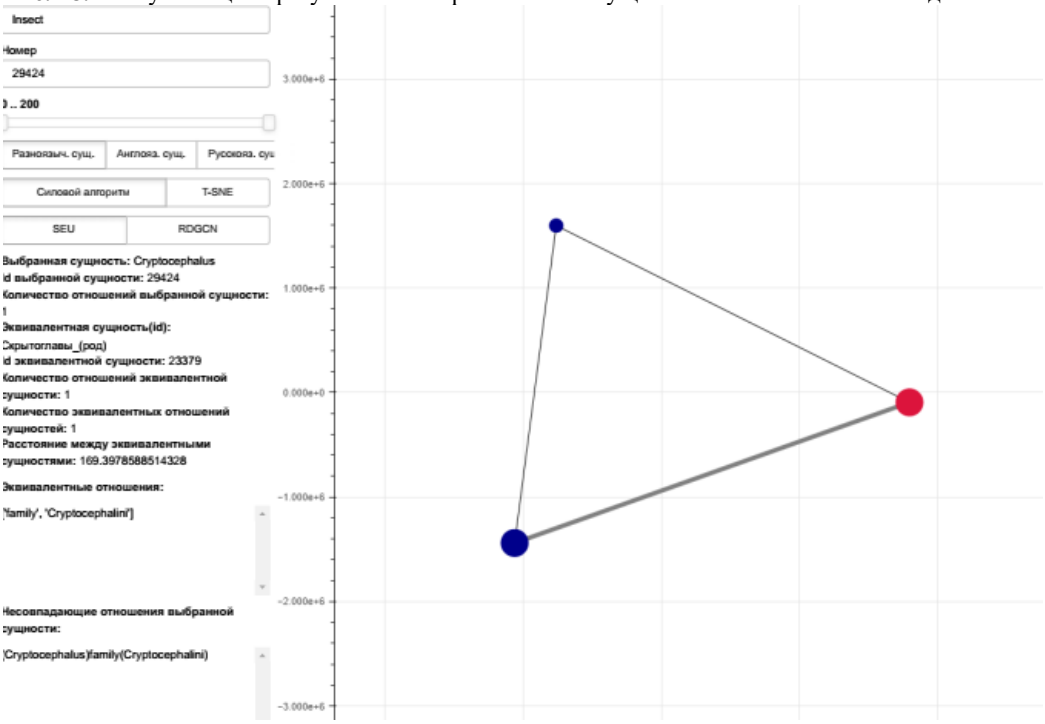


Рис. 9. Отношения эквивалентных сущностей Id29424 *Cryptocephalus* и Id23379 *Скрытоглавы_(род)*.

Заключение

В данной работе представлен алгоритм визуализации, сочетающий как обычную визуализация векторных представлений посредством проекции на плоскость, так и графовую визуализацию. Представленный инструмент является продолжением работ по визуализации контента порталов знаний [9, 10]. Реализован широкий набор возможностей для указанных методов визуализации, позволяющий:

- Визуализацию графа или графов знаний в виде обобщенного графа типов;
- Визуализацию сущностей и отношений выбранного класса;
- Возможность выбора алгоритма выравнивания сущностей для визуализации;
- Возможность переключаться между режимами графовой визуализации и визуализации векторного представления графа знаний;

- Возможность визуализации связей между отдельными сущностями;
- Возможность визуализации сущностей, оказавшихся в результате работы алгоритма выравнивания на заданном расстоянии.

Набор опций визуализации упрощает понимание не только результатов работы различных алгоритмов выравнивания сущностей, но и структуры тестового набора данных.

Литература

[1] Sun Z., Zhang Q., Hu W., Wang C., Chen M., Akrami F. et al. A benchmarking study of embedding-based entity alignment for knowledge graphs // Proc. VLDB Endowment. 2020. Vol. 13. P. 2326–2340.

[2] Rui Zhang, Bayu Distiawan, Trisedy Miao Li, Yong Jiang, Jianzhong Qi A Comprehensive Survey on Knowledge Graph Entity Alignment via Representation Learning.// https://www.researchgate.net/publication/350484214_A_Comprehensive_Survey_on_Knowledge_Graph_Entity_Alignment_via_Representation_Learning.

[3] Gnezdilova V.A., Apanovich Z.V., Russian-English dataset and comparative analysis of algorithms for cross-language embedding-based entity alignment // Journal of Physics: Conference Series. 2021. Vol. 2099.

[4] Gusev D., Apanovich Z. Methods of processing textual information in entity alignment algorithms, Bulletin of the Novosibirsk Computing Center. Series: Computer Science. 2021. № 45. С. 49-58.

[5] Mao X., Wang W., Wu Y., Lan M. From alignment to assignment: frustratingly simple unsupervised entity alignment // Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021. P. 2843–2853.

[6] Wu Y., Liu X., Feng Y., Wang Z., Yan R., Zhao D. Relation-aware entity alignment for heterogeneous knowledge graphs // Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019. P. 5278–5284.

[7] Maaten L. van der, Hinton G. Visualizing data using t-SNE // Journal of Machine LearningResearch. 2008. Vol. 86. P. 2579–2605.

[8] Jacomy M., Venturini T., Heymann S., Bastian M., ForceAtlas2, a Continuous Graph Layout Algorithm for Handy NetworkVisualization Designed for the Gephi Software// Plos One https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0098679

[9] Апанович З.В., Кислицина Т.А., Расширение подсистемы визуализации наполнения информационного портала средствами визуальной аналитики. // Проблемы управления и моделирования в сложных системах. Труды XII Международной конференции. Российская академия наук, Самарский научный центр, Институт проблем управления сложными системами; редколлегия: Е.А. Федосов, Н.А. Кузнецов, В.А. Виттих. 2010. С. 518-525.

[10] Apanovich Z.V., Vinokurov P.S., Elagin V., An approach to visualization of knowledge portal content//Bulletin of the Novosibirsk Computing Center. Series: Computer Science. 2009. № 29. С. 17-32.

Классификация типовых моделей рассуждения и их применение для исследования аргументации в текстах научной коммуникации

Кононенко И.С.[1], Загорулько Ю.А.[1], Серый А.С.[1], Сидорова Е.А.[1], Шестаков В.К.[1]

[1]*Институт систем информатики им. А.П. Ершова СО РАН, пр. Академика Лаврентьева, д. 6, г. Новосибирск, 630090, Россия.*

irina_k@cn.ru, zagor@iis.nsk.su, alexey.seryj@iis.nsk.su, lena@iis.nsk.su, shestakov@iis.nsk.su

Аннотация. *Работа посвящена вопросам исследования аргументации в текстах научной коммуникации. Рассматривается типовая структура рассуждения, фиксируемая с помощью системы связанных утверждений, и предлагается классификация типовых моделей рассуждения по различным основаниям. Предложены четыре основания классификации: по типу заключения, основанию убедительности, семантическому отношению и направленности атаки. Описаны особенности применения онтологического подхода при разработке веб-платформы, предназначенной для исследования аргументации и аргументационного аннотирования корпусов текстов. Обсуждаются особенности платформы, связанные с представлением схем аргументации и их использованием аннотаторами и исследователями. На основе проведенных исследований предложены новые типовые модели рассуждения, характерные для текстов научной коммуникации, и приведены их формальные описания.*

Ключевые слова: анализ аргументации, схема аргументации, онтология аргументации, научная коммуникация, классификация аргументов

1 Введение

Аргументация играет важную роль в сфере научной коммуникации. Она служит как для обоснования выдвигаемых тезисов и гипотез, с целью сделать их более убедительными для аудитории, так и для опровержения высказанных кем-то тезисов и гипотез. При оценке статей важно не только наличие в них научных результатов и выводов, но и то, как эти результаты и выводы обосновываются, какие при этом используются стратегии аргументации.

При анализе аргументации, используемой в текстах, когда требуется извлекать элементы аргументов из фрагментов текста на естественном языке, структура аргумента играет важную роль. Поэтому при решении таких задач используются модели структурированной аргументации. Наиболее известной моделью структурированной аргументации, нашедшей применение в практических системах анализа аргументации, является модель Дугласа Уолтона [1]. В этой модели структурированный аргумент определяется как набор утверждений, состоящий из трёх частей: набора посылок, заключения и вывода из посылок к заключению.

Схемы аргументации в модели Д. Уолтона – это формы аргументов, описывающие структуры аргументативного вывода [2]. Это понятие позволяет выявлять и оценивать общие типы аргументации. Такие схемы могут использоваться для представления знаний, необходимых для приведения аргументов и объяснений. Они отражают стереотипные модели мышления, которые в общем случае не являются дедуктивными и монотонными. Поэтому их

также называют моделями правдоподобного вывода, в том смысле, что если посылки аргумента верны, то предположительно верно и заключение.

Каждая схема имеет название, заключение, набор посылок и набор критических вопросов. Критические вопросы дают возможность определить слабые стороны аргумента, основанного на этой схеме. Их можно использовать для моделирования дополнительных посылок или порождения не представленных явно в тексте или опущенных в рассуждении посылок, которые будут поддерживать аргумент. Схемы аргументации используются в ряде инструментов для анализа аргументации (Araucaria [3], OVA [4], Carneades [5], ArgDF [6]).

Анализ аргументации, предполагающий обнаружение аргументов в текстах, является довольно сложным и трудоемким процессом. В связи с этим актуальной представляется задача автоматизации этого процесса. Для обучения методов автоматического извлечения аргументации требуется создать корпусы текстов, содержащих разметку аргументации. Поскольку типовых схем аргументации довольно много, необходимо так их представить и классифицировать, чтобы максимально упростить разметчикам (аналитикам, экспертам) работу по обнаружению фрагментов текстов, содержащих аргументы, и выбору подходящей для них схемы.

В данной работе рассматривается подход к построению классификаций схем аргументации по различным основаниям, а также использование таких классификаций в веб-платформе, предназначенной для исследования аргументации и аргументационного аннотирования корпусов текстов.

2 Анализ моделей рассуждения

Схемы аргументации являются, согласно Д. Уолтону и К. Риду [2], наиболее полезным и широко признанным инструментом, разработанным к настоящему времени в теории аргументации. Рис.1 демонстрирует одну из схем, обеспечивающих основу для построения аргумента в рамках определенной логической структуры вывода, описывающей переход от посылки к заключению (тезису) и представляющей этот переход как правдоподобный (что находит отражение в схемах презумптивных рассуждений, Presumptive Inference).

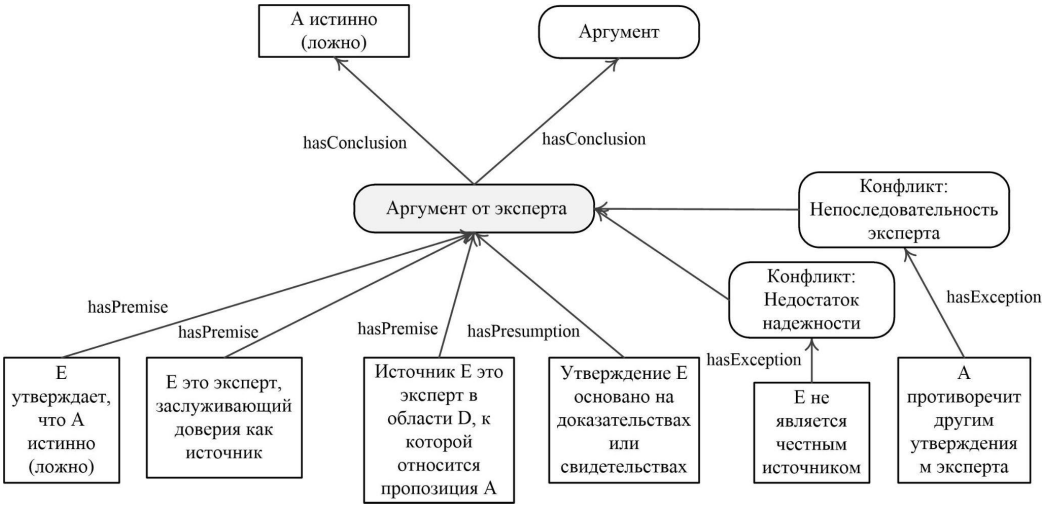


Рис.1. Пример схемы аргументации.

Таким образом схема представляет структуру и рекомендации по построению и оценке аргументов в рамках определенного шаблона. Различные ученые и исследователи могут интерпретировать и концептуализировать схемы аргументации немного по-разному, использовать несколько отличающуюся терминологию или классификации, поэтому могут быть различия и в том, как эти схемы называются или описываются, но основная цель состоит в том, чтобы обеспечить систематическое понимание того, как аргументы структурируются и оцениваются в рамках определенных моделей рассуждений. Компендиум Д. Уолтона, К. Рида и Ф. Макагно [7] – первая книга, в которой собрано воедино и проанализировано наибольшее количество схем, служащих основой для исследования и оценки различных типов аргументов.

Используемая в рамках платформы онтология опирается на компендиум Уолтона, как наиболее представительное и продвинутое в плане классификации и формализации описание

схем аргументации. Исходная онтология аргументации [8] была модифицирована нами в двух направлениях: введена классификация схем на основе многоаспектного подхода и добавлены схемы, необходимые для анализа научной коммуникации.

2.1 Схемы аргументации: от содержательного описания к формальному определению

Распространенные способы рассуждения и убеждения аудитории оформляются в виде схем, содержащих одну или несколько посылок и вытекающее из них заключение. Ниже представлены примеры содержательного описания популярных схем рассуждения, большинство из которых характерны для научной коммуникации.

1. Аргумент от примера: аргументация от отдельных случаев или примеров к общему утверждению.
2. Аргумент от знака: утверждение о том, что знак или индикатор является свидетельством в пользу определенного заключения.
3. Аргумент от причины к следствию: утверждение о том, что причина приведет к определенному следствию.
4. Аргумент от следствия к причине: утверждение о том, что наблюдаемое следствие подразумевает конкретную причину.
5. Аргумент от общего: утверждение о том, что обобщение применимо к конкретному случаю.
6. Аргумент от аналогии: утверждение о том, что два или более случая или ситуации похожи, и то, что верно для одного, должно быть верно и для других.
7. Аргумент от позитивных последствий: аргумент о том, что следует предпринять определенные действия, поскольку оно влечет позитивные последствия.
8. Аргумент от негативных последствий: аргумент о том, что не следует предпринимать определенных действий, чтобы избежать негативных последствий.
9. Аргумент от классификации: утверждение, что что-то принадлежит к определенной категории на основе его определяющих характеристик.
10. Аргумент от свидетеля: аргумент о том, что утверждение верно, потому что оно подтверждается свидетельством или заявлениями других.
11. Аргумент от эксперта: аргумент о том, что утверждение верно, поскольку оно подтверждается мнением эксперта в соответствующей области.
12. Аргумент от общего мнения: аргумент о том, что утверждение действительно, потому что ему следуют многие.
13. Аргумент от общепринятой практики: утверждение о том, что практика действительна, потому что она принята многими.
14. Аргумент от практического рассуждения: аргумент о том, что следует предпринять определенное действие, поскольку оно является способом достижения цели.
15. Аргумент от напрасных затрат: аргумент о том, что следует продолжать добиваться цели путем определенных действий, чтобы избежать негативного состояния, связанного с напрасными усилиями.

В исходной онтологии такие рассуждения представлены формально. Так, формальное представление схемы 1 выглядит следующим образом:

«***Аргумент от примера***»

Посылка_о_Свойстве_в_Этом_Случае: В этом случае индивид a имеет свойство F, а также свойство G

Посылка_о_Типичном_Объекте: a – типичный представитель вещей, имеющих свойство F и имеющих или не имеющих свойство G

Заключение: Как правило, если x имеет свойство F, то (обычно, вероятно, типично) x также имеет свойство G

Очевидно, что такое рассуждение распространено очень широко в различных типах дискурса. Так, на Рис. 2 приведен пример аргумента от примера в тексте научной статьи.

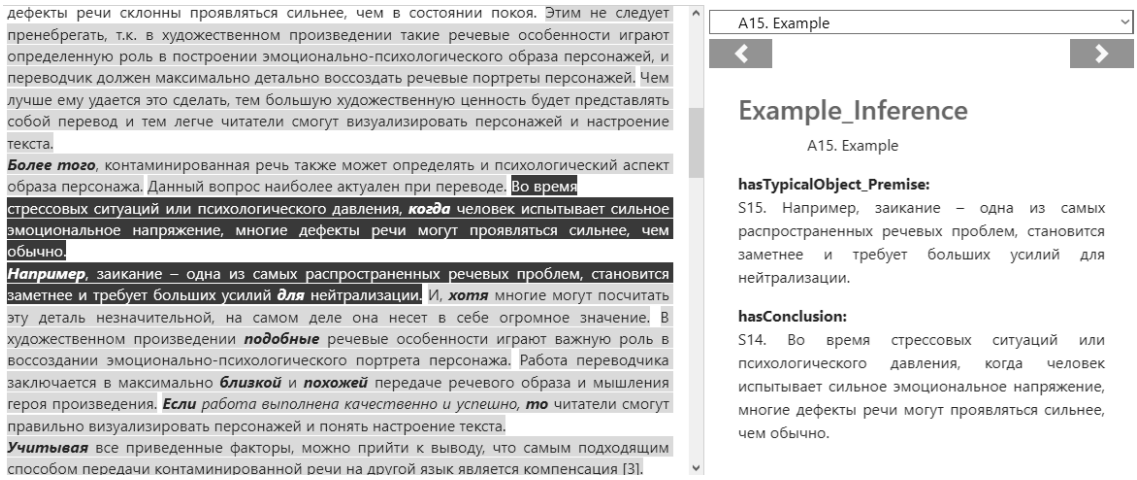


Рис.2. Пример рассуждения по схеме «Аргумент от примера».

Еще один пример (Рис.3) формального представления схемы – для аргумента 14.

«***Практическое рассуждение***»
Посылка_о_Цели: цель – осуществить A
Посылка_о_Способе: достижение B – это способ добиться A
Заключение: B должно быть реализовано

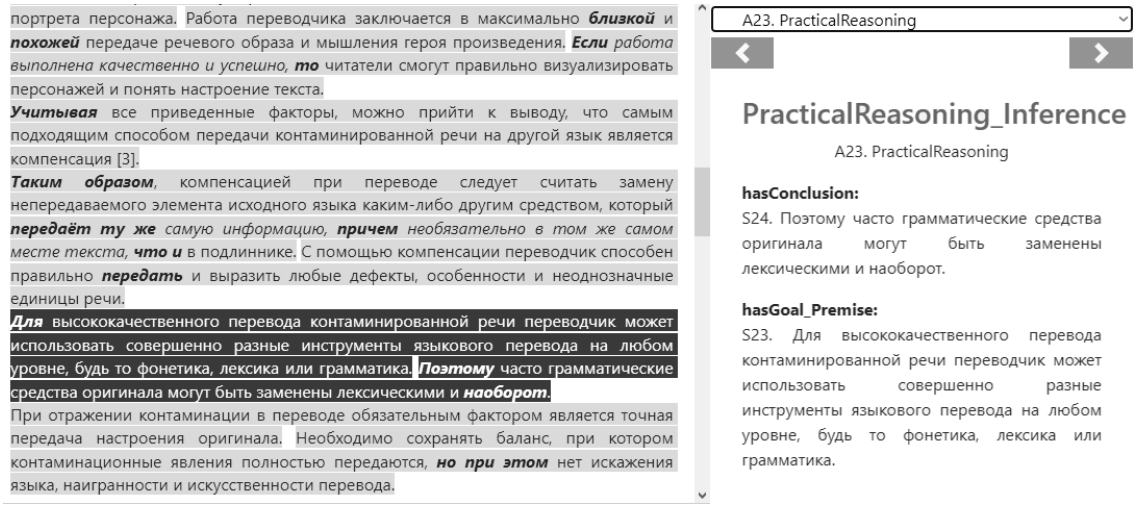


Рис.3. Пример рассуждения по схеме «Практическое рассуждение».

2.2 Классификация схем аргументации

К сожалению, исходное представление всей совокупности схем аргументации в онтологии носит недостаточно структурированный характер: схемы представлены почти плоским списком (с небольшими вкраплениями иерархизации), что, учитывая немалое количество схем (всего 64 схемы), делает их множество трудно обозримым для пользователя, прежде всего, для аннотатора (а также для исследователя). В то же время при описании тех или иных схем аргументации аналитики группируют их, исходя из различных семантических и прагматических критериев. Так в [7] обсуждаются следующие группы:

- каузальные схемы (От причины к следствию, От следствия к причине, От корреляции с причиной),
- схемы аргументации от знания (От эксперта, От знающего, От свидетеля),
- практические схемы (Практическое рассуждение, От негативных последствий, От позитивных последствий).

К этим группам можно добавить еще одну:

- схемы рассуждения через детализацию (От примера, От общего) [9].

Такого рода группировки схем называют функциональными. Однако, с одной стороны, очевидно, что группы выделены на разных основаниях, с другой стороны, не все схемы поддаются такой односторонней классификации. Учитывая изложенные выше соображения, было принято решение представить схемы аргументации с помощью многоаспектной, фасетной классификации.

2.2.1 Тип заключения

Анализ множества схем исходной онтологии показывает, что практическим схемам, нацеленным на убеждение аудитории осуществить некую деятельность, как заключение схемы 14 (*B должно быть реализовано*), либо отказаться от нее, можно противопоставить теоретические, апеллирующие к картине мира аудитории и утверждающие истинность некоторого тезиса, как заключение схемы 1 (*Как правило, если x имеет свойство F, то … x также имеет свойство G*).

Таблица 1. Классификация на основе типа заключения.

Тип заключения			
Практические аргументы	Q_P	**Теоретические аргументы**	Q_T
К_действию	16	Устройство_реальности	8
К_цели	1	Установление_реальности	17
К_обязательству, вовлеченности	2		

Основанием деления схем и описываемых ими аргументов на практические и теоретические можно считать ‘Тип заключения’ (см. Таблицу 1, в таблицах Q обозначает количество схем, которым соответствует данное значение параметра классификации). Сами ‘Практические аргументы’ разнятся, в зависимости от того, к какому аспекту деятельности они апеллируют: ‘к действию’ (От Негативных последствий), ‘к цели’ (От Напрасных трат), ‘к обязательству’ (От Обязательства).

‘Теоретические аргументы’ далее подразделяются на две группы, в зависимости от того, направлен ли аргумент на изменение представлений аудитории о реальности, ‘Установление_реальности’ (От Аналогии, От Причины к Следствию, От Следствия к Причине, От Корреляции с Причиной, От Эксперта, От Знающего, От Свидетеля, Доказательство Гипотезы, Фальсификация Гипотезы, От Незнания и др.), или нет, ‘Устройство_реальности’ (От Знака, От Части к Целому, от Группы к Членам, От Членов к Группе, От Причины к Следствию, От Общего, От Классификации).

2.2.2 Основание убедительности

Классификация схем по основанию убедительности учитывает зависимость аргумента от его источника или внутренней структуры (см. Таблицу 2).

Таблица 2. Классификация по типу основания убедительности.

Основание убедительности			
Внутреннее (предмет)	Q_I	Внешнее (источник)	Q_O
От_определения	2	От_человека	8
От_классификации	6	От_знания	6
От_факта	13	От_ценности	6
От_объяснения	15		
От_цели	2		

Аргументация может строиться на предметной основе: для убеждения аудитории используются факты, объяснения, определения, классификации, относящиеся к предметной области. Например, в «Аргументе от примера» в качестве ‘Основания убедительности’ выступает факт, связанный с частным случаем, позволяющий сделать обобщение.

С другой стороны, аргумент может опираться на внешнее по отношению к предмету основание: ‘От_человека’, как в «Этическом Аргументе», в котором убеждение основано на характере человека, ‘От_знания’ источника, как в «Аргументе от Эксперта», или присущие коммуникантам ценности, как аргументы, апеллирующие к положительному или отрицательному отношению к последствиям деятельности.

2.2.3 Онтолого-семантические отношения

В основе каждой схемы лежит та или иная онтологическая концепция, относящаяся к природе сущностей или явлений, вовлеченных в аргументативный дискурс, и соответствующий семантический концепт, относящийся к отношениям между элементами аргумента. Эти концепции и отношения обеспечивают основу для понимания и анализа аргументов в рамках каждой схемы. Рассмотрим некоторые типы отношений.

a) В «Аргументе от примера» онтологическая концепция вращается вокруг случаев или примеров, представляющих более широкую категорию, или обобщение. Семантический концепт включает в себя выявление и выделение общих характеристик или паттернов среди примеров для поддержки обобщающего вывода (‘Гипер-гипонимия’ или ‘Элемент-множество’).
b) В «Аргументе от причины к следствию» онтологическая концепция сосредоточена вокруг причинно-следственных связей (‘Каузальность’). Семантический концепт фокусируется на установлении причинно-следственной связи между выявленной причиной и прогнозируемым следствием, часто с помощью эмпирических данных или логических рассуждений.
c) В «Аргументе от аналогии» онтологическая концепция включает в себя идентификацию двух или более аналогичных случаев или ситуаций. Семантический концепт соотносится с оценкой релевантности и сходства значений общих свойств между аналогичными случаями и использованием этого сходства для выводов или предсказаний (‘Аналогия’).

В таблице 3 представлены используемые на текущий момент отношения и количество схем, соотнесенных с каждым из отношений (QS).

Таблица 3. Онтолого-семантические отношения.

***Онтолого-семантические отношения* (Q_S)**			
Каузальность	8	Противоречие	3
Меронимия	3	Способ	2
Гипер-гипонимия	5	Коммуникация	5
Элемент-множество	3	Корреляция	2
Аналогия	4	Кондициональность	7

Представленные онтолого-семантические отношения не исчерпывают списка возможных отношений, которые обеспечивают концептуальную основу для схем аргументации и определяют построение и оценку аргументов в рамках этих схем, помогают понять природу сущностей, отношений и рассуждений, связанных с конкретным контекстом аргументации. Важно отметить, что эти отношения могут варьироваться в зависимости от конкретной схемы аргументации и используемой теоретической основы.

2.2.4 Направление атаки

Существуют схемы, описывающие рассуждения, фактически представляющие собой контраргументы, то есть отрицающие справедливость тезиса (атака на тезис, как в «Аргументе от Незнания», в котором участник дискурса не принимает утверждение, опираясь на мудрость “если бы это было так, все бы это знали”, или как в «Аргументе от Негативных последствий», в котором предлагается не совершать действие, поскольку оно имеет плохие последствия), аргумента в целом (атака на аргумент, как в «Этическом аргументе», в котором участник дискурса “переходит на личности”, считая, что аргумент его оппонента не_может/может быть принят, поскольку тот плохой/хороший человек). В последнем примере одновременно осуществляется атака на аргумент и его источник – таким образом, схеме может соответствовать несколько значений одного и того же параметра, в данном случае параметра ‘Направление атаки’.

Таблица 4. Направление атаки.

Направление атаки (Presumptive Inference)	Q_A	***Направление атаки (Conflict Inference)***	Q_A
На_тезис	11	На_тезис	1
На_аргумент	4	На_аргумент	19
На_источник	5		

Выделенная в исходной онтологии группа конфликтных схем почти вся представляет собой различные виды типичных атак на конкретные, прописанные в схемах, аргументы (см. Таблица 4). Так, конфликт «Альтернативный способ» атакует аргумент «Практическое

рассуждение», утверждая, что имеются альтернативные средства достижения цели. Исключением в группе конфликтов является лишь «Логический конфликт», представляющий собой логическое отрицание тезиса. Таким образом здесь различия представляются противопоставлениями ‘атака’+ ‘на аргумент’ vs. ‘на тезис’.

2.2.5 Многоаспектность

Описанные четыре параметра и множество их значений позволяют представить семантические и прагматические различия аргументов и выстроить их многоаспектную иерархию (некоторые примеры приведены в Таблице 5).

Таблица 5. Примеры фасетной классификации схем аргументации.

	Аргумент от примера	***Практическое рассуждение***	***От негативных последствий***
1. Тип заключения	Установление_реальности	К_действию	К_действию
2. Основание убедительности	От_факта	a. От_объяснения b. От_цели	a. От_объяснения b. От_ценности
3. Семантическое отношение	a. Элемент-множество b. Гипер-гипонимия	Способ	Каузальность
4. Направление атаки			На_тезис

Такая классификация представляется более емкой и гибкой, чем простая иерархия, поскольку не смешивает различные основания классификации, а также позволяет при необходимости вводить новые параметры.

2.3 Дополнительные схемы для научной коммуникации

Экспертный анализ корпуса текстов научной коммуникации, собранный нами и размещенный в открытом доступе на платформе ArgNetBank Studio (uniserv.iis.nsk.su/arg), показал, что во множестве схем, представленных в исходной онтологии, отсутствует ряд схем, необходимых для адекватного аргументационного аннотирования научных текстов. Недостающие схемы были сформулированы, формально описаны, классифицированы и добавлены в онтологию.

1. Были недостаточно представлены аргументы функциональной группы ‘Рассуждение через детализацию’. В связи с этим добавлены схемы «От общего», «От частного», «От частей к целому», «От группы к членам», «От членов к группе». Соответствующие рассуждения базируются на значениях параметров ‘Устройство_реальности’ и ‘От_классификации’. Что же касается основного онтолого-семантического отношения, описывающего тип детализации, то для первых трех схем это ‘Гипер-гипонимия’ и ‘Меронимия’, а для двух последних это ‘Элемент-множество’. Рассмотрим в качестве примера последнюю схему.

«***От членов к группе***»
Посылка_о_Члене: А является членом (идеей, привычкой, обычаем, продуктом, методом) группы В.
Посылка_о_Свойстве: А имеет свойство G
Заключение: В имеет свойство G
Классификация: 1. Установление_реальности, 2. От_классификации, 3. Элемент-множество.

2. Были неполно представлены схемы функциональной группы причинно-следственных переходов. Добавлена абдуктивная схема «От следствия к причине» (обратная имеющейся схеме «От причины к следствию»). Это теоретический аргумент, апеллирующий к соображению, что если имеет место следствие, то, вероятно, имеет место и причина.

«***От следствия к причине***»
Посылка_о_Причинно-следственной_Связи: А является причиной В
Посылка_о_Событии: В этом случае В имеет место
Заключение: В этом случае предположительно А имело место
Классификация: 1. Установление_реальности, 2. От_объяснения, 3. Каузальность.

3. Теоретическая схема «От наилучшего объяснения» утверждает, что, при наличии нескольких объяснений некоторых фактов, одно из них является наилучшим и должно быть принято как наиболее правдоподобный тезис.

«***От наилучшего объяснения***»

Посылка_о_Предполагаемых_Фактах: В этом случае D это набор данных или предполагаемых фактов

Посылка_об_Объяснении: Каждое из A1…An объясняет D

Посылка_о_Лучшем_Объяснении : A1 лучше всего объясняет D

Заключение: Следовательно, A1 это наиболее правдоподобная гипотеза

Классификация: 1. Установление_реальности, 2. От_объяснения, 3. От_факта.

4. Теоретическая схема «От свидетеля», утверждающая, что некий тезис, представленный заявлением свидетеля, может быть принят как правдоподобный, в предположении что свидетель может знать правду и говорит правду.

«***От свидетеля***»

Посылка_о_Знании: E утверждает, что A истинно (ложно)

Посылка_о_Свидетельских_Показаниях: Свидетель E может знать, истинно ли A в определённой области D, содержащей пропозицию A.

Посылка_о_Правдивости: Свидетель E говорит правду (как он её знает)

Заключение: A истинно (ложно)

Классификация: 1. Установление_реальности, 2. От_знания, 3. Коммуникация.

5. Наконец, еще одна схема, «От метода», свойственна аргументации научных текстов: правдоподобие утверждения основано на том, что оно получено путем применения метода, характерного для данной предметной области.

«***От метода***»

Посылка_о_Методе: Метод T применяется в области D

Посылка_о_Результате: A получено методом T

Посылка_об_Области: A относится к D

Заключение: A истинно (ложно)

Классификация: 1. Установление_реальности, 2. От_знания, 3. Способ.

3 Платформа для исследования аргументации

Разрабатываемая авторами веб-платформа [10] обеспечивает практическую реализацию представленной в данной работе классификации схем аргументации. Платформа предоставляет экспертам инструменты для создания текстовых корпусов с аргументационной разметкой и поддержки исследований аргументации (поисковые сервисы, создание датасетов, исследование индикаторов аргументации). Архитектура данной платформы включает три основных уровня: уровень данных, серверный уровень и уровень клиента (см. Рис. 4.).

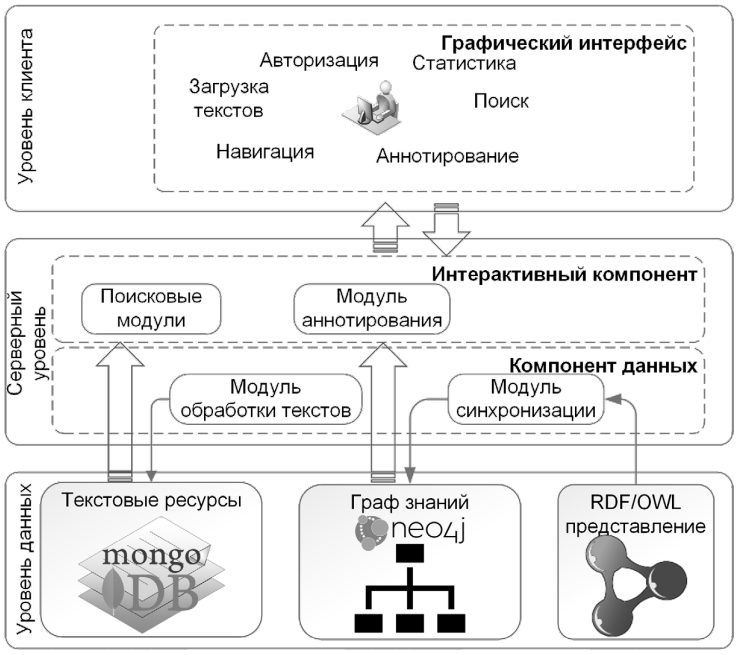


Рис.4. Архитектура веб-платформы для исследования аргументации.

Ядром информационной модели на уровне данных является онтологическая модель, разработанная на основе онтологии аргументации [8]. Множество аннотаций, представляющих аргументационную разметку текстов, представляется графом знаний, который хранится в виде помеченного ориентированного графа (Labeled Property Graph, LPG) в графовой СУБД Neo4j. Neo4j обладает хорошей производительностью, предоставляет удобный интерфейс администратора, простой и мощный язык запросов Cypher и, в большинстве случаев, превосходит [11] в быстродействии традиционные хранилища триплетов. Помимо этого, Neo4j предоставляет средства для работы с онтологиями и графами знаний, реализованные в свободно распространяемом дополнении Neo4j RDF & Semantics Toolkit. Данная функциональность позволяет сохранять взаимно-однозначное соответствие между LPG и OWL-представлениями логического ядра информационной модели.

Основной частью логического ядра является описание схем аргументации. На данный момент на веб-платформе отсутствуют средства, дающие возможность вносить изменения в онтологию. Таким образом, создание новых и доработка существующих схем должна проводиться на уровне OWL-представления с последующей трансляцией изменений в LPG. Как правило, эксперты в области теории аргументации и разметки текстов не являются инженерами знаний и не способны работать с онтологиями в формате OWL. Разработанные в рамках данной работы классификация и типовые паттерны призваны, в том числе, облегчить работу по дальнейшей разработке онтологии.

На рис. 5 показано, как структура онтологической модели влияет на внешний вид и функциональность платформы на примере схемы “От части к целому” (PartToWhole).

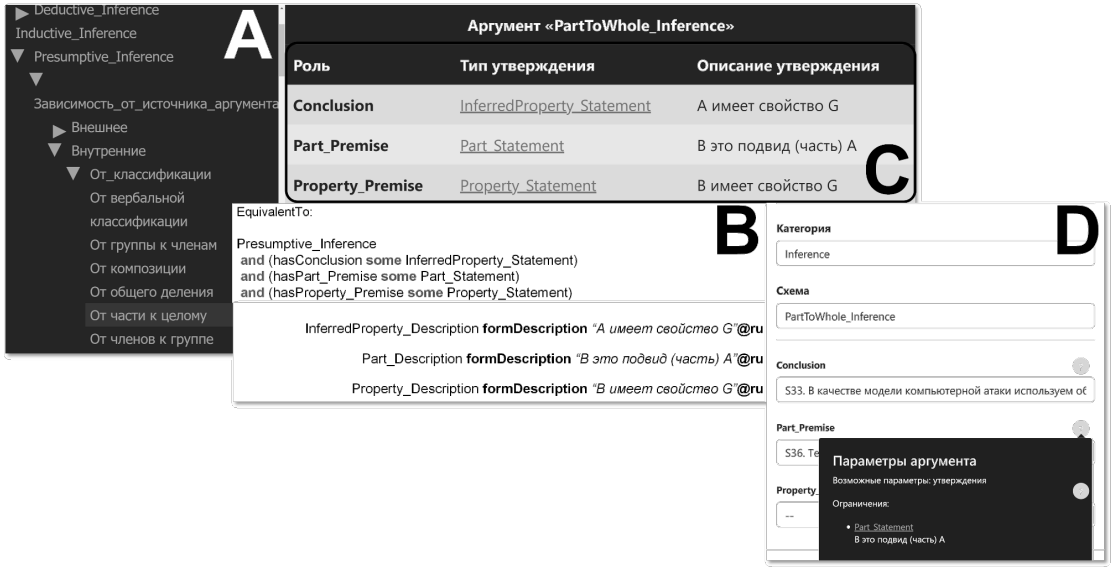


Рис. 5. Пример реализации онтологической модели.

В части A изображения показано дерево иерархии схем, структура которого соответствует классификатору из п.1.2 данной статьи. В части B приведен фрагмент логического описания схемы, дающий информацию о количестве и типах параметров схемы, их ролях и их краткие описания на русском языке. Имея информацию о схеме, можно перейти к описаниям каждого из параметров, чтобы увидеть все схемы, в которых они могут применяться (часть C). В части D показан фрагмент окна редактирования аргумента, удовлетворяющего заданной схеме. Структура схемы используется при отображении формы и всплывающих подсказок, дающих дополнительную информацию о назначении параметров.

Таким образом, как можно видеть из Рис. 5, архитектура платформы непосредственно связана со структурой онтологической модели на уровне графического интерфейса, а также через поисковый модуль и модуль аннотирования на серверном уровне.

4 Заключение

В данной работе проводится исследование типологии моделей рассуждения, представленных классами (схемами) онтологии аргументации. Предложен многоаспектный

подход к типизации схем аргументации для аргументационного аннотирования корпусов текстов. Преимущество многоаспектной классификации по сравнению с простой иерархией состоит в возможности независимого анализа различных прагматических и семантических аспектов описываемых схемами аргументации рассуждений. Кроме того, такая классификация позволяет при необходимости (например, при исследовании новых текстовых жанров) добавлять новые классификационные параметры и/или значения, вводить недостающие схемы и классы схем, необходимые для представления аргументативного содержания текстов, относящихся к исследуемым жанрам.

В качестве развития предложенного подхода предполагается анализ применимости и достаточности имеющегося набора схем к малоизученным субжанрам научной коммуникации (таким, как рецензии, аналитические статьи и комментарии к ним). С технической точки зрения онтология играет роль базы знаний информационно-исследовательской системы. На основе онтологии реализуется графовая БД системы и поисковый сервис, обеспечивающий выдачу всех аргументов заданного класса, осуществляется аргументационная разметка текста и моделирование структуры рассуждений в виде связного ориентированного графа.

В рамках развития платформы планируется интеграция редактора схем аргументации непосредственно в графический интерфейс конечного пользователя, что позволит экспертам работать с онтологией без необходимости погружаться в технические детали ее реализации.

5 Благодарности

Работа выполнена при финансовой поддержке Российского научного фонда (проект № 23-11-00261).

Литература

[1] Walton D. Argumentation theory: A very short introduction // G. Simari, I. Rahwan (eds.) Argumentation in Artificial Intelligence. – Springer, Boston; 2009: 1–22.

[2] Reed C., Walton D. Argumentation schemes in argument-as-process and argument-as-product // Proc. of Conference Celebrating Informal Logic. OSSA Conference Archive 75, University of Windsor. – OSSA, Windsor, Ontario, 2003. (1 CD-ROM).

[3] Reed C., Rowe G. Araucaria: Software for argument analysis, diagramming and representation // International Journal on Artificial Intelligence Tools 2004, 13(4): 961–979.

[4] Janier M., Lawrence J., Reed C. OVA+: An argument analysis interface // Computational Models of Argument: Proc. of COMMA 2014. - IOS Press, Amsterdam; 2014, 266: 463–464.

[5] Gordon T.F., Walton D. The Carneades argumentation framework - using presumptions and exceptions to model critical questions // Proc. of the 2006 Conf. on Computational Models of Argument: Proceedings of COMMA 2006. – IOS Press, Amsterdam; 2006, 6: 195–207.

[6] Rahwan I., Zablith F., Reed C. Laying the foundations for a world wide argument web // Artificial intelligence 2007, 171(10–15): 897–921.

[7] Walton D. Argumentation schemes / D. Walton, C. Reed, F. Macagno. – Cambridge: Cambridge University Press, 2008. – 456 p.

[8] Rahwan I. Representing and classifying arguments on the semantic web / I. Rahwan, B. Banihashemi, C. Reed, D. Walton, S. Abdallah // The Knowledge Engineering Review. – 2011. – Vol. 26(4). – P. 487–511

[9] Пименов И.С. Сочетаемость аргументов разных функциональных групп в научных текстах // Филологические науки. Вопросы теории и практики. Тамбов: Грамота, 2022, Т. 11. С.3672–3680.

[10] Сидорова Е.А., Ахмадеева И.Р., Загорулько Ю.А., Серый А.С., Шестаков В.К. Платформа для исследования аргументации в научно-популярном дискурсе // Онтология проектирования. – 2020. – Т. 10, №4(38). - С. 489–502. – DOI: 10.18287/2223-9537-2020-10-4-489-502.

[11] Grinevich A., Sery A.. Analyzing the Cultural Universals of the Folklore of Peoples of Siberia and the Far East // Proceedings of the XXIII International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2021). Moscow, Russia, October 26-29, 2021. CEUR Workshop Proceedings, Vol-3036. 2021. pp. 387-401.

Анализ и Распознавание Транспортных Шумов с Использованием Нейросетевого Подхода

Копылова О.А.[1], Хайретдинов М.С.[1]

[1] *Институт вычислительной математики и математической геофизики СО РАН, пр. акад. Лаврентьева, д. 6, г. Новосибирск, 630090, Россия.*

oksana@opg.sscc.ru, marat@opg.sscc.ru

Аннотация. *В работе рассматривается задача распознавания движущихся транспортных средств по записям сейсмических колебаний в условиях реальной фоновой обстановки. Целью работы является разработка методов и алгоритмов распознавания, предназначенных для использования в сейсмоакустических системах наблюдения в режиме реального времени. Рассматриваются различные виды железнодорожного, тяжелого колесного, гусеничного транспорта. Решение задачи основано на использовании сверточной нейронной сети с учетом спектральных характеристик транспортных колебаний. Отдельно рассматриваются вопросы распознавания в условиях меняющегося уровня внешних шумов и пространственного положения транспорта по отношению к пунктам регистрации колебаний. Приводятся результаты работы алгоритма распознавания на данных полевых экспериментов.*

Ключевые слова: распознавание транспортных вибраций, сверточная нейронная сеть, сейсмические и акустические колебания, полевой эксперимент

1 Введение

Одной из основных задач геоэкологического мониторинга окружающей среды является выявление источников загрязняющего шума в условиях города. В такой постановке решение задачи связано с задачей распознавания источников транспортных колебаний, возникающих в земле и атмосфере. С целью формирования информативной системы признаков в работе приводятся результаты спектрального-временного анализа колебаний от тяжелых видов транспорта – железнодорожного, тяжелого колесного и гусеничного видов транспорта. Такие колебания распространяются в виде сейсмических в земле и акустических в атмосфере.

Решение задачи распознавания движущихся транспортных средств по записям сейсмических сигналов может быть построено на сравнении текущего вектора значений информативных признаков для различных источников с их эталонами, полученными статистическими методами [1, 2]. В работе [3] распознавание объектов классов «пешеход» и «автомобиль» проводилось с помощью вероятностной нейронной сети. Авторы статьи [4] предложили решение проблемы распознавания людей, животных и вездеходов по данным сейсмических и инфракрасных датчиков на основе бинарной классификации с использованием метода опорных векторов. В работе [5] рассматривалось распознавание техногенных шумов с использованием нейронной сети прямого распространения и метода k-ближайших соседей. В статье [6] рассматривается проблема распознавания классов «человек», «транспортное средство» и «животное» по данным акустических записей и классов «человек», «транспортное средство» и «группа людей» по данным вибрационных записей с помощью метода опорных векторов (SVM). На вход алгоритма классификации подавались необработанные данные вибрационных колебаний и мел-кепстральные коэффициенты (MFCC) акустических колебаний. В работе [7] для классификации колесных и гусеничных транспортных средств по сейсмическому сигналу предлагается сверточная нейронная сеть (CNN). В качестве информативных признаков

используются частотно-кепстральные коэффициенты в логарифмическом масштабе. В работе [8] авторы предлагают решение проблемы распознавания движущихся пешеходов, колесных и гусеничных транспортных средств по сейсмическим данным на основе блока SE-Inception. Авторы статьи [9] рассматривают задачу классификации объектов классов «колесный трактор» и «автобус» по частотно-временным коэффициентам на основе эмпирического вейвлет-преобразования (EWT) с использованием алгоритмов SVM и KNN. В работе [10] распознавание объектов классов «группа людей», «колесная» и «гусеничная техника» основывалось на сверточной нейронной сети. В работе [11] для распознавания пешеходов, сигналов транспортных средств и шумов окружающей среды по сейсмическим колебаниям также основой служил алгоритм SVM. SVM применялся для распознавания поездов, людей, слонов, оленей, электрического шума по сейсмическим колебаниям в [12]. В статье [13] для распознавания движущихся объектов таких, как беспилотники, гусеничные машины, колесные машины, люди и животные, предложено использовать новый метод CO-SDL (Compression Observation-Seismic DL). Авторами [14] для распознавании колесной и гусеничной техники было предложено использовать метод сжатия CS и одномерную сверточную нейронную сеть 1-D CNN. В работе [15] для распознавания движущихся транспортных средств использовалась сверточная нейронная сеть с двунаправленной долговременной кратковременной памятью (1D CNN-BiLSTM). Для классификации движущегося транспорта категориям «большие», «средние» и «малые» авторы работы [16] использовали логистическую регрессию, метод опорных векторов и наивный байесовский алгоритм. Для распознавания движущихся транспортных средств по данным сейсмических колебаний авторы [17] предлагают свое решение, основанное на использовании сверточной нейронной сети и глубокого двунаправленного LSTM (Long short-term memory) – CNN-DBiLSTM.

Отличительная особенность настоящей работы увязывается с вопросами обнаружения и распознавания на предельных расстояниях, превосходящих известные в условиях воздействия внешних шумов. При этом алгоритмы и программы ориентированы на обработку данных в режиме реального времени.

2 Постановка задачи

2.1 Общая постановка задачи

В качестве критерия распознавания используется вероятность правильного распознавания как функция трех составляющих:

$$P=\psi(\boldsymbol{F}, \boldsymbol{N}, A) \tag{1}$$

где P вероятность правильного распознавания, F – вектор признаков, N – составляющая внешнего шума, A – алгоритм распознавания.

Вектор признаков $\boldsymbol{F}$ формируется из значений спектральных функций $F(x_k,t)$ регистрируемых транспортных колебаний, представляемых в виде модели:

$$x_k(t_i) = A_x(k)h_k L\left[\cos\left(\omega_0(t_i - \triangle t_k) + \varphi_x(k)\right) + n_k(t_i)\right] \tag{2}$$

где ω_0 – преобладающая в спектре колебаний частота вращения двигателя; $\varphi_x(k)$ – фаза ее в k-ой точке регистрации; t_i меняется в пределах от t_{0i} до $t_{0i}+T$, где T – длительность сигнала, а t_{oi} соответствует моменту появления полезного колебания, $n_k(t)$ – широкополосная шумовая составляющая, отражающая взаимодействие источника со средой в процессе его движения, и вклад внешнего шума; A_x – амплитуда колебания на k-ом датчике; h_k – чувствительность датчика; L –оператор фильтрации сигнала.

С учетом вариабельности основных составляющих параметров в (2), обусловленных меняющимися характеристиками режимов движения транспортных средств по скорости и их положения в пространстве, задача распознавания приобретает динамический характер и ее решение распадается на этапы обнаружения транспортных колебаний на фоне внешних шумов, выделения информативных признаков, выбор непосредственно алгоритма распознавания и конечный этап, связанный с оцениванием эффективности работы комплекса алгоритмов обработки на экспериментальных данных. Авторами рассмотрены все этапы решения задачи распознавания.

Предлагаемые авторами алгоритмы обнаружения и результаты оценивания их эффективности по критерию дальности обнаружения по отношению к некоторым видам транспортных средств представлены в [18]. Остановимся на следующем этапе.

2.2 Выделение информативных признаков

Решение задачи выделения информативных признаков осуществляется на основе использования секционированного спектрального анализа на основе дискретного преобразования Фурье, позволяющего проследить динамику изменения спектра во времени, обусловленную меняющимися режимами движения транспортных средств во времени и пространстве:

$$F(k,l) = \sum_{n=0}^{N-1} x_l(t_n) \exp(-i\frac{2\pi mk}{N}), l = 1,...,L \quad (3)$$

Здесь L - количество секций длительностью $\Delta T = N \cdot \Delta t$ каждая, на которые секционируется искомый сигнал $x(t)$, Δt - интервал выборки дискретных значений сигнала.

Следствием меняющегося пространственного положения транспорта является ярко выраженный уровень вариаций регистрируемых транспортных колебаний в системе сейсмоакустического наблюдения. В связи с этим процедура нормировки исходных данных по уровню является необходимой предобработкой для формирования спектральных признаков, инвариантных к расстоянию и уровню внешних шумов. Преобразование исходных данных результата спектрального анализа выполняется так, чтобы все амплитудные значения бинов (отсчетов) лежали в диапазоне [0;1] в соответствие: $x_i = x_i / \max(x)$, где x_i – i-ое значение бина спектральной функции, $\boldsymbol{x}$ – вектор значений бинов спектральной функции.

Для повышения помехоустойчивости спектральных признаков применяется полосовая цифровая фильтрация спектра транспортных колебаний из спектров внешних шумов согласно заданию оператора L в (2). В качестве его выбрана передаточная функция цифрового фильтра Баттерворта 4-го порядка:

$$L(\omega) = \frac{1}{\sqrt{1+\left(\frac{\omega}{\omega_0}\right)^{2n}}}, \quad (4)$$

где ω_0 – граничная частота, n – порядок фильтра.

3 . Нейросетевой алгоритм распознавания транспортных источников

Набор исходных данных для распознавания представляется в виде набора матриц с размером $M \times N$, где M – длительность секции в сек (3), в пределах которого осуществляется анализ, N – количество бинов спектральной функции.

Распределение образцов по классам представлено ниже (Рис. 1). Классы кодировались следующим образом: 0 – электропоезд, 1 – грузовой поезд, 2 – гусеничный транспорт, 3 –тяжелый колесный транспорт.

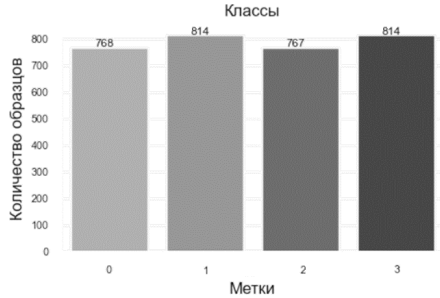
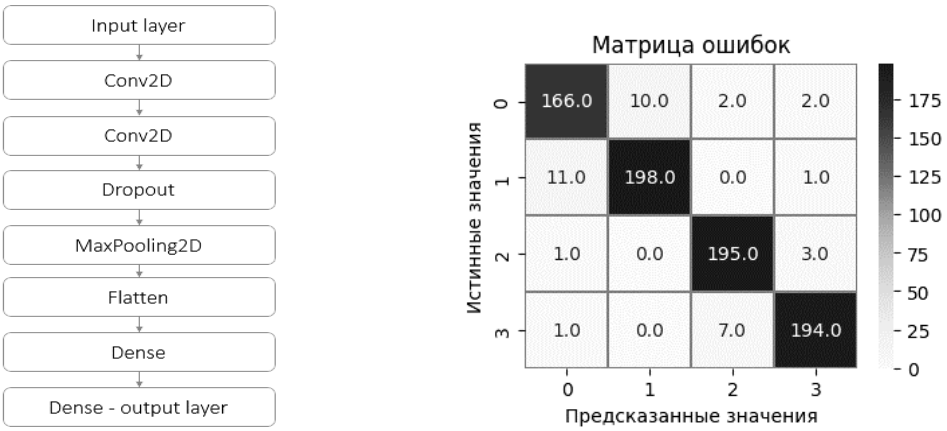


Рис. 1. Распределение набора данных по классам

Полученный набор из экспериментальных данных был перемешан случайным образом и разделен на тренировочную и тестовую выборки в соотношении 3:1. Такое разделение обеспечивает достаточное количество экземпляров в обучающей и тестовой выборках для обучения модели и ее тестирования. Нейронная сеть состоит из двух двумерных сверточных слоев (Conv2D), выпрямляющего слоя (Flatten), слоя подвыборки для снижения размернсти (MaxPooling2D), полносвязного слоя классификатора (Dense) и выходного полносвязного слоя

(Dense – output layer). Основные компоненты предлагаемой нейронной сети представлены на Рис. 2 (а).



а) б)

Рис. 2. а) Структурная схема нейронной сети б) Матрица ошибок классификации

В качестве функции потерь использовалась категориальная кроссэнтропия, которая описывает, насколько близко предсказанное распределение к истинному распределению. В качестве алгоритма оптимизации подбора весов и повышения скорости обучения используется оптимизатор «Adam» [19], который вычисляет индивидуальные адаптивные скорости обучения для различных параметров из оценок первого и второго моментов градиентов.

Проверка точности работы классификатора, обученного на тестовой выборке, производилась на данных, не участвовавших в процессе обучения нейронной сети, что обеспечивает оценку обобщающей способности модели к обучению и исключает влияние эффекта переобучения на значение показателя точности.

Процент правильного распознавания рассчитывается как отношение числа правильно классифицированных образцов к числу всех образцов. На тестовой выборке такой показатель составил 95%. Для достижения данного результата потребовалось 21 эпоха. Ниже на рисунке (Рис. 2 (б)) представлена матрица ошибок, показывающая распределение случаев правильного и ошибочного распознавания по классам.

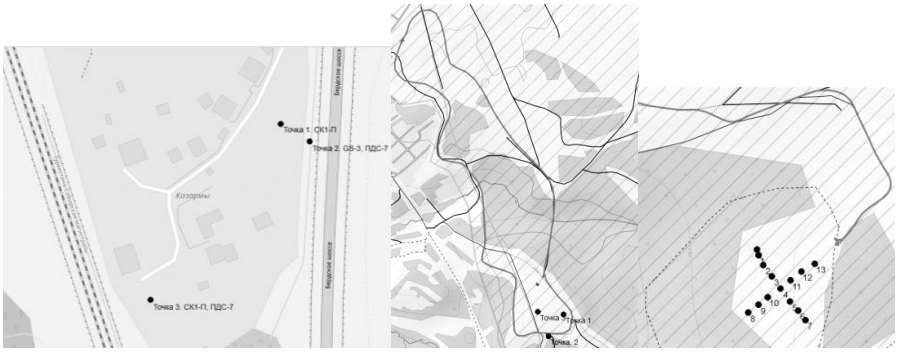
4 . Методика проведения экспериментов

Для получения статистики записей шумов проводились эксперименты по записи сейсмических и акустических колебаний, порождаемых различными видами движущейся транспортной техники – железнодорожной, автомобильной, гусеничной.

Был проведён ряд натурных экспериментов по записи и обнаружению транспортных колебаний на расстояниях до 3-х километров. Регистрация сейсмоакустического волнового поля от транспортных средств осуществлялась с помощью трёхкомпонентных сейсмических датчиков GS-3[20] и СК1-П [21], а также акустических датчиков ПДС-7 [22]. Запись колебаний осуществлялась с помощью регистраторов серии «Байкал-АСН», которые представляют собой 3-канальные автономные сейсмические станции расширенного частотного диапазона с внутренним модулем GPS, каналом USB 2.0 для связи с компьютером и внутренней памятью на основе FLASH- или SD-карт [23].

С целью исследования особенностей сейсмических и акустических сигналов от различного вида железнодорожного и автомобильного движущегося транспорта вблизи расположения оживленной автомобильной дороги и железнодорожных путей были проведены эксперименты по записи колебаний. Дополнительно была проведена серия экспериментов по записи сейсмических и акустических колебаний от тяжелого колесного и гусеничного транспорта, направленная на получение записей данных для уточнения характеристик каждого из них в отдельности. Эксперименты проводились утреннее, дневное, вечернее и ночное время суток в разные сезоны года. На Рис. 3 в качестве примера приведены карты расстановок

аппаратуры некоторых из проведенных экспериментов. Точные значения геометрических параметров расстановки датчиков в каждом эксперименте определялись по GPS.



а) б)

Рис. 3. Схема эксперимента по записи сейсмоакустических колебаний а) от железнодорожного и автомобильного транспорта б) от тяжелого колесного и гусеничного транспорта

Полученные данные являются материалом для предварительного анализа и выделения информативных признаков, а также обучения и тестирования нейронной сети.

Ниже на Рис. 4 приведен пример нормированных спектрально-временных функций, полученных для различного вида транспорта. Видно, что спектр колебаний от тяжелого транспорта располагается в области низких и инфранизких частот. Для некоторых из рассматриваемых видов транспорта характерно наличие узкополосной составляющей в спектре сигнала.

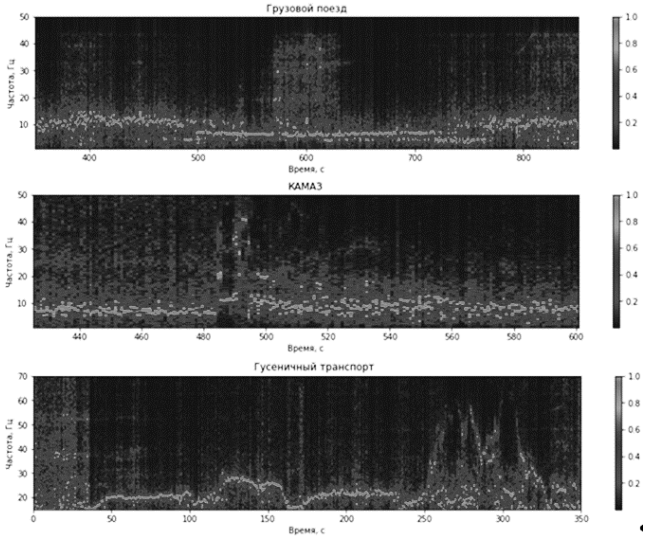


Рис. 4. Нормированная СВФ записей сейсмических колебаний от движущегося транспорта

5 Заключение

В работе предложен и реализован алгоритм распознавания транспортных объектов по сейсмическим колебаниям на основе сверточной нейронной сети. В качестве входных данных были использованы спектрально-временные признаки сейсмических колебаний, возникающих при движении тяжелых видов транспорта средств таких, как электропоезд, грузовой поезд, тяжелый колесный и гусеничный виды транспорта. Алгоритм был протестирован на данных полевых экспериментов, собранных в различной местности и в разные сезоны года. Достигнут результат правильного распознавания видов транспорта около 95%. Алгоритм работает на данных, полученных при различных дальностях движения транспорта, начиная от 700 метров и

выше в зависимости от типа транспорта. Полученные при этом результаты одновременно по дальности обнаружения при той же точности распознавания превосходят ранее полученные другими авторами.

6 Благодарности

Работа выполнена в рамках госзадания FWNM–2022–0004.

Литература

[1] Левковская Т.В., Козлов Э.В., Мурашко Н.И. Обработка сейсмических сигналов в интеллектуальных системах пассивной локации. Информатика. 2010;(3(27)) с. 89-96

[2] Хайретдинов М.С., Авроров С.А. Обнаружение и распознавание взрывных источников. // Вестник НЯЦ РК. — 2012. — №12. — с.17–24.

[3] Алямкин С.А. Классификация объектов в сейсмической системе обнаружения с учетом параметров их движения. Автореф. дис. кан. тех. наук. – Новосибирск.: 2014. – 21с.

[4] X. Jin, S. Sarkar, A. Ray, S. Gupta and T. Damarla, "Target Detection and Classification Using Seismic and PIR Sensors," in IEEE Sensors Journal, vol. 12, no. 6, pp. 1709-1718, June 2012, doi: 10.1109/JSEN.2011.2177257.

[5] P. Khunarsal, C. Lursinsap, and T. Raicharoen, 'Very short time environmental sound classification based on spectrogram pattern matching', Information Sciences, vol. 243, no. Complete, 2013, pp. 57–74, doi:10.1016/j.ins.2013.04.014.

[6] S. E. Küçükbay, M. Sert and A. Yazici, "Use of Acoustic and Vibration Sensor Data to Detect Objects in Surveillance Wireless Sensor Networks," 2017 21st International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 2017, pp. 207-212, doi: 10.1109/CSCS.2017.35.

[7] G. Jin, B. Ye, Y. Wu and F. Qu, "Vehicle Classification Based on Seismic Signatures Using Convolutional Neural Network," in IEEE Geoscience and Remote Sensing Letters, vol. 16, no. 4, pp. 628-632, April 2019, doi: 10.1109/LGRS.2018.2879687.

[8] Y. Wang, X. Cheng, P. Zhou, B. Li and X. Yuan, "Convolutional Neural Network-Based Moving Ground Target Classification Using Raw Seismic Waveforms as Input," in IEEE Sensors Journal, vol. 19, no. 14, pp. 5751-5759, 15 July15, 2019, doi: 10.1109/JSEN.2019.2907051.

[9] M. Kalra, S. Kumar and B. Das, "Seismic Signal Analysis Using Empirical Wavelet Transform for Moving Ground Target Detection and Classification," in IEEE Sensors Journal, vol. 20, no. 14, pp. 7886-7895, 15 July15, 2020, doi: 10.1109/JSEN.2020.2980857.

[10] Y. Wang, X. Cheng, X. Li, B. Li and X. Yuan, "Powerset Fusion Network for Target Classification in Unattended Ground Sensors," in IEEE Sensors Journal, vol. 21, no. 12, pp. 13466-13473, 15 June15, 2021, doi: 10.1109/JSEN.2021.3067648.

[11] Q. Zhou, X. Yao, C. Wang, J. Hu, P. Liu and J. Lin, "Adaptive Moving Ground-Target Detection Method Based on Seismic Signal," in IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1-5, 2022, Art no. 2503705, doi: 10.1109/LGRS.2022.3153368.

[12] D. S. Parihar, R. Ghosh, A. Akula, S. Kumar and H. K. Sardana, "Variational Mode Decomposition of Seismic Signals for Detection of Moving Elephants," in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-8, 2022, Art no. 2510908, doi: 10.1109/TIM.2022.3178465.

[13] K. Bin, J. Lin, X. Tong and T. Nie, "Intelligent Moving Target Recognition Based on Compressed Seismic Measurements and Deep Neural Networks," in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-13, 2022, Art no. 5907213, doi: 10.1109/TGRS.2021.3108467.

[14] K. Bin, J. Lin and X. Tong, "Edge Intelligence-Based Moving Target Classification Using Compressed Seismic Measurements and Convolutional Neural Networks," in IEEE Geoscience

and Remote Sensing Letters, vol. 19, pp. 1-5, 2022, Art no. 7502105, doi: 10.1109/LGRS.2021.3055795.

[15]S. Mohine, B. S. Bansod, R. Bhalla and A. Basra, "Acoustic Modality Based Hybrid Deep 1D CNN-BiLSTM Algorithm for Moving Vehicle Classification," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 9, pp. 16206-16216, Sept. 2022, doi: 10.1109/TITS.2022.3148783.

[16]Ahmad, A.B.; Saibi, H.; Belkacem, A.N.; Tsuji, T. Vehicle Auto-Classification Using Machine Learning Algorithms Based on Seismic Fingerprinting. Computers 2022, 11, 148. https://doi.org/10.3390/computers11100148

[17]Tongyu Nie, Sirui Wang, Yuqi Wang, Xunqian Tong & Feng Sun. An effective recognition of moving target seismic anomaly for security region based on deep bidirectional LSTM combined CNN. Multimedia Tools and Applications, 2023, 14p, doi: 10.1007/s11042-023-14382-5.

[18]Доброродный В.И., Копылова О.А., Хайретдинов М.С. Обнаружение и пеленгование транспортных объектов в сейсмоакустических системах наблюдения. Вестник СибГУТИ. 2023;17(1):3-17. https://doi.org/10.55648/1998-6920-2023-17-1-3-17

[19]D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In Proc. of the Int. Conf. on Learning Representations (ICLR), 2015.

[20]GS-20DX трехкомпонентный // ООО «Геоспейс Технолоджис Евразия». URL: https://geospace-ufa.ru/products/geofony-odinochnye-v-korpuse/GS-20DX-trekhkomponentnyy/ (дата обращения: 10.05.2023)

[21]Сейсмоприемник низкочастотный трехкомпонентный СК-1П. Техническое описание и инструкция по эксплуатации. Алма-Ата: НПО Геофизика. Завод Казгеофизприбор, 1977г. 12 с. [ТОО "ЭЛГЕО" https://elgeo.kz/lfgeoph.shtml]

[22]Геофизические методы исследований // авт. В.К. Хмелевской, Ю.И. Горбачев, А.В. Калинин, М.Г. Попов, Н.И. Селиверстов, В.А. Шевнин. // Под редакцией Н.И. Селиверстова. Учебное пособие для геологических специальностей вузов. Петропавловск-Камчатский: изд-во КГПУ, 2004, 232 с.

[23]Техническое описание «Байкала-ACN» // ФГБУН Федеральный исследовательский центр «Единая геофизическая служба Российской академии наук» URL: http://www.gsras.ru/unu/uploads/files/Dataloggers/Baikal-ACN.pdf (дата обращения: 10.05.2023)

О применении переноса Ферми-Уолкера в звёздных наблюдениях: общий подход и солнечная конкретика

Левичев А.В.[1,2], Нещадим М.В.[1], Пальянов А.Ю.[2,3]

[1]Институт математики им. С.Л. Соболева СО РАН, пр. акад. Коптюга, д. 4, г. Новосибирск, 630090, Россия.
[2]Институт систем информатики им. А.П. Ершова СО РАН, пр. акад. Лаврентьева, д. 6, г. Новосибирск, 630090, Россия.
[3]Новосибирский государственный университет, ул. Пирогова, д. 2, г. Новосибирск, 630090, Россия

alevichev@gmail.com, neshch@math.nsc.ru, palyanov@iis.nsk.su

Аннотация. *В контексте приложений к наблюдательной астрономии, рассматривается использование (в рамках как специальной, так и общей теории относительности) параллельного переноса Ферми-Уолкера. Мы следуем схемам, предложенным в монографиях Синга и Сакса/Ву. Система Ферми-Уолкера составлена для одного (довольно известного) солнечного эксперимента и найдено её решение, что позволяет математически строго найти 'траекторию Солнца на небосводе'.*

Ключевые слова: моделирование астро-наблюдений в мире Минковского, параллельный перенос Ферми-Уолкера, траектория Солнца на небосводе.

1 Введение

Авторы заинтересовались тем, что в [1] (см. с.70, ПРИЛОЖЕНИЕ) и в [2] (начиная с Секции 4.3, и далее на сс. 186, 187) для теоретического описания некоторых экспериментов, связанных с солнечными наблюдениями (или, более общо – с астро-наблюдениями, т.е., с наблюдениями звёзд), используется пространство-время Минковского. Значительную часть этих экспериментов (и их теоретическую подоплёку) И. А. Еганова, автор книги [2], соотносит с именем Н. А. Козырева, известного советского астронома. Ниже фигурируют некоторые конструкции теории относительности, которые могут быть примененены (или уже применяются) для теоретического обоснования/описания этих экспериментов. Одной из них является параллельный перенос Ферми-Уолкера (Fermi-Walker) вдоль времени-подобной кривой. Соответствующая (этому математическому понятию) часть результатов нашей статьи была анонсирована в [3]. В данной статье мы попытались (более подробно, чем в [3]) отследить появление и проанализировать использование нескольких понятий, имеющих непосредственное отношение к теории относительности. В этот список мы включили, в частности, следующие термины: 'истинное Солнце', 'видимое Солнце'. Их кратко обозначаем так: иС, вС – соответственно (дальнейшие пояснения и точные ссылки нами приводятся ниже).

2 Предварительное обсуждение и частичная формулировка результатов

Сразу же отметим, что (в итоге, см. Секцию 4) нами будут строго определены два термина: 'направление на вС' и 'направление на иС'. Кроме того, под 'вС' и 'иС' будут пониматься некие два события (т.е., точки) в пространстве-времени Минковского **M**. Но сначала кратко отследим появление этих терминов в литературе.

В [4] термин ‘истинное положение Солнце’ впервые употребляется на с. 368 (используемые в [4] словосочетания ‘истинное положение Солнце’ и ‘истинное Солнце’ мы считаем синонимами).

В [5] используются оба эти термина; кроме того, на с. 352 авторы говорят, что их формула (1) «получена на основе общепринятых астрометрических соотношений и подробно обсуждается в работах [6, 7]». В этой формуле (1) фигурирует угол между направлениями на вС и на иС, где под С подразумевается наблюдаемая звезда. Сразу же отметим, что (при моделировании в **M**) речь идёт о подсчёте углов между направлениями/векторами в т.н. 3-пространстве R_z (мгновенного) наблюдателя z, см. ниже Секцию 4. Как известно, метрика, индуцируемая в R_z из **M**, является евклидовой. В [6, 7] термины вС и иС тоже фигурируют, причём под ‘С’ также можно понимать любую наблюдаемую звезду, а не только Солнце.

Уместно вовлечь в наше теперешнее обсуждение и статью [8]: она была опубликована в 1980г., но мы используем её текст (сс. 132 – 140) из сборника “Время и звёзды: к 100-летию со дня рождения Н.А.Козырева”. На наш взгляд, ни в одной из статей [4-8] не даны строгие определение терминов вС и иС. Математически аккуратные определения вС и иС мы обнаружили в [9]: см. Рис. 2 на с. 43. Эти определения совпадают с введёнными в [3], но авторы тезисов [3], отправляя их в печать, не знали о статье [9]. Отметим также, что Рис.2 из [9] – это, фактически, расширенная версия Рис. 1 из [8, с. 137]. При этом мы продолжаем считать, что в [8] термины вС и иС не были строго определены (хотя бы потому, что там присутствуют такие словосочетания, как ‘три возможных канала связи’ и ‘воздействие через время, которое может передаваться не только мгновенно, но и со скоростью света’). Далее (в Секции 4) мы воспроизводим (согласно [9, с.43]) определения обоих терминов и находим (в случае Солнца) угол между направлениями на них. Полученное нами значение угла α (между направлениями на вС и на иС) значительно (на несколько порядков) меньше, чем в [2, с.188], где α близок к 2°. Вот, на наш взгляд, объяснение такого расхождения: каждый из двух вышеупомянутых рисунков апеллирует к миру Минковского **M**, но подсчёт угла производится не на основе геометрии этого мира. Дальнейшее обсуждение продолжим ниже, после введения других необходимых определений.

3 О пространственно-временном моделировании астро-наблюдений

В соответствии с общей теорией относительности (= ОТО), рассматривается ‘искривлённое’ пространство-время M (синоним: мир M). В [2, с.188] приведены ссылки на [10], мы же используем [11]. Лишь начиная с Секции 4, под M будем понимать (‘плоский’) мир Минковского. Через M_z обозначаем касательное пространство в точке (= событии) z. Из [11, с.41]: «наблюдатель – это времени-подобная кривая γ в M, параметризованная собственным временем». Из [11, Секция «Оси гироскопов», с.50]: “При выполнении астрономических экспериментов, наблюдатель γ должен уметь решать следующую задачу: когда единичный вектор X_1 в его локальном 3-пространстве R_z в момент собственного времени u_1 имеет то же самое ‘пространственное направление’, что единичный вектор X_2?” Здесь X_2 - в 3-пространстве, соответствующем другому собственному времени u_2 наблюдателя γ. Само же R_z определяется как ортогональное дополнение касательного вектора A к γ в её точке z.

Определение **небесной сферы** L_z ([11, с. 131]) таково, что каждая такая L_z содержится в касательном пространстве M_z (и даже в R_z). Поэтому терминологию [2, с.179] (со словосочетаниями типа «небесная сфера наземного наблюдателя как его мир событий»), следует отличать от стандартной: в ОТО под событиями понимаются лишь точки самого пространства-времени M.

Так как (в следующей секции) Солнце будет моделироваться как точка, то (‘мгновенное’) направление на Солнце задаётся вектором, исходящим из z (= центра L_z) и оканчивающимся на точке небесной сферы. Понятие ‘наблюдаемой траектории Солнца на небосклоне’ не является тривиальным. Именно, со с.51 в [11]: два вектора (удобно пронумеровать их и считать единичными), X_1 в R_1 и X_2 в R_2 имеют **одно и то же направление** (авторы: 3-пространственное направление, конечно же), **если один получается из другого переносом Ферми-Уолкера** (вдоль кривой γ). Применив этот перенос, получим (как бы **одну**) небесную сферу L, в которой мы наблюдаем ‘движение Солнца’ (см. [2, с. 178: Рис. 18, его левую часть]). В следующей секции центром сферы L будет точка $z = P(0)$, т.е. мы выберем значение параметра $t = 0$ на кривой γ. Здесь под $P(t)$ подразумевается точка кривой γ из [2, с. 187], см. наше уравнение (4.2) в следующей Секции.

Замечание 1. В [2, с.178] приведён Рис.18: Небесная сфера наземного наблюдателя как его мир событий. На наш взгляд, ω на Рис.18 – это ‘наблюдаемая траектория Солнца на небосклоне’. А ω на Рис.23 (см. [2, с.186]) – это наша кривая γ, мировая линия 'точечного' телескопа. Автор же книги [2], И. А. Еганова пояснила так: “Первый рисунок для двух моментов чисто иллюстративный – для демонстрации (согласно [12]) `небесной сферы' как `Мира событий'. Подробная (адекватная) астрономическая ситуация изображена, насколько это возможно, на втором чертеже”. Действительно, термин ‘Небесная сфера’ присутствует в [12, Предметный указатель], где дана ссылка на с. 23 и на с. 288. На с. 23 мы обнаружили лишь словосочетание ‘так называемая небесная сфера’, т.е. её определение не было приведено. Что касается с. 288, то правильная ссылка такова: с. 297. Приводимое на этой странице определение задает, фактически, вышеупомянутую сферу L. Она расположена в 3-пространстве R_z, событие z зафиксировано, а радиус сферы L может быть выбран любым положительным числом (а после зафиксирован, конечно же).

Тем самым, Рис. 18 из [2] намного более содержателен, чем просто некая иллюстрация. В частности, на сайте [13] приведено множество таких солнечных траекторий. Вероятно, эти траектории построены ‘по точкам’, в то время как наше решение (см. ниже Секции 5, 6) динамической системы Ферми-Уолкера позволяет задать эти траектории **аналитическими формулами**. По-видимому, предлагаемая нами конкретика применения переноса Ферми-Уолкера для определения траектории Солнца (или, более общо, наблюдаемой звезды) является оригинальной (см., также, нашу Секцию 6).

4 В мире Минковского: мировая линия Солнца и мировая линия γ телескопа на Земле

Координаты x^1, x^2, x^3, x^4 возникают при выборе ортонормированного базиса e_1, e_2, e_3, e_4 в M: лоренцев скалярный квадрат каждого из векторов e_1, e_2, e_3 равен 1, а для e_4 – минус единице.

Солнце моделируем как точечный объект, вот его мировая линия в этих координатах ([2],с.187):

$$S(t) = (RC_\varphi, 0, RS_\varphi, ct). \tag{4.1}$$

Вот мировая линия (‘точечного’) телескопа на поверхности Земли (см. [2, Рис. на с.187]):

$$P(t) = (rC_\lambda C_{\omega t}, rC_\lambda S_{\omega t}, rS_\lambda, ct). \tag{4.2}$$

Выше, $R, r, c, \varphi, \lambda, \omega$ – это некие используемые в астрономии величины, см. [2], с.187. В данной статье достаточно знать, что это константы (отметим, всё же, что R – расстояние от Земли до Солнца, r – радиус сферической Земли, c – скорость света). В контексте рассматриваемой нами задачи (как и в [2]), можно считать R константой. Параметр t – это ‘лабораторное’ время. Через C_φ, S_φ обозначаем косинус и синус угла φ. Аналогично – C_λ, S_λ; $C_{\omega t}$, $S_{\omega t}$. Вот соотношение между исходным параметром t в (4.1), (4.2), и длиной дуги s на кривой γ:

$$t = s/(c\sqrt{1 - m^2}). \tag{4.3}$$

Тем самым, точка $t = 0$ является и точкой $s = 0$. Обозначим соответствующее событие так: $P_0 = (rC_\lambda, 0, rS_\lambda, 0)$. В (4.3) и далее, через m обозначена постоянная $\omega rC_\lambda/c$, она чрезвычайно мала.

Единичный (т.е., его скалярный квадрат равен минус 1 – ведь метрика лоренцева!) касательный вектор A в каждой точке s кривой γ таков:

$$A = \{ -mS_{\omega t}, mC_{\omega t}, 0, 1\}/\sqrt{1 - m^2}. \tag{4.4}$$

В событии P_0 вектор A равен

$$A_0 = \{ 0, m/\sqrt{1 - m^2}, 0, 1/\sqrt{1 - m^2}\}. \tag{4.5}$$

На мировой линии Солнца, событие $t = 0$ таково:

$$S_0 = (RC_\varphi, 0, RS_\varphi, 0). \tag{4.6}$$

В [2, с.178], направление на ‘иС’задаётся вектором

$$T = P_0S_0 = -\{RC_\varphi - rC_\lambda,\ 0,\ RS_\varphi - rS_\lambda,\ 0\}. \tag{4.7}$$

Вектор T ортогонален (в смысле лоренцевой метрики) вектору A_0. Другими словами, событие S_0 – это пересечение мировой линии Солнца с тем (проходящим через P_0) 3-пространством R_0, которое ортогонально вектору A_0. Тем самым, строго определяя термин ‘направление на иС’, как в [2], так и нами (что было отмечено в [3], на сс. 74, 75) используется хорошо известное понятие: ‘одновременность по Эйнштейну’.

А понятие ‘вС’, как мы уже отметили выше (в конце Секции 2), нами (следуя [11]) вводится (чуть ниже!) иначе, чем это было сделано в [2]. Именно, в [2, с.187] на 3-пространство R_0 проектируется не изотропный, а времени-подобный 4-вектор $P(t)S(t - e)$ и где мы полагаем $t = 0$. Излагаем правильный, согласно [11, с.130], подход:

Найдём событие S^* на мировой линии Солнца, которое состоит в испускании фотона, попадающего в телескоп в момент $t = 0$. Очевидно, что для события S^*, $t = t^*$ должно быть отрицательным. В силу элементарных геометрических свойств конусов, искомое S^* удобно найти как (единственную) точку пересечения светового ‘конуса прошлого’ (с вершиной в P_0) и мировой линии Солнца. Обозначим этот конус через $\bar{C}$, его уравнение таково:

$$(x^1 - rC_\lambda)^2 + (x^2)^2 + (x^3 - rS_\lambda)^2 = (x^4)^2. \tag{4.8}$$

Подставляя вместо всех x^k соответствующие координаты (искомого) события $S(t^*)$, согласно (4.1), получаем

$$S^* = (RC_\varphi,\ 0,\ RS_\varphi,\ -\sigma), \tag{4.9}$$

где (положительное)

$$\sigma = -ct^* = [(RC_\varphi - rC_\lambda)^2 + (RS_\varphi - rS_\lambda)^2]^{1/2}. \tag{4.10}$$

Изотропный вектор S^*P_0 равен $\{rC_\lambda - RC_\varphi,\ 0,\ rS_\lambda - RS_\varphi,\ \sigma\}$.

Согласно [11, с.130], направление на ‘вС’ задаётся таким вектором u (ортогональным вектору A_0), что

$$-u = xA_0 + S^*P_0. \tag{4.11}$$

Найдём коэффициент x в (4.11), воспользовавшись равенством нулю лоренцева произведения:

$$< u,\ A_0> = 0. \tag{4.12}$$

Ясно, что (4.12), с учётом (4.11), эквивалентно равенству $x = -\sigma/\sqrt{1 - m^2}$, откуда

$$u = \{RC_\varphi - rC_\lambda,\ m\sigma/(1-m^2),\ RS_\varphi - rS_\lambda,\ m^2\sigma/(1-m^2)\}. \tag{4.13}$$

Лоренцево скалярное произведение векторов T и u равно

$$(RC_\varphi - rC_\lambda)^2 + (RS_\varphi - rS_\lambda)^2. \tag{4.14}$$

В силу (4.10), это произведение равно σ^2. Так как каждый из векторов ортогонален A_0, то это и их евклидово скалярное произведение. Чтобы найти квадрат косинуса угла между T и u, надо поделить квадрат найденного произведения на квадраты длин векторов T и u.

Из (4.7) и (4.10) следует, что квадрат длины вектора T равен σ^2. Согласно (4.13), квадрат длины вектора **u** равен $\sigma^2/(1-m^2)$. Поэтому, квадрат косинуса угла между T и u равен $1 - m^2$. В частности, угол между T и u чрезвычайно близок к нулю. Нами доказана следующая

Теорема 1. Квадрат косинуса угла α (между направлением на истинное Солнце и направлением на видимое Солнце) равен $1 - m^2$.

5 Динамическая система Ферми-Уолкера и её решение

В [3] была приведена (с точностью до опущенного постоянного множителя в правой части) система ОДУ, задающая векторное поле $F = \{F^1, F^2, F^3, F^4\}$ на кривой γ, переносимое вдоль неё параллельно (в смысле параллелизма Ферми-Уолкера). Вот этот множитель (он не был приведён в [3]):

$$m\omega/(c(m^2 - 1)\sqrt{1 - m^2}), \tag{5.1}$$

Ниже (см. формулы (5.2) и далее) мы, в частности, находим общее решение этой системы. Для наших целей достаточно продолжать рассматривать ‘упрощённую’ систему Ферми-Уолкера, не добавляя множитель (5.1) в её правой части. Поэтому он отсутствует, см. ниже, в (5.2), которую мы продолжаем называть системой Ферми-Уолкера. Эту систему (и

множитель (5.1)) нетрудно получить непосредственным применением формул (1.72) из [10, с.21]. Отсутствие каких-либо трудностей применения (1.72) объясняется относительной простотой нашей кривой (4.2), являющейся, фактически, винтовой линией. Кроме того, ковариантные производные в (1.72) из [10, с.21] сводятся к обычным производным, так как мир M – 'плоский'. Опуская выкладки и используя нижнюю нумерацию координат, подытоживаем:

Теорема 2. Система ОДУ, задающая векторное поле $F = \{F_1, F_2, F_3, F_4\}$ на кривой γ с уравнением (4.2), переносимое вдоль этой кривой параллельно (в смысле параллелизма Ферми-Уолкера) такова (напоминаем, что множитель (5.1) нами опущен):

$$\begin{gathered} \dot{F}_1 = -mF_2 + \cos(\omega t)\, F_4 \\ \dot{F}_2 = mF_1 + \sin(\omega t)\, F_4 \\ \dot{F}_3 = 0 \\ \dot{F}_4 = \cos(\omega t)\, F_1 + \sin(\omega t)\, F_2, \end{gathered} \tag{5.2}$$

где точка (в левых частях) означает дифференцирование по натуральному параметру, который мы теперь обозначаем не s, как было в (4.3), а τ.

Введем обозначения

$$u = F_1,\ v = F_2,\ w = F_4,\ c = \cos(\omega t)\ ,\ s = \sin(\omega t).$$

В силу системы (5.2) функция F_3 является константой, а (5.2), фактически, равносильна системе

$$\frac{d}{d\tau}\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 0 & -m & c \\ m & 0 & s \\ c & s & 0 \end{pmatrix}\begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{5.3}$$

или

$$\dot{X} = AX,$$

где

$$X = \begin{pmatrix} u \\ v \\ w \end{pmatrix}, \quad A = \begin{pmatrix} 0 & -m & c \\ m & 0 & s \\ c & s & 0 \end{pmatrix}.$$

Далее, обозначим через μ положительный квадратный корень из $(1 - m^2)$. Непосредственно проверяется, что $0, \pm\mu$ – собственные числа матрицы A. Так как $0 < m < 1$, то собственные числа различные. Соответствующие собственные векторы таковы:

$$v_1 = \begin{pmatrix} -s \\ c \\ m \end{pmatrix}, \quad v_2 = \begin{pmatrix} \mu c - ms \\ mc + \mu s \\ 1 \end{pmatrix}, \quad v_3 = \begin{pmatrix} \mu c + ms \\ -mc + \mu s \\ -1 \end{pmatrix}.$$

Эти векторы линейно независимы.

Представим решение X системы (5.3) в виде

$$X = pv_1 + qv_2 + rv_3 \tag{5.4}$$

Векторы v_1, v_2, v_3 удовлетворяют соотношениям

$$\dot{v}_1 = -\frac{\omega}{2\mu}(v_2 + v_3), \quad \dot{v}_2 = \frac{\omega}{\mu}(v_1 - mv_2), \quad \dot{v}_3 = \frac{\omega}{\mu}(v_1 + mv_3), \tag{5.5}$$

Следовательно,

$$\dot{X} = \left(\dot{p} + q\frac{\omega}{\mu} + r\frac{\omega}{\mu}\right)v_1 + \left(\dot{q} - p\frac{\omega}{2\mu} - q\frac{m\omega}{\mu}\right)v_2 + \left(\dot{r} - p\frac{\omega}{2\mu} + r\frac{m\omega}{\mu}\right)v_3. \tag{5.6}$$

Так как

$$AX = q\mu v_2 - r\mu v_3 \tag{5.7}$$

то система (5.3) после подстановки (5.4) в силу (5.6), (5.7) и линейной независимости векторов v_1, v_2, v_3 принимает вид

$$\dot{Y} = BY, \tag{5.7}$$

$$Y = \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \quad B = \begin{pmatrix} 0 & -2k & -2k \\ k & \mu + 2mk & 0 \\ k & 0 & -\mu - 2mk \end{pmatrix}. \tag{5.8}$$

Ниже будет приведено явное решение для тех трёх (‘существенных’) координат, о которых идёт речь в уравнении (5.3) – но уже с учётом правильного постоянного множителя

$$\lambda = (- m\omega)/(c\mu^3)$$

в правой части исходной системы (5.3). Очевидно, что $0, \pm \lambda\mu -$ собственные числа матрицы λA. Собственные векторы этой последней матрицы – те же, v_1, v_2, v_3. Соотношения же (5.5), (5.6), (5.7) меняются. В частности, вместо (5.7), получаем

$$\lambda AX = q\mu\lambda v_2 - r\mu\lambda v_3. \tag{5.9}$$

В (приводимом ниже) Замечании 2, мы поясняем смысл сохранения нами части вычислений для случая системы (5.3). Сейчас же продолжим рассмотрение именно FW-системы, т.е. учитывая множитель λ.

Именно, вместо (5.8) получаем

$$Y = \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \quad B = \begin{pmatrix} 0 & -2k & -2k \\ k & 0 & 0 \\ k & 0 & 0 \end{pmatrix}, \tag{5.10}$$

где $k = \omega/(2c\mu^2)$. Понятно, что собственные числа матрицы B – это 0 и пара комплексно сопряжённых чисел. Легко отыскивается решение $Y = \{p,q,r\}$для FW-аналога системы (5.8) – опускаем эти выкладки.

Обозначим sin ωt и cos ωt через S и C соответственно; рассматриваем их как функции от натурального параметра τ, а не от лабораторного времени t. В частности, производная от S равна $(\omega/(c\mu))$C, а производная от C равна $-(\omega/(c\mu))$S. В выражении для $Y = \{p,q,r\}$ фигурируют cos $(2k\tau)$ и sin $(2k\tau)$, обозначим их через C^* и S^*, соответственно.

Возвращаясь ‘назад’, получим $X(\tau)$:

Теорема 3. Общее решение 3 на 3 FW-аналога системы (5.3) таково:

$X_1 = - 2mSC_1 – (SS^* + \mu CC^*)C_2 + (\mu CS^* - SC^*)C_3$,

$X_2 = 2mCC_1 + (CS^* - \mu SC^*)C_2 + (\mu SS^* + CC^*)C_3$,

$X_3 = 2C_1 + mS^*C_2 + mC^*C_3$.

Здесь C_1, C_2, C_3 – это (произвольные, вещественные) константы, которые надо задать при поиске единственного решения.

Замечание 2. При проверке удобно группировать члены по этим трём константам C_1, C_2, C_3.

Замечание 3. Нами также было получено общее решение системы (5.2). Это решение мы планируем привести в явном виде в следующей статье, так как методом замены независимой переменной оно позволяет найти и решение системы, получающейся из (5.2) введением любого постоянного множителя в её правой части.

6 Полученные результаты и заключение

Продолжая обсуждение одной из основных тем статьи (см. окончание нашей Секции 3), приведём дополнительные математические детали моделирования ‘траектории Солнца на небосклоне’. В каждой точке $z = P(t)$ (см. уравнение (4.2), выше), проделаем точно такое же ‘построение’, какое было нами осуществлено в точке $\boldsymbol{P}_0$ - см. нашу Секцию 4. Т.е. в каждом таком событии z будет найдено ‘направление на вС’, задаваемое вектором $u(t)$ в R_z. Как было отмечено выше, каждое такое R_z канонически изометрично исходно выбранному R_0. Именно, если в R_0 выбран ортонормированный базис $e_1(0)$, $e_2(0)$, $e_3(0)$, то в R_z однозначно задаётся такой ортонормированный базис $e_1(t)$, $e_2(t)$, $e_3(t)$, что каждый $e_m(t)$ получен из $e_m(0)$ параллельным переносом Ферми-Уолкера. На фоне таких двух базисов, упомянутая изометрия f_z 3-пространства R_z на 3-пространство R_0 является отображением ‘по равенству координат’. ‘Небосвод’ – это небесная сфера L в R_0. А направление на ‘Солнце на небосводе’ в момент

времени t задаётся вектором $f_z(u(t))$. Тем самым, в 3-пространстве R_0 траектория W 'Солнца на небосводе' является кривой, параметризованной той же самой переменной t – лабораторным временем наблюдателя (4.2). Как совокупность точек на L, W будет получена пересечением L с лучами, направленными вдоль всех векторов $f_z(u(t))$ и исходящими из центра сферы L.

В следующей статье, применяя полученные выше результаты, мы планируем окончательно конкретизировать уравнение траектории Солнца на небосводе и продолжить обсуждение 'Козыревской' тематики.

7 Финансовая поддержка

Работа А.В. Левичева выполнена в рамках государственного задания Министерства образования и науки РФ для Института математики Сибирского отделения Российской академии наук (проект № FWNF-2022-0006). Работа М.В.Нещадима выполнена в рамках государственного задания Министерства образования и науки РФ для Института математики Сибирского отделения Российской академии наук (проект FWNF-2022-0009).

Литература

[1] Еганова, И. А. Проблема исследования взаимосвязей в мире событий. //*Поиск математических закономерностей Мироздания: физические идеи, подходы, концепции*/Ред. М. М. Лаврентьев. – Новосибирск: Изд-во ИМ, 1999, с. 61-73. – (Избранные труды Второй сибирской конференции по математическим проблемам физики пространства-времени сложных систем (ФПВ-98), Новосибирск, 19-21 июня 1998г.).

[2] Еганова, И. А. Природа пространства-времени/Отв. ред. акад. М.М.Лаврентьев. – Новосибирск: Изд-во СО РАН, филиал "Гео", 2005.

[3] Левичев, А.В.; Пальянов, А.Ю. Наблюдения звёзд и динамическая система переноса Ферми-Уолкера: общий подход и солнечная конкретика/В сборнике *Международная конференция по геометрическому анализу, посвящённая памяти академика Ю. Г. Решетняка.* 23-29 октября 2022 г. Тезисы докладов. – ИПЦ НГУ., 2022.– С.73-75.– ISBN9785443713779.

[4] Лаврентьев М. М., В. А. Гусев, И. А. Еганова, М. К. Луцет, С. Ф. Фоминых, "О регистрации истинного положения Солнца", *Докл. АН СССР*, **315**:2 (1990), 368–370.

[5] Лаврентьев М. М., И. А. Еганова, М. К. Луцет, С. Ф. Фоминых, "О дистанционном воздействии звезд на резистор", *Докл. АН СССР*, **314**:2 (1990), 352–355.

[6] Козырев Н.А., Насонов В.В. Новый метод определения тригонометрических параллаксов на основе измерения разности между истинным и видимым положениями звезд // Астрометрия и небесная механика. – М., Л., 1978, с. 168-179.

[7] Козырев Н.А., Насонов В.В. О некоторых свойствах времени, обнаруженных астрономическими наблюдениями // Проявление космических факторов на Земле и в звездах. – М., Л., 1980, с. 76-84.

[8] Козырев, Н.А. Астрономическое доказательство реальности четырехмерной геометрии Минковского // Проявление космических факторов на Земле и звездах. – М.; Л., 1980. – С. 85 – 93. См. сборник "Время и звёзды: к 100-летию со дня рождения Н.А.Козырева". Изд-во: СПб, Нестор-История, 2008, сс. 132-140.

[9] Еганова И. А., В. Каллис. Основание мира Минковского как математической структуры: к ответу на вопрос Римана. *Математические структуры и моделирование (2017)*, 4(44), 33-48.

[10] Синг, Дж. Л. Общая теория относительности. – М.: ИИЛ, 1963.

[11] Sachs, R. K.; H-H. Wu. General Relativity for Mathematicians, Springer-Verlag, NY. 1977.

[12] Угаров В. А. Специальная теория относительности. – М.: Наука, 1975.

[13] SunEarthTools_SunPath_1667029182462.pdf, https://www.sunearthtools.com/index.php

Знания, Онтологии и Мультимодальные Аксиоматические Теории (Дедуктивное доказательство непротиворечивости формальной теории $Ф^0$, синтезирующей формальную аксиологию, универсальную эпистемологию и философскую онтологию)

В.О. Лобовиков

Институт Философии и Права Уральского отделения Российской академии наук, ул. Софьи Ковалевской, д. 16, г. Екатеринбург, 620108, Россия.
vlobovikov@mail.ru

Аннотация. *Интеллект (как естественный, так и искусственный) рассматривается как сложная система, необходимо включающая и непротиворечиво синтезирующая две качественно различные подсистемы знания, а именно, подсистему знания a priori и подсистему знания эмпирического. Согласно И. Канту, априорное знание законов природы есть предписывание ей этих законов разумом. Для представителей опытного естествознания этот тезис кёнигсбергского мыслителя кажется, мягко говоря, очень странным, но для собственно теоретической концепции универсального интеллекта как целого, учение Канта о предписывании разумом законов природе является не случайным, а очень важным. В данной работе указанная проблема рассматривается сперва с точки зрения некой логически формализованной аксиоматической теории Ф, представляющей собой систему мультимодальной эпистемологии, а затем с точки зрения некой формальной теории $Ф^0$, представляющей собой результат существенной мутации теории Ф. Существенным отличием $Ф^0$ от Ф, является пустота множества логических аксиом теории $Ф^0$; все аксиомы теории $Ф^0$ являются ее собственными аксиомами, сформулированными с использованием языка логики. При этом, в теории $Ф^0$ существует единственное правило логического вывода – modus ponens. На обсуждение выносится некое дедуктивное доказательство логической непротиворечивости формальной теории Ф.*

Ключевые слова: онтология, аксиология, знание, априорное, эмпирическое, формальная мультимодальная аксиоматическая теория, дедуктивное доказательство непротиворечивости

1. Введение

Согласно рационализму Г.В. Лейбница, система *априорного* знания является *необходимой*, хотя и недостаточной, подсистемой всякого интеллекта. Согласно совершенному И. Кантом «*трансцендентальному* повороту» в философии, относящееся к чистому разуму *априорное* знание *строго универсальных* законов природы есть *необходимое условие* возможности ее *опытного* познания. Это условие *организует, объединяет весь возможный опыт* (все принципиально возможные наблюдения и эксперименты). В этой связи кажется вполне правдоподобной гипотеза, согласно которой, в любую удачную конструкцию универсального познающего ИИ-робота с самого начала (в момент её создания) должна быть «вшита» некая *неотделимая* от этой конструкции *априорная* система знания, делающая возможным и *организующая* весь опыт (систему эмпирических знаний) такого ИИ-робота. В данной статье упомянутая трактовка разума, необходимо включающего и непротиворечиво синтезирующего подсистему знания *a priori* и подсистему знания эмпирического, *моделируется* с помощью неких

логически формализованных аксиоматических систем мультимодальной эпистемологии, содержащих в себе также некую формальную аксиологию и онтологию.

Назовем формальной теорией Ф мультимодальную аксиоматическую систему эпистемологии и аксиологии, включающую в себя (1) все аксиомы и правила вывода классической пропозициональной логики, а также (2) *собственные* (не логические, а эпистемологические, аксиологические и т.п.) аксиомы теории Ф, представленные ниже с помощью одиннадцати схем аксиом AX1 – AX11 и определения DF1.

AX1: $A\alpha \supset (\Omega\beta \supset \beta)$. (Здесь принадлежащие метаязыку символы α , β, ω обозначают любые формулы теории Ф, а принадлежащий метаязыку символ Ω обозначает любой элемент множества «модальностей совершенства», точно определяемого в семантике языка-объекта теории Ф.)

AX2: $A\alpha \supset (\Omega(\omega \supset \beta) \supset (\Omega\omega \supset \Omega\beta))$.

AX3 AX3: $A\alpha \leftrightarrow (K\alpha \,\&\, (\neg\Diamond\neg\alpha \,\&\, \neg\Diamond S\alpha \,\&\, \Box(\beta \leftrightarrow \Omega\beta)))$.

AX4: $E\alpha \leftrightarrow (K\alpha \,\&\, (\Diamond\neg\alpha \vee \Diamond S\alpha \vee \neg\Box(\beta \leftrightarrow \Omega\beta)))$.

AX5: $\Omega\beta \supset \Diamond\beta$.

AX6: $(\Box\omega \,\&\, \Box\Omega\omega) \supset \omega$.

AX7: $(t_i{=}{+}{=}t_k) \leftrightarrow (G[t_i] \leftrightarrow G[t_k])$, где: символы t_i и t_k, принадлежащие метаязыку, обозначают некие (любые) *термы языка-объекта* теории Ф; символ «=+=» обозначает бинарное отношение «*формально-аксиологическая равноценность термов*» теории Ф; а «взятие в *квадратные* скобки» есть некая процедура приписывания *онтологического значения* в стандартной интерпретации теории Ф.

AX8: $(t_i{=}{+}{=}g) \supset \Box G[t_i]$.

AX9: $(t_i{=}{+}{=}b) \supset \Box W[t_i]$.

AX10: $(G\alpha \supset \neg W\alpha)$.

AX11: $(W\alpha \supset \neg G\alpha)$.

Определение DF1: $\Diamond\omega$ есть *сокращенное название* (имя) для $\neg\Box\neg\omega$, т.е. $(\Diamond\omega \leftrightarrow \neg\Box\neg\omega)$, по определению (где ω есть некая формула теории Ф).

В шести схемах аксиом AX1 – AX6, символ Ω (принадлежащий к метаязыку) обозначает некий (любой) элемент множества £ = {□, K, T, F, P, D, C, Y, G, O, B, U, J}. Назовем элементы множества £ «модальностями совершенства» или просто «совершенствами». Множество £ модальностей совершенства является подмножеством множества ¥ всех тех модальностей, которые принимаются во внимание теорией Ф, а именно, ¥ = {◊, □, K, A, E, S, T, F, P, D, C, Y, G, W, O, B, U, J}. В связи с этим уместно заметить, что символы S, W и ◊ обозначают в языке мультимодальной теории Ф такие модальности, которые совершенствами не являются, т.е. подставлять их вместо Ω нельзя.

Рассмотрение семантики языка теории Ф запланировано в третьем разделе этой статьи, но, чтобы читатель смог дотерпеть до третьего раздела, забегая вперед, заметим, что привычные символы ◊ и □ обозначают, соответственно, алетические модальности «возможно» и «необходимо». Символы K, A, E, S, T, F, P, D обозначают, соответственно, модальности «агент знает, что …», «агент *a-priori* знает, что …», «агент *эмпирически* (*a-posteriori*) знает, что …», «при некоторых условиях в некотором пространстве-времени, некий субъект (непосредственно или с помощью неких инструментов) *чувственно* воспринимает (*верифицирует* своим *ощущением*), что…», «*истинно*, что…», «агент *верит*, что …», «(в некой непротиворечивой теории) *доказуемо*, что …», «существует некий *алгоритм* (может быть построена некая машина) для *решения*, что …».

Символы C, Y, G, W, O, B, U, J обозначают, соответственно, модальности «логически *непротиворечиво*, что …», «*полно*, что …», «*хорошо*, что …», «*плохо*, что …», «*обязательно*, что …», «*красиво* (*прекрасно*), что …», «*полезно*, что …», «*приятно, радостно*, что …».

Представленная выше система собственных аксиом мультимодальной теории Ф является точным (хотя и не явным) определением значений модальных символов, используемых в этой теории.

Обсуждаемая формальная теория Ф является результатом нескольких существенных изменений (добавлений и обобщений) в ранее предложенных формальных теориях: Σ (Sigma) [8], [9], [10]; Σ+C [11]; Σ+2C [12].

Для того, чтобы адекватно понять вышесказанное и в особенности смысл схем собственных аксиом теории Ф, сформулированных на некоем весьма необычном искусственном языке, необходимо дать точные дефиниции синтаксиса и семантики такого непривычного языка. Именно этому посвящены следующие два раздела статьи.

2. Синтаксис языка-объекта формальной теории Ф: точные определения алфавита, термов и формул этого языка

Записанные на метаязыке схемы собственных аксиом формальной теории Ф сформулированы выше во введении, но, строго говоря, чтобы эти формулировки стали вполне осмысленными и понятными, необходимо дать точные определения понятий «язык-объект теории Ф», «терм теории Ф» и «формула теории Ф». Определения этих понятий теории Ф *сходны (подобны)* дефинициям *аналогичных* понятий теорий: Σ (Sigma) [8], [9], [10]; Σ+C [11]; Σ+2C [12]. Однако, несмотря на это *сходство (подобие)*, строго говоря, в настоящей статье необходимо дать точные определения понятий «язык-объект теории Ф», «терм теории Ф» и «формула теории Ф», так как, вообще говоря, *сходство (подобие) не является отношением эквивалентности*, поэтому, *множества теорем упомянутых теорий не совпадают*. Начнем с определения алфавита обсуждаемого искусственного языка.

1) Строчные латинские буквы p, q, d (и эти же самые буквы с нижними числовыми индексами) принадлежат алфавиту языка-объекта теории Ф. Эти и только эти строчные латинские буквы именуются «*диктум-переменными*». (“*dictum-variables*”). В алфавите языка-объекта теории Ф, *не все строчные латинские буквы именуются диктум-переменными*, так как, согласно данному определению, те строчные латинские буквы, которые принадлежат множеству {g, b, e, n, x, y, z, a, s, h, t, f}, не принадлежат множеству диктум-переменных языка-объекта теории Ф.

2) Строчные латинские буквы a, s, h (и эти же самые буквы с нижними числовыми индексами: a_t, s_m, h_s) принадлежат алфавиту языка-объекта теории Ф. Эти и только эти строчные латинские буквы именуются «*диктум-постоянными*» (“*dictum-constants*”).

3) Привычные логические символы ¬, ⊃, ↔, &, ∨, именуемые, соответственно, «классическим отрицанием», «классической (материальной) импликацией», «классической эквивалентностью», «классической конъюнкцией», «классической не-исключающей дизъюнкцией», принадлежат алфавиту языка-объекта теории Ф.

4) Элементы множества {□, K, A, E, S, T, F, P, D, C, Y, G, W, O, B, U, J}, содержащего □ и некоторые (но не все) заглавные латинские буквы, не имеющие индексов, являются элементами алфавита языка-объекта теории Ф. Эти элементы алфавита именуются в теории Ф *модальными символами* (знаками модальности).

5) Строчные латинские буквы *x*, *y*, *z* (и эти же самые буквы с нижними числовыми индексами) принадлежат алфавиту языка-объекта теории Ф. Эти и только эти строчные латинские буквы именуются «*аксиологическими (ценностными) переменными*» в теории Ф.

6) Строчные латинские буквы «g» и «b», именуемые «*аксиологическими (ценностными) константами*» также являются элементами алфавита языка-объекта теории Ф.

7) Заглавные латинские буквы, имеющие числовые индексы, – E^1, C^1, K^1, K^2, E^2, C_j^n, D_i^n, A_k^n, … суть элементы алфавита языка-объекта теории Ф. Такие заглавные латинские буквы называются «*ценностно-функциональными* символами». Верхний числовой индекс *n* информируем о том, что проиндексированный ценностно-функциональный символ является *n*-местным. Ценностно-функциональные символы могут не иметь нижних числовых индексов. Но, если ценностно-функциональные символы имеют нижние числовые индексы, то, если эти индексы различны, то проиндексированные функциональные символы различны.

8) Знаки “(” и “)”, именуемые «*круглыми* скобками», суть элементы алфавита языка-объекта теории Ф. Эти вспомогательные знаки используются в данной статье как обычно в символической логике, а именно, как чисто технические символы.

9) Знаки “[” и “]”, именуемые «*квадратными* скобками», тоже суть элементы алфавита языка-объекта теории Ф. Однако, здесь очень важно подчеркнуть, что в отличие от «круглых скобок», в теории Ф «*квадратные* скобки» используются не как привычные чисто технические знаки, а как *особые (специальные) символы*, указывающие на собственно *онтологическое значение* того, что в квадратные скобки заключено. Такое нестандартное использование квадратных скобок психологически неожиданно (непривычно). С психологической точки зрения, квадратные и круглые скобки кажутся почти неразличимыми и очень часто используются в естественном языке как синонимы. Но в языке-объекте теории Ф, обсуждаемые два вида скобок имеют *качественно различные* значения (играют существеннно различные роли): использование круглых скобок является чисто техническим (вспомогательным), а заключение в квадратные скобки означает приписывание (фиксацию) *онтологического* значения. *Собственно онтологический* смысл процедуры заключения в квадратные скобки точно определяется в той части данной статьи, которая посвящена *семантике* языка-объекта теории Ф. Однако, даже на уровне *синтаксиса* искусственного языка-объекта теории Ф, квадратные скобки *играют очень*

важную роль в точном определении понятия «*формула* теории Ф». (Это определение дается ниже в данном разделе статьи.) Более того, заключение в квадратные скобки *играет очень важную роль* в точной формулировке некоторых схем аксиом теории Ф, представленных выше во введении.

10) Весьма необычный искусственно созданный символ «=+=», именуемый «*формально-аксиологической эквивалентностью (термов)*» является элементом алфавита языка-объекта теории Ф. Непривычный символ «=+=» *играет важную роль* в точном определении понятия «*формула* теории Ф», а также в точной формулировке некоторых схем аксиом теории Ф, представленных выше во введении.

11) Некий (любой) знак есть элемент алфавита языка-объекта теории Ф, если и только если этот знак принадлежит этому алфавиту, согласно какому-то из пунктов 1) – 10) данного определения.

Некая (любая) конечная последовательность символов называется «*выражением* языка-объекта теории Ф, если и только если эта последовательность состоит из таких и только таких знаков, которые являются элементами алфавита языка-объекта теории Ф. Понятие «*терм* теории Ф» определяется следующим образом.

1) Вышеупомянутые *аксиологические (ценностные) переменные и константы* являются термами теории Ф.

2) Если $Ф_k^n$ есть некий (любой) *n-местный ценностно-функциональный символ,* а t_i, …, t_n суть некие (любые) *термы* теории Ф, то выражение, имеющее вид $Ф_k^n t_i$, …, t_n есть терм теории Ф. (Здесь уместно заметить, что знаки t_i, …, t_n принадлежат метаязыку, так как они обозначают любые термы теории Ф; аналогичное замечание уместно сделать относительно знака $Ф_k^n$, который также принадлежат метаязыку.)

3) Некое (любое) выражение языка-объекта теории Ф является термом теории Ф, если и только если это так, согласно какому-то из пунктов 1) – 2) данного определения.

Теперь перейдем к определению понятия «формула теории Ф». Для этого договоримся, что строчные греческие буквы α, β и ω (принадлежащие метаязыку) обозначают некие (любые) формулы теории Ф. С помощью такого соглашения можно определить понятие «формула теории Ф» следующим образом.

1) Все те строчные латинские буквы, которые были названы выше *dictum-переменными* и все те строчные латинские буквы, которые были названы выше *dictum-постоянными*, являются формулами теории Ф.

2) Если t_i и t_k суть некие (любые) термы теории Ф, то любое выражение, имеющее вид $(t_i=+=t_k)$ есть формула теории Ф.

3) Если t_i есть некий (любой) терм теории Ф, то любое выражение, имеющее вид $[t_i]$, есть формула теории Ф.

4) Если α есть некая (любая) формула теории Ф, а символ Ψ (принадлежащий метаязыку) обозначает некий (любой) *модальный символ* (знак модальности) из множества {□, K, A, E, S, T, F, P, D, C, Y, G, W, O, B, U, J}, то любое выражение языка-объекта теории Ф, имеющее вид Ψα, есть формула теории Ф. Здесь важно иметь в виду, что, строго говоря, выражение Ψα (принадлежащее метаязыку) не есть формула теории Ф, а есть схема формул этой теории.

5) Если α и β суть формулы теории Ф, то все те выражения языка-объекта теории Ф, которые (выражения) имеют формы $\neg\alpha$, $(\alpha \supset \beta)$, $(\alpha \leftrightarrow \beta)$, $(\alpha \,\&\, \beta)$, $(\alpha \vee \beta)$, принадлежат множеству формул теории Ф.

6) Любая конечная последовательность символов, взятых из алфавита языка-объекта теории Ф, есть формула теории Ф, если и только если это так, согласно какому-то из пунктов 1) – 5) данного определения.

Единственным исходным *правилом логического вывода* в теории Ф является *modus ponens*.

Понятие «*чисто логическая* аксиома теории Ф» точно определяется ссылкой на множество схем аксиом *классической пропозициональной логики*, в которых теперь вместо формул классического исчисления высказываний подразумеваются формулы теории Ф.

В *чисто синтаксическом* отношении, понятие «*собственная аксиома* теории Ф» точно определено выше во введении к данной статье. Теперь предстоит рассмотреть собственные аксиомы теории Ф в *семантическом* отношении, так как очевидно, что данные выше точные синаксические дефиниции пока семантически бессмысленны. Но это не результат досадного случайного упущения по небрежности, а результат принятия и последовательного осуществления *научной абстракции* от семантических значений символов; такая абстракция вполне допустима и целесообразна в пределах сферы своей адекватности, но не вообще. Поэтому, теперь перейдем от точно определенного выше синтаксиса искусственного языка-объекта теории Ф к точному определению *семантических значений* символов и выражений этого языка.

3. Семантика языка-объекта формальной теории Ф: точные определения семантических значений термов и формул

В предыдущей части статьи рассматривался исключительно *синтаксис* языка формальной *мультимодальной* теории Ф, поэтому, *явно* ничего конкретного о множестве семантических значений модальных символов сказано не было: значения этих символов были определены точно, но *неявно*, а именно, *аксиоматически*. Теория Ф рассматривалась как именно *формальная*: ее *интерпретация* не предлагалась и не обсуждалась. Теперь настало время перейти к точному определению именно семантики языка-объекта теории Ф. Семантические значения собственно логических символов искусственного языка классической пропозициональной логики общеизвестны и хорошо определены в респектабельных учебниках, поэтому, еще раз определять значения чисто логических символов здесь нет необходимости. А вот существенно новые (почти неизвестные) понятия теории Ф и семантические значения необычных знаков алфавита языка-объекта этой теории необходимо точно определить.

Значения строчных латинских букв p, q, d (и этих же самых букв с нижними числовыми индексами), названных выше *диктум-переменными*, аналогичны значениям общеизвестных в логике *пропозициональных переменных*. Однако, здесь есть некое существенное различие: в теории Ф, диктум-переменные принимают значения из *множества диктумов* (обозначим его символом Δ), которому принадлежат не только все истинные или ложные пропозиции (высказывания), но и все истинные или ложные теории. Согласно статистической норме словоупотребления, слово «*dictum*» переводится с латинского языка как «утверждение некого предложения …» или «выражение некой мысли … в словах». Однако, здесь существует интересная возможность *обобщения*, а именно, перехода от диктума как «утверждения некого предложения …» к диктуму как дизъюнкции «утверждение или некого предложения …, или некой теории …», так как теория тоже есть нечто такое, что можно утверждать. В данной статье принимается презумпция, что любая истинная или ложная теория тоже есть элемент множества Δ. Тем самым объясняется возможность применения (в теории Ф) *de-dicto*-модальностей, например, C (логическая непротиворечивость) и Y (полнота) к истинным или ложным *теориям*.

Если достаточно точно определена *интерпретация* формальной теории Ф, то значением некой (любой) *диктум-постоянной* является (в этой интерпретации) некий вполне определенный (фиксированный) элемент множества Δ, т.е. или некая истинная или ложная сентенция, или некая истинная или ложная теория.

Определение семантических значений есть определение некой функции, именуемой «интерпретация». Для точного определения такой функции необходимо точно определить (1) некое непустое множество, играющее роль «*области интерпретации*» (договоримся обозначать область интерпретации заглавной буквой «*M*») и (2) некого «*интерпретатора (оценщика)*» (договоримся обозначать его заглавной буквой «*V*»). По определению, в любой стандартной интерпретации формальной теории Ф, *M* есть такое непустое множество, каждый элемент которого имеет: (1) одно и только одно *аксиологическое (ценностное) значение* из множества {g (хорошо), b (плохо)}; (2) одно и только одно *онтологическое значение* из множества {e (exists), n (not-exists)}.

Аксиологические переменные (z, x, y, z_i, x_k, y_m) принимают значения из множества *M*.

Аксиологические константы “b” и “g” интерпретируются как “плохо” и “хорошо”, соответственно.

Оценка некого (любого) элемента из множества *M* неким конкретным (фиксированным) оценщиком *V* представляет собой приписывание этому элементу некого *аксиологического значения* (хорошо или плохо). Интерпретатор (оценщик) *V* может быть или индивидуальным или коллективным – неважно. Бесспорно, какое-то изменение оценщика *V* может привести к изменению некоторых относительных оценок, но никакое изменение интерпретатора *V* не может изменить множество законов двузначной алгебры формальной аксиологии, которые (законы) являются не относительными, а абсолютными оценками, а именно, такими *ценностными функциями-константами*, которые принимают значение g (хорошо) при любой возможной комбинации аксиологических значений своих ценностных переменных. Хотя *V* представляет собой *переменную*, принимающую значения из множества всех возможных оценщиков, любая вполне определенная стандартная интерпретация теории Ф необходимо фиксирует значение переменной *V*. Любое изменение значения *V* означает изменение интерпретации теории Ф.

Символы, “e” и “n”, именуемые *онтологическими константами*, обозначают, соответственно, “… *существует*” и “… *не существует*”. По определению, в любой стандартной интерпретации теории Ф, один и только один элемент множества {{g, e}, {g, n}, {b, e}, {b, n}} соответствует

каждому элементу множества *M*. Знаки “e” и “n” принадлежат метаязыку. Согласно определению алфавита языка-объекта теории Ф, символы “e” и “n” не принадлежат языку-объекту этой теории. Тем не менее, онтологические константы “e” и “n” все-таки представлены некоторым образом (опосредованно) на уровне языка-объекта теории Ф процедурой *взятия в квадратные скобки*: «t_j существует» представляется с помощью $[t_j]$; «t_j не существует» представляется с помощью $\neg[t_j]$. Это означает, что процедура *взятия в квадратные скобки* является необходимым аспектом точного определения формально-аксиологической и формально-онтологической семантики формальной теории Ф.

N-местные термы теории Ф интерпретируются как *n-местные ценностные функции*, определенные на множестве *M*. Понятие «*одноместная* ценностная функция» экземплифицируется приведенной ниже таблицей 1. (Напоминаю, что верхний индекс 1, находящийся справа от некой (любой) заглавной буквы, указывает на то, что эта заглавная буква интерпретируется некой *одноместной* ценностной функцией; различие нижних числовых индексов означает различие проиндексированных символов, например, в следующей ниже таблице 1, F_1^1x и F_2^1x являются различными символами.)

Таблица 1. Ценностные функции, зависящие от одного ценностного аргумента

x	L_1^1x	N^1x	F_1^1x	F_2^1x	A_1^1x	A_2^1x	Z_1^1x	Z_2^1x	P^1x	Y^1x	R^1x	I^1x	B^1x	T^1x	J^1x
g	g	b	g	b	g	b	g	b	g	b	b	g	b	b	g
b	b	g	b	g	b	g	b	g	b	g	g	b	g	g	b

В таблице 1, одноместный терм L_1^1x интерпретируется как *одноместная ценностная функция «жизнь, бытие, существование* (чего, кого) *x*». Терм N^1x интерпретируется как ценностная функция *«смерть, небытие* (чего, кого) *x*». Терм F_1^1x – ценностная функция «*свобода* (чего, кого, чья) *x*». F_2^1x – «*свобода от* (чего, кого) *x*». A_1^1x – «*атака, нападение* (чего, кого) *x*». A_2^1x – «*атака, нападение на* (что, кого) *x*». Z_1^1x – «*защита, оборона* (чего, кого, чья) *x*». Z_2^1x – “*защита, оборона от* (чего, кого) *x*”. P^1x – «*положительная оценка, одобрение* (чего, кого) *x*». Y^1x – «*отрицательная оценка, осуждение* (чего, кого) *x*». R^1x – “*сопротивление* (чему, кому) *x*”. I^1x – “*участие, соучастие в* (чем) *x*”. B^1x – “*воздержание, уклонение от* (чего) *x*”. T^1x – «*уничтожение* (чего, кого) *x*». J^1x – «*сохранение* (чего, кого) *x*».

Понятие «*двуместная* ценностная функция» экземплифицируется приведенной ниже таблицей 2. (Напоминаю, что верхний индекс 2, находящийся справа от некой (любой) заглавной буквы, указывает на то, что эта заглавная буква интерпретируется некой *двуместной* ценностной функцией.)

Таблица 2. Ценностные функции, зависящие от двух ценностных аргументов

x	y	K^2xy	S^2xy	N^2xy	X^2xy	T^2xy	J^2xy	A^2xy	Z^2xy	Y^1x	F^2xy	C^2xy	E^2xy	V^2xy
g	g	g	b	b	b	b	g	b	b	b	b	g	g	b
g	b	b	g	b	b	b	g	b	b	b	b	b	b	g
b	g	b	g	b	g	g	b	g	g	g	g	g	b	g
b	b	b	g	g	b	b	g	b	b	b	b	g	g	b

В таблице 2, двуместный терм K^2xy интерпретируется как ценностная функция «*единство x* и *y*», или «*бытие x* и *y вместе*», или «*совместное бытие x* и *y*». Терм S^2xy интерпретируется как ценностная функция «*разделение x* и *y*. N^2xy интерпретируется как ценностная функция «*единство небытия x и небытия y*”. X^2xy – ценностная функция «*бытие y’s без x*», или «*совместное бытие y* с отсутствием *x*». T^2xy – «*уничтожение* (чего, кого) *x* (чем, кем) *y*». J^2xy – «*сохранение* (чего, кого) *x* (чем, кем) *y*». A^2x – «*атака, нападение* (чего, кого) *y* на (что, кого) *x*». Z^2x – «*защита, оборона* (чего, кого, чья) *y* от (чего, кого) *x*». Y^1x – «*отрицательная оценка, осуждение* (чего, кого) *x* (чем, кем) *y*». F^1x – «*свобода, освобождение* (чего, кого) *y* от (чего, кого) *x*». C^2xy интерпретируется как ценностная функция «*существование (содержание, присутствие) y* в (внутри) *x*». E^2xy – ценностная функция «эквивалентность (тождество ценностных значений) *x* и *y*». V^2xy – ценностная функция «*выбор и реализация такого и только такого элемента множества* {*x, y*}, который является: 1) *наилучшим*, если *x* и *y* оба хороши»; 2) *наименее плохим (меньшим из двух зол)*, если *x* и *y* оба плохи»; 3) хорошим, если *x* и y имеют противоположные ценностные значения». Таким образом, терм V^2xy интерпретируется как «*исключающий* выбор и реализация *только оптимального* элемента множества {*x, y*}». Дополнительные экземплификации понятия «*двуместная ценностная функция»* содержатся в [8] – [12].

Чтобы исключить возможность досадных недоразумений, здесь уместно заметить и подчеркнуть, что в любой стандартной интерпретации формальной теории Ф, символы B_1^1x, N_1^1x, C_1^1x, K^2xy, C^2xy, E^2xy, V^2xy обозначают *не предикаты, а ценностные функции*. Предикатами (в стандартной интерпретации теории Ф) являются, например, такие выражения языка-объекта этой теории, которые имеют формы (t_i=+=t_k), (t_i=+=g), (t_i=+=b).

По определению семантики теории Ф, если t_i есть терм этой теории, то, принадлежащая этой теории формула, имеющая вид [t_i], является в стандартной интерпретации этой теории *или истинным или ложным* высказыванием «t_i существует». Таким образом, по определению, в некой (любой) стандартной интерпретации, формула, имеющая вид [t_i], истинна, если и только если она имеет *онтологическое значение* «e (существует)» в этой интерпретации. Также по определению, формула, имеющая вид [t_i], ложна, если и только если она имеет *онтологическое значение* «n (не существует)» в этой интерпретации теории Ф.

По определению семантики теории Ф, в некой (любой) стандартной интерпретации этой теории, формула, имеющая вид (t_i=+=t_k), является *высказыванием*, имеющим форму «(ценностная функция) t_i *формально-аксиологически эквивалентна* (ценностной функции) t_k» ; это высказывание является *истинным*, если и только если *в данной интерпретации* термы t_i и t_k принимают одинаковые *аксиологические значения* из множества {g (хорошо), b (плохо)} при любой возможной комбинации *аксиологических значений* входящих в эти термы *аксиологических переменных*.

По определению семантики теории Ф, в некой (любой) стандартной интерпретации этой теории, формула, имеющая вид (t_i=+=b), является *высказыванием*, имеющим форму «(ценностная функция) t_i есть *формально-аксиологическое противоречие*», или «t_i есть *тождественно плохая* ценностная функция»; это высказывание является *истинным*, если и только если *в данной интерпретации* терм t_i принимает *аксиологическое значение* b (плохо) при любой возможной комбинации *аксиологических значений* входящих в этот терм *аксиологических переменных*.

По определению семантики теории Ф, в некой (любой) стандартной интерпретации этой теории, формула, имеющая вид (t_i=+=g), является *высказыванием*, имеющим форму «(ценностная функция) t_i есть *формально-аксиологический закон*», или «t_i есть *тождественно хорошая* ценностная функция»; это высказывание является *истинным*, если и только если *в данной интерпретации* терм t_i принимает *аксиологическое значение* g (хорошо) при любой возможной комбинации *аксиологических значений* входящих в этот терм *аксиологических переменных*.

В естественном языке, известном своей нечеткостью и многозначностью, обозначаемое символом «=+=» бинарное отношение *формально-аксиологической эквивалентности* ценностных функций выражается словами «есть», «является», «значит», «эквивалентно», «следовательно» и т.п., нередко заменяемыми тире. Однако, общеизвестно, что эти же самые слова естественного языка систематически используются в формальной логике для выражения собственно логической связки «есть» и соответствующих собственно логических операций: «импликация», «эквивалентность (истинностных функций)». Таким образом, налицо лингвистический факт *омонимии* вышеупомянутых слов в естественном языке. Поэтому, во избежание досадных недоразумений, порождающих иллюзии парадоксов, рассуждая на стыке между формальной логикой и формальной аксиологией, упомянутую омонимию нужно иметь в виду; качественно различные значения омонимов должны быть эффективно разделены.

Формальная теория Ф является *мультимодальной*. В ней систематически рассматривается довольно большое множество очень важных для философии модальностей *de dicto*, а именно, множество ¥ = {◊, □, K, A, E, S, T, F, P, D, C, Y, G, W, O, B, U, J}. Напомню, что символы ◊ и □, обозначают модальности «*возможно*, что …» и «*необходимо*, что …», соответственно. Символы K, A, E, S, T, F, P, D, C, Y, соответственно, являются именами для модальных выражений «агент *знает*, что …», «агент *a-priori знает*, что …», «агент *эмпирически (a-posteriori) знает*, что …», «при некоторых условиях в некотором пространстве-времени, некий субъект (непосредственно или с помощью неких инструментов) *чувственно* воспринимает (*верифицирует* своим *ощущением*), что…», «*истинно*, что…», «агент *верит*, что …», «(в некой непротиворечивой теории) *доказуемо*, что …», «существует некий *алгоритм* (может быть построена некая машина) для *решения*, что …», «*логически непротиворечиво*, что …», «семантически *полно*, что …». Символы G, W, O, B, U, J, соответственно, являются именами для модальных выражений «*хорошо*, что …», «*плохо*, что …», «*обязательно*, что …», «*красиво* (эстетически совершенно), что …», «полезно*, что …*», «*приятно, радостно*, …». Значения всех обсуждаемых в теории Ф модальностей *определяются в ней аксиоматически*., т.е. *совокупностью аксиом* (и правил вывода) этой теории. Аксиоматическое определение не является явным, но оно вполне пригодно для некоторых важных видов достаточно точных рассуждений.

Особое внимание в теории Ф уделяется подмножеству £ множества ¥, где £ = {□, K, T, F, P, D, C, Y, G, O, B, U, J}. Элементы множества £ называются «модальностями совершенства» или

просто «совершенствами». В представленных выше первых шести схемах аксиом (AX1 – AX6) теории Ф, принадлежащий к метаязыку символ Ω обозначает некий (любой) элемент множества £ = {□, K, T, F, P, D, C, Y, G, O, B, U, J}. Таким образом, все модальности совершенства рассматриваются в мультимодальной теории Ф с некой *единой* точки зрения. Возможно, это представляет собой некий важный шаг в приближении к идеалу так называемой «универсальной логики» [1], [2].

4. Формальная теория $Ф^0$, синтезирующая формальную аксиологию, универсальную эпистемологию и философскую онтологию

В качестве дальнейшего развития концепции синтеза априоризма и эмпиризма, в настоящей работе предлагается *существенная мутация* формальной теории Ф путем удаления из нее всех логических аксиом и добавления в список схем ее *собственных* аксиом еще одной, а именно, AX12: (Aα ⊃ ©), где символ © обозначает какую-то (любую) произвольно взятую тавтологию классической пропозициональной логики. Пусть результат такой мутации теории Ф именуется формальной теорией $Ф^0$. Хотя логических аксиом в новой теории нет, в ней есть одно правило логического вывода, а именно, *modus ponens* (и при желании, с его помощью можно получить, из допущения Aα, множество *производных* правил логического вывода в $Ф^0$). Что дает переход от Ф к $Ф^0$? Он создает возможность *синтеза* систем *априорного* знания, основанного на *классической* логике высказываний и систем знания *эмпирического*, в той или иной форме *отклоняющегося от классической* пропозициональной логики. Упомянутый *синтез* может быть промоделирован графически с помощью логического квадрата и гексагона, объединяющего системы априорного знания и эмпирические теории, а также, соответственно, теории, основанные на классической пропозициональной логике и как-то отклоняющиеся от нее. В частности, можно предположить, что $Ф^0$ представляет собой формализацию интуиционизма Брауэра [3], [4], [5], [7] более адекватную его философии логики, чем та формализация, которую предложил Гейтинг [6], [7].

Пусть символ “t” обозначает некую (любую) теорию, имеющую рекурсивно перечислимое множество аксиом, а “Emp(t)” обозначает мета-теоретическое свойство «теория t, как целое, является *эмпирической*». Символ “Apr(t)” обозначает мета-теоретическое свойство «теория t представляет собой систему исключительно *априорного* знания». Символ “Cla(t)” – «теория t основывается на *классической* пропозициональной логике». “Con(t)” – «теория t логически *непротиворечива*». “Com(t)” – «теория t *полна*». “Dec(t)” – «теория t *разрешима*».

Понятия “Emp(t)” и “Apr(t)” определяются следующим образом. Определение DF2: Apr(t) ↔ (Cla(t) & Con(t) & Com(t) & Dec(t)). Определение DF3: Emp(t) ↔ (¬Cla(t) ∨ ¬Con(t) ∨ ¬Com(t) ∨ ¬Dec(t)). Следствие 1: Apr(t) ↔ (¬Emp(t). Следствие 2: Emp(t) ↔ ¬Apr(t)).

Система логических отношений между упомянутыми выше мета-теоретическими понятиями графически моделируется приведенными ниже (на Рис. 1) логическим квадратом и гексагоном. Впервые эти квадрат и гексагон концептуальной оппозиции мета-теоретических понятий были представлены в устном докладе автора на седьмом всемирном конгрессе по логическому квадрату [13].

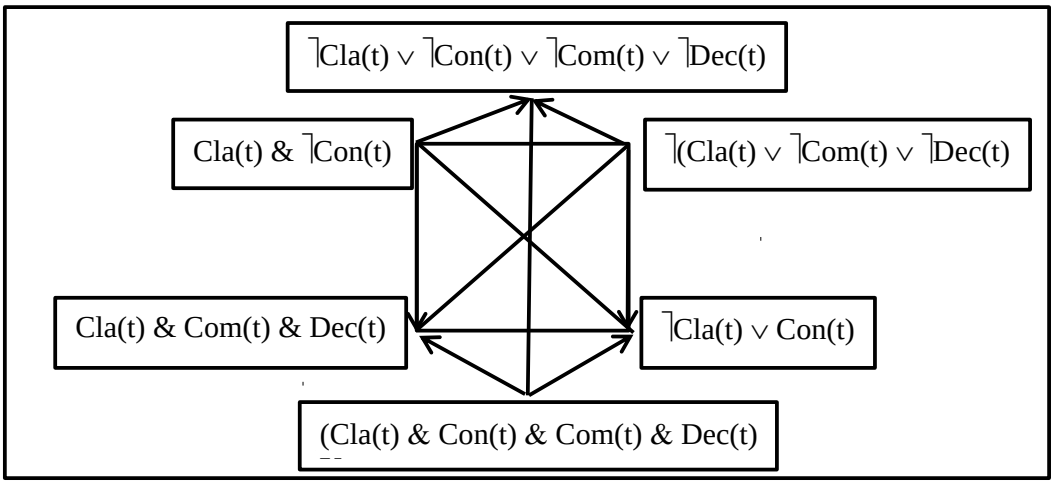


Рис.1. Логические квадрат и гексагон, объединяющие эмпирические теории с априорными

5. Доказательство непротиворечивости теории Φ^0

Для доказательства непротиворечивости формальной теории Φ^0 используется некая ее *интерпретация*. Прежде всего, перейдем от схем аксиом теории Φ^0, записанных на метаязыке, к аксиомам, записанным на ее языке-объекте. Результат такого перехода – следующая ниже система аксиом.

AX1*: $Ap \supset (\Box q \supset q)$.

AX2*: $Ap \supset (\Box(p \supset q) \supset (\Box p \supset \Box q)$.

AX3*: $Ap \leftrightarrow (Kq \,\&\, (\neg\Diamond\neg p \,\&\, \neg\Diamond Sp \,\&\, \Box(q \leftrightarrow \Box q)))$.

AX4*: $Ep \leftrightarrow (Kq \,\&\, (\Diamond\neg p \,\&\, \Diamond Sp \,\&\, \neg\Box(q \leftrightarrow \Box q)))$.

AX5*: $(\Box p \supset \Diamond p)$.

AX6*: $(\Box q \,\&\, \Box\Box q) \supset q)$.

AX7*: $(F_1^1x=+=L_1^1x) \leftrightarrow (G[F_1^1x] \leftrightarrow G[L_1^1x])$.

AX8*: $(F_1^1x=+=g) \supset \Box G[F_1^1x])$.

AX9*: $(F_1^1x=+=b) \supset \Box W[F_1^1x])$.

AX10*: $(Gp \supset \neg Wp)$.

AX11*: $(Wp \supset \neg Gp)$.

AX12*: $(Ap \supset (p \supset p))$.

Дефиниция DF1*: $\Diamond p$ есть *сокращенное название* для $\neg\Box\neg p$, т.е. $(\Diamond\omega \leftrightarrow \neg\Box\neg\omega)$, по определению.

Некая функция, обозначаемая символом #, и именуемая *интерпретацией* аксиом формальной теории Φ^0, точно определяется следующей ниже совокупностью пунктов 1—26. (В данной части работы, символ «t» обозначает «истинно», а символ «f» обозначает «ложно»).

1) Для любых формул δ и λ теории Φ^0, и для любой классической бинарной логической связки $\otimes$, истинно, что $\#(\delta \otimes \lambda) = (\#\delta \otimes \#\lambda)$.

2) Для любой формулы δ теории Φ^0, истинно, что $\#\neg\delta = \neg\#\delta$. (Здесь символ «$\neg$» обозначает классическую унарную логическую операцию «отрицание».)

3) $\#Ap = f$.

4) $\#\Box q = f$.

5) $\#q = t$.

6) $\#p = t$.

7) $\#\Box(p \supset q) = f$.

8) $\#\Box p = f$.

9) $\#Kp = t$.

10) $\#\Diamond\neg p = t$.

11) $\#\Diamond Sp = t$.

12) $\#\Box(q \leftrightarrow \Box q) = f$.

13) $\#Ep = t$.

14) $\#\Diamond p = t$.

15) $\#\Box\Box q) = f$.

16) $\#G[F_1^1x] = t$. (В данной интерпретации символ F_1^1 обозначает *ценностную функцию* «*свобода* (кого) *x*». См. Табл.1.)

17) $\#G[L_1^1x] = t$. (В данной интерпретации символ L_1^1 обозначает *ценностную функцию* «*бытие, жизнь* (кого) *x*». См. Табл.1.)

18) $\#(F_1^1x=+=L_1^1x) = t$.

19) $\#(F_1^1x=+=g) = f$.

20) $\#(F_1^1x=+=b) = f$.

21) $\#\Box G[F_1^1x] = f$.

22) $\#\Box W[F_1^1x] = f$.

23) $\#Gp = t$.

24) $\#Wp = f$.

25) $\#\Box\neg p = f$.

26) $\#(p \supset p) = t$.

В определенной выше *интерпретации* # формальной теории Φ^0, аксиомы AX1*—AX12* истинны, определение DF1* истинно, а единственное правило логического вывода (modus ponens) сохраняет истинность, следовательно, преложенная интерпретация # формальной теории Φ^0, является моделью этой теории и, следовательно, формальная теория Φ^0 является логически непротиворечивой, так как для нее существует модель.

Литература

[1] Béziau, J.-Y. (Ed.). *Universal Logic: An Anthology*. Springer Nature, 2012.

[2] Béziau, J.-Y., Buchsbaum, A., Rey, C. (Eds.). *Handbook of the 6th World Congress and School on Universal Logic*. Vichy, France, University Clermont Auvergne, 2018.

[3] Brouwer, L. E. J. (1975). On the unreliability of the logical principles. In A. Heyting (Ed.). *The collected works of L.E.J. Brouwer* (pp. 107-111). Amsterdam: North Holland.

[4] Brouwer, L. E. J. (1983). Intuitionism and formalism. In P. Benacerraf and H. Putnam (Eds.). *Philosophy of Mathematics* (pp. 77-89). Prentice Hall, Englewood Cliffs N.J., Cambridge: Cambridge University Press.

[5] Dalen, D. Van. (1981). Brouwer's Cambridge lectures on intuitionism. Cambridge: The University Press.

[6] Heyting, A. (1956). *Intuitionism: an introduction*. Amsterdam: North-Holland.

[7] Heyting, A. (ed.). (1975). *L. E. J. Brouwer: Collected Works* (Volume 1: *Philosophy and Foundations of Mathematics*), Amsterdam and New York: Elsevier.

[8] Lobovikov, V. O. Knowledge Logic and Algebra of Formal Axiology: A Formal Axiomatic Epistemology Theory Sigma Used for Precise Defining the Exotic Condition Under Which Hume-and-Moore Doctrine of Logically Unbridgeable Gap Between Statements of Being and Statements of Value is Falsified, *Antinomies* 2020, 20/4, pp. 7–23, https://doi.org/10.24411/2686- 7206-2020-10401

[9] Lobovikov, V. O. A Formal Deductive Inference of the Law of Inertia in a Logically Formalized Axiomatic Epistemology System Sigma from the Assumption of Knowledge A-Priori-Ness, *Journal of Applied Mathematics and Physics*, 2021, 9/3, pp. 441-467, https://doi.org/10.4236/jamp.2021.93031

[10] Lobovikov, V. O. Formal Inferring the Law of Conservation of Energy from Assuming A-Priori-ness of Knowledge in a Formal Axiomatic Epistemology System Sigma, *Journal of Applied Mathematics and Physics*, 2021, 9/5, pp. 1011-1040, https://doi.org/10.4236/jamp.2021.95070

[11] Lobovikov, V. O. A Logically Formalized Axiomatic Epistemology System Σ + C and Philosophical Grounding Mathematics as a Self-Sufficing System, *Mathematics*, 2021, 9/16, 1859 https://doi.org/10.3390/math9161859

[12] Lobovikov, V. O. Axiomatizing Philosophical Epistemology, a Formal Theory "Sigma + 2C", and Philosophical Foundations of Mathematics, in: A.V. Hlebalin (ed.). *Analytical Philosophy: History Trajectories and Development Vectors: Proceedings of the International Scientific Conference Devoted to 80th Anniversary of V.V. Tselischev – Scientific Leader of the Institute of Philosophy and Law of Siberian Branch of Russian Academy of Sciences (February 25–26, 2022).* Novosibirsk: Ofset TM, 2022, pp. 43-52.

[13] Lobovikov, V. O. A new metatheoretic square and hexagon uniting empirical theories with *a-priori* ones, and uniting theories based on the classical logic with ones based on a nonclassical logic, in: Lorenz Demey, Dany Jaspers & Hans Smessaert (eds.). *Handbook of the 7th World Congress on the Square of Opposition "Square 2022", September 9-13, 2022* (www.square-of-opposition.org), Leuven, Belgium, 2022, pp. 53-54.

Сравнение традиционных и формальных структурных единиц знаменного распева

Мирошниченко Л.А., Бахмутова И.В., Гусев В.Д.

Институт математики им. С.Л.Соболева, пр. ак. Коптюга, 4, г. Новосибирск, 630090, Россия.

luba@math.nsc.ru, bakh@ math.nsc.ru, gusev@math.nsc.ru

Аннотация. *Объектом исследования являются древнерусские церковные песнопения, представленные в знаменной форме записи. Знаменный распев – это одноголосное пение (монодия), подчиняющееся системе осмогласия. Певческие книги состоят из 8 частей (гласов), отличающихся составом «устойчиво» повторяющихся мелодических оборотов (попевок), списки которых были собраны различными авторами в певческих руководствах, называемых кокизниками. Однако устойчиво повторяющиеся мелодические обороты можно выделять и формальными методами непосредственно из текстов певческих рукописей, обогащая тем самым своды попевок, изложенных в теоретических руководствах. В статье описываются различные подходы к выделению структурных единиц знаменного распева. Показана взаимосвязь между структурными единицами, выделенными формальными методами, и множеством попевок из подборки известного музыкального медиевиста В.М. Металлова. Полученные результаты имеют непосредственное отношение к проблеме нотолинейной реконструкции знаменных песнопений.*

Ключевые слова: знаменные песнопения, двознаменники, попевки, внутригласовые инварианты, сложностные разложения

1 Введение

Объектом исследования в данной работе являются древнерусские церковные песнопения, представленные в знаменной форме записи. Проблема перевода их в современную нотолинейную форму носит дешифровочный характер и в общем случае беспометной нотации не утратила своей актуальности. Общие сведения о знаменном распеве можно найти в [1]. Это одноголосное пение (монодия), подчиняющееся системе осмогласия. Каждая певческая книга состоит из 8 частей (гласов). Песнопения разных гласов отличаются составом «устойчиво» повторяющихся мелодических оборотов (попевок). Одно и то же знамя может по разному интерпретироваться в разных гласах. Поэтому построение азбук и выявление структурных единиц осуществляется отдельно по каждому гласу. Но и в пределах одного гласа имеет место многозначность соответствия «знамя – нота» [2]. Конкретное знамя может иметь множество различных интерпретаций, отличающихся друг от друга интервально-ритмическими и звуковысотными характеристиками. В то же время один и тот же мелодический оборот может быть записан разными знаменами.

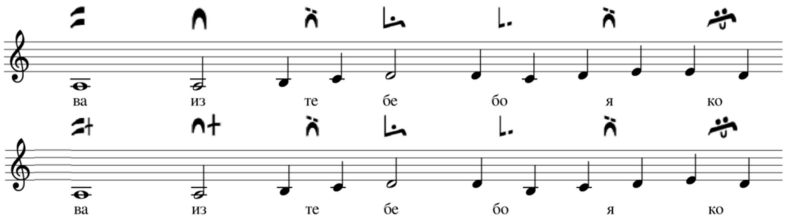


Рис. 1. Сравнение фрагментов песнопений «На стиховне. Слава и ныне. Богородичен» из разных Октоихов.

Рисунок 1 иллюстрирует характер «допустимых» разночтений на мелодическим уровне между одноимёнными песнопениями из одного гласа разных Октоихов [3]. Можно видеть, что первым двум нотам соответствуют близкие, но формально разные знамена, и, наоборот, пятое и шестое знамена совпадают в обоих фрагментах, но распеваются по разному. Такой аспект многозначности может считаться проявлением допустимого варьирования.

Вследствие многозначности соответствия «знамя – нота» дешифровка с использованием одних только азбук, содержащих допустимые интерпретации каждого знамени, приводит к большому количеству ошибок, для устранения которых следует привлекать контекст. Это равносильно использованию более крупных структурных единиц, выделению которых из текстов знаменных песнопений и посвящена данная работа.

Материал исследования формируется на основе двознаменников – певческих книг, представленных параллельно в знаменной и нотолинейной форме. В нашем случае это многожанровые певческие книги – Октоихи конца XVII– начала XVIII в.в. (Российская Национальная Библиотека, г. С.–Петербург, Соловецкое собрание, шифры 618/644, 619/647 и QI 188). В первом гласе Октоиха Сол. 619/647 содержится 27 песнопений, Сол 618/644 — 28 песнопений и QI188 — 17, среди которых много одноимённых [3]. Длины песнопений колеблются в диапазоне от 35 до 157 знамен.

Основными структурными единицами знаменного распева являются попевки: устойчивые мелодические обороты, представленные преимущественно цепочками из 3 – 7 знамен [4]. Среди известных авторских сборников попевок выделяется подборка В.М. Металлова [5], содержащая свыше 500 попевок в нотолинейной форме.

Целью работы является выделение формальными методами структурных единиц из текстов реальных певческих рукописей и сопоставление их с множеством попевок из авторской подборки В.М. Металлова. Предполагается, что в качестве формальных структурных единиц могут выступать повторяющиеся фрагменты двознаменников, удовлетворяющие определенным ограничениям.

2 Структурные единицы знаменного распева

2.1 Попевки

Формального определения понятия «попевка» не существует. Впервые этот термин появляется в русских азбуках XVI века в связи с объяснением распева отдельных знамен в разных гласах. В начале XVII века появились руководства, называемые кокизниками, содержащие музыкальные обороты с подтекстовкой (фрагментами текстов из певческих книг или названиями оборотов). В дальнейшем появились кокизники «толкования по гласам». Они отличались четким делением на восемь частей, соответствующим разным гласам. В каждом гласе был представлен достаточно обширный свод мелодических оборотов, названных составителями попевками или строками [4].

Общими чертами музыкальных оборотов из кокизников XVII века, именуемых попевками, являлись: 1) обязательное указание на конкретный глас; 2) ограничение на количество знаков (от трех до семи знамен); 3) связь с текстом (окончание оборота совпадает с окончанием слова); 4) наличие устойчивого каданса (завершения) [4].

Кручинина А.Н. [4] определяет попевку как характерную мелодическую гласовую формулу, зафиксированную неизменным графическим начертанием, состоящую их двух элементов: вариантно изменяемого подвода и архетипа. В [4] все попевки разделены по типам кадансов (завершений) на 24 группы, каждая из которых имеет свою первооснову (архетип) из трёх знамен. Знамена, завершающие попевку, по большей части интерпретируются целой нотой и относятся преимущественно к семейству статей (, , , и др.).

В используемом в данной работе авторском сборнике В.М. Металлова [5] попевки представлены в нотолинейной форме. Самые короткие попевки в этой подборке — «хамила» (e2d4e4d2) и «подъем малый» (d2e4f4g1). Здесь и далее используется кодировка высоты нот ступенями обиходного звукоряда GAHcdefgabCD [1], где символ «c» соответствует ноте «до», а длительности нот кодируются символами 1, 2, 2., 4, 4., 8 (1 – , 2 – , и т.д.). Самая длинная попевка в подборке Металлова — «стезя высокая»: e4d4e4f4g2f2e4d4f2e2d4c4d4e4f2e2d2e2f2g1 (20 нот).

Для поиска попевок из подборки Металлова в нотолинейных текстах двознаменников нами использована модификация алгоритма множественного поиска образцов в текстах,

позволяющая после предобработки образцов (построения детерминированного автомата) осуществлять поиск всех образцов (попевок) за один просмотр текста [6].

Далеко не все попевки из подборки Металлова обнаруживаются в текстах Октоихов. Так, из 94 попевок, приписанных Металловым к первому гласу, 37 совсем не встречаются в Октоихах, 11 попевок встречаются по одному разу, и только 40 попевок встречаются по три и более раз. Нотолинейная компонента двознаменников покрывается попевками из подборки Металлова лишь на 25-30% (в зависимости от гласа).

Лидером по числу вхождений в тексты песнопений первого гласа Октоихов является попевка «колесо или шибок малый» (e4f4d4e1d1). Она встречается 50 раз в 32 песнопениях. Второе место по частоте встречаемости (39 раз в 32 песнопениях) занимает «кулизма средняя» (e4f4g2e4f4e4d4c4d4c4H4A1). Попевка «колчанец» встречается 26 раз в 21 песнопении.

2.2 Внутригласовые инварианты

Внутригласовыми инвариантами (ВИ) мы называем «устойчивые» повторы, а именно цепочки знамен длины l, не меняющие нотную интерпретацию в конкретном гласе [7]. Места вхождения ВИ в текст можно трактовать как своего рода «островки надёжности», характеризующиеся существенным снижением уровня неоднозначности в интерпретации знамен. Словари внутригласовых инвариантов и квазиинвариантов (почти однозначно интерпретируемых цепочек знамен) [7] успешно используются в алгоритмах дешифровки [8].

Для произвольной последовательности символов (текста) $T = t_1t_2\ldots t_N$ длины N, составленной из элементов конечного алфавита Σ, l-грамма определяется как связная цепочка текста, содержащая l символов, т.е. фрагмент текста $T[i : i + l - 1]$, включающий элементы с i-го по $(i + l - 1)$-й ($i \geq 1$; $i + l - 1 \leq N$). Текст $T = t_1t_2\ldots t_N$ можно представить в виде последовательности l-грамм, выделяемых скользящим окном ширины l, движущимся вдоль текста с шагом в один символ. Полное число l-грамм равно $N - l + 1$. С учетом возможных повторений число различных l-грамм $M_l \leq N - l + 1$.

Частотной характеристикой l-го порядка текста T называется совокупность элементов $\Phi_l(T) = \{\phi_{l1}, \phi_{l2}, \ldots, \phi_{l\,Ml}\}$, где $\phi_{l\,Mi}$ $(1 \leq i \leq M_l)$ есть пара: <i-я l-грамма – x_i, число ее вхождений в текст – $F_l(x_i)$ >. Пусть $l_{\max}(T)$ – наибольшее значение l, при котором в тексте T еще содержатся повторяющиеся l-граммы. Совокупность частотных характеристик $\Phi(T) = \{\Phi_1(T), \Phi_2(T),\ldots, \Phi_{l\,max}(T)\}$ будем называть полным частотным спектром текста T [9].

Полный частотный спектр повторов – универсальный способ описания любого текста. Он может быть использован, например, для решения различных классификационных задач, связанных с текстами. Группу текстов (множество песнопений) можно рассматривать как единый текст (с пробелами между составляющими его компонентами). Полный частотный спектр такого текста будет содержать наряду с повторами, локализованными в отдельных текстах, также и фрагменты, общие для разных текстов. В нашем случае, когда объектом исследования являются двознаменники, можно говорить о параллельных текстах $T = t_1t_2\ldots t_N$, где каждый элемент есть пара (знамя, цепочка нот), т.е. $t_i = (z_i, v_{i1} \ldots v_{ik_i})$, $1 \leq k_i \leq 5$ (в большинстве случаев), $1 \leq i \leq N$, $z_i \in \Sigma_1$ (алфавит знамен), $v_{ij} \in \Sigma_2$ (алфавит нот). Соответствующим образом расширяется и понятие полного частотного спектра повторов для двознаменников.

Из полного спектра повторяющихся цепочек «знамя − нота» выделяется множество внутригласовых инвариантов (ВИ): цепочек знамен, однозначно интерпретируемые в данном гласе. Формально, это фрагменты $(z_i, v_{i1} \ldots v_{ik_i})\,(z_{i+1}, v_{i+1,1} \ldots v_{i+1,k_{i+1}}) \ldots (z_j, v_{j1} \ldots v_{jk_j})$, где $j = i + l - 1$, обладающие свойством: $F_{zn}([z_i : z_j]) = F_{zn_nota}([(z_i, v_{i1} \ldots v_{ik_i}) : (z_j, v_{j1} \ldots v_{jk_j})])$: частота цепочки знамен совпадает с частотой этой же цепочки с учетом ее нотной интерпретации. Кроме того, частота должна быть не менее некоторого заданного порога $F \geq F_{\min}$; в данной работе полагаем $F_{\min} = 3$. Однако заметим, что нотолинейная цепочка, соответствующая инварианту, может быть записана и другими знаменами (возможно, синонимичными), её частота в этом случае выше, чем $F_{zn}([z_i : z_j])$.

В текстах первого гласа трех Октоихов выявлено 695 внутригласовых инвариантов длиной от 3 до 11 знамен. Они характеризуются высокой степенью вложенности и пересекаемости. Так, инвариант (e4f4) (g2) (e4f4e4d4) встречается в первом гласе Октоихов 27 раз, 26 раз встречается фрагмент (g2) (e4f4e4d4) (c4d4c4H4) и его расширение влево и вправо (e4f4) (g2) (e4f4e4d4) (c4d4c4H4) (A1), соответствующее

попевке «кулизма средняя». Дальнейшие расширения этого фрагмента также встречаются среди инвариантов, но уже с меньшей частотой и, наконец, инвариант максимальной длины, содержащий все перечисленные фрагменты, встречается 5 раз: (H4c4) (d2) (c4H4) (c4d4) (e2.d4) (e4f4) (g2) (e4f4e4d4) (c4d4c4H4) (A1).

Высокой степенью вложенности характеризуются и попевки. Так, попевка «таганец средний», совпадающая с инвариантом (f4g4a2)(g4f4)(g2)(e1), содержит при этом «таганец малый» (a2g4f4g2e1) и вложена в «таганец большой» (e4d4f4g4a2g4f4g2e1).

Результаты сравнения каждого из выделенных инвариантов с каждой попевкой представлены в разделе 3.

2.3 Длинные повторяющиеся фрагменты разных песнопений

Наряду с полным частотным спектром, понятие сложности конечной символьной последовательности также связано с представлением последовательности в терминах повторов. Длинные нерасширяемые повторы, выделяемые с помощью сложностных разложений, могут рассматриваться в качестве формальных структурных единиц.

В случае двух последовательностей S_1 и S_2 мы представляем одну из них, например, S_1, в виде конкатенации фрагментов из S_2 [10]. Покрытие S_1 фрагментами из S_2 ведётся от начала к концу (слева–направо). На каждом шаге копируется фрагмент S_2, максимально удлиняющий уже синтезированный префикс S_1. Один и тот же фрагмент S_2 может использоваться несколько раз для копирования, либо совсем не использоваться. Если в S_1 встретился символ, отсутствующий в S_2, этот символ генерируется. Последовательность фрагментов H(S_1 / S_2), соответствующую описанному процессу синтеза S_1 по S_2, называем *сложностным разложением* S_1 по S_2, а число компонентов в нём c(S_1 / S_2) – сложностью S_1 относительно S_2. Следует отметить, что в общем случае $c(S_1 / S_2) \neq c\ (S_2 / S_1)$. Сложностные разложения – более компактное по сравнению с полным частотным спектром представление текстов в виде повторов.

Из всего множества повторов, содержащихся в H(S_1 / S_2), интерес представляют наиболее длинные. По определению, компоненты сложностных разложений нерасширяемы вправо, но иногда их можно расширить влево [11]. Таким способом формируется множество нерасширяемых повторов, длина которых не меньше некоторого заданного порога r. Он определяет количество повторов, подлежащих рассмотрению, т.е. значимых с точки зрения эксперта.

В случае множества последовательностей $\Pi = \{S_1,S_2,\ldots S_m\}$ строится разложение текста $S_i \in \Pi$ ($2 \le i \le m$) по каждому тексту $S_j \in \Pi$ ($1 \le j < i$), т.е. проводится всего C_m^2 сравнений в отличие от метода, описанного в разделе 2.2. Напомним, что множество внутригласовых инвариантов формируется на основе одного текста, образованного конкатенацией $S_1S_2\ldots S_m$. В данном случае $\Pi = \{S_1,S_2,\ldots S_m\}$ – множество песнопений первого гласа трех певческих книг (Октоихи Сол. 619/647, Сол. 618/644 и QI 188), $m = 72$. Сравнение песнопений можно проводить отдельно по трем составляющим: «стихотворному» тексту, знаменному и нотному, а также по их комбинациям, например, одновременно знаменному и нотному текстам. При анализе сложностных разложений на первый план выступают такие характеристики как длина максимального компонента в разложении и число компонентов разложения, длина которых превышает заданный порог r.

При сравнении стихотворных текстов песнопений одного гласа конкретной певческой книги иногда выявляются повторы длиной в несколько слов. Длинным совпадающим фрагментам текстов зачастую соответствуют и одинаковые распевы, особенно когда повторяющиеся стихотворные фрагменты приходятся на начало (концы) песнопений [10]. Однако нередки ситуации, когда распевы одинаковых фрагментов текстов отличаются довольно сильно.

Довольно много повторов выявляется на знаменном уровне (вне зависимости от текста песнопения). В частности, в первом гласе трех Октоихов выявлено 691 повторов знамен длины 7 и выше, у 450 из них совпадают и нотные тексты, в остальных случаях имеет место варьирование. С повышением порога r, количество повторов уменьшается. В первом гласе трех Октоихов выявлено 90 повторов длиной 10 и более, у 42 из которых совпадают нотные тексты. Самые длинные повторы содержатся в одноименных песнопениях разных Октоихов. Так, общий фрагмент песнопений «Стихиры восточны» из Октоихов Сол. 619/647 и Сол. 618/644 (№7) состоит из 39 знамен, повтор между песнопениями «Слава и ныне. Богородичен» из этих

же Октоихов (№ 10) содержит 30 знамен и 28 общих знамен у песнопений «На господи воззвах. Слава и ныне. Богородичен. (№ 1). В нотных текстах, соответствующих длинным повторам на знаменном уровне, наблюдается варьирование. Длина общих фрагментов между разными (не одноименными) песнопениями не превышает 15 знамен. Например, 15 знамен содержит общий фрагмент песнопений «Стихиры восточны» (№ 7) и «В понедельник утра. Богородичен» (№ 21), нотные тексты также совпадают и соответствуют распеву фиты [12].

В данной работе нас больше интересует выявление повторов на нотном уровне с целью сопоставления их с известными попевочными структурами из подборки Металлова. В нотных текстах первого гласа трех Октоихов выявлено 625 повторяющихся фрагментов длиной 15 и выше, в 130 из них совпадают и знаменные тексты. При уменьшении порога *r* до 10 количество повторов возрастает до 2707. Этим повторам соответствуют цепочки знамен длиной от 3 до 51, среди которых только 16% (431) представляют собой точные повторы, остальные – несовершенные. Максимальный повтор состоит из 83 нот у одноименных песнопений «Антифон первый» из Октоихов Сол. 619/647 и Сол. 618/644 (№13). Их знаменные тексты (51 знамя) слегка варьированы.

Итак, сопоставление песнопений одного гласа выявляет наличие значимых повторов на всех трёх уровнях, многие из которых взаимосвязаны, но имеются и примеры практически полного отсутствия какой-либо корреляции между повторами разного типа.

Каждый из повторяющихся фрагментов (мелодических оборотов), выявляемых с помощью сложностных разложений, сравнивается затем с каждой из 94 попевок подборки Металлова. При этом выделяются случаи, когда фрагмент совпадает с одной из попевок, содержит попевку или является частью попевки. Результаты такого сравнения представлены в следующем разделе.

3 Сравнение попевок с формальными структурными единицами

Сравнение попевок с внутригласовыми инвариантами и выделенными фрагментами сложностных разложений обсуждается на материале первого гласа трех Октоихов. Сопоставимые результаты получены и по остальным гласам.

Из 40 попевок подборки Металлова, встречающихся не менее трёх раз каждая в текстах первого гласа Октоихов, 14 попевок совпадают с инвариантами и 11 вложены в инварианты. Из 94 попевок первого гласа, представленных в подборке В.М. Металлова, 67 попевок содержат инварианты в качестве своих фрагментов. Например, самая длинная попевка «стезя высокая» отсутствует в текстах Октоихов, однако встречается довольно много ее фрагментов, 47 из которых совпадают с внутригласовыми инвариантами.

Чаще всего попевке соответствует один из инвариантов, однако наблюдаются случаи совпадения одной попевки с двумя - тремя инвариантами. Это происходит, когда разным вхождениям попевки в тексты Октоихов соответствуют записи разными знаменами и каждая из таких записей встречается не менее трех раз. Например, попевке «колчанец», входящей в тексты первого гласа Октоихов 26 раз, соответствуют три разных внутригласовых инварианта, отличающиеся последним знаменем (кадансом) и заменой сложного знамени (f2e2) на конкатенацию простых: (f2) (e2). А именно, 4 раза встречается инвариант (e4d4) (f2e2) (d4c4) (d4e4f2) (e1) (d1) , 3 раза — инвариант (e4d4) (f2e2) (d4c4) (d4e4f2) (e1) (d1) и 8 раз — (e4d4) (f2) (e2) (d4c4) (d4e4f2) (e1) (d1).

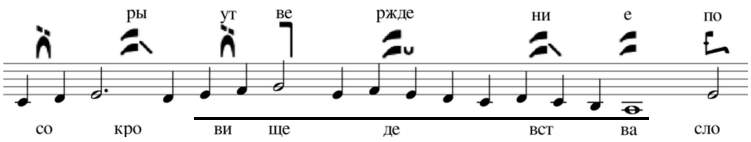


Рис 2. Совпадающие мелодические обороты песнопений «На господи воззвах. Слава и ныне. Богородичен.» (№ 1) и «Слава и ныне. Богородичен» (№10) из Октоиха Сол. 619/644.

На рисунке 2 представлены фрагменты песнопений «На господи воззвах. Слава и ныне. Богородичен» (№ 1) и «Слава и ныне. Богородичен» (№10) из Октоиха Сол. 619/647, в которых совпадает мелодическая составляющая на знаменном и нотном уровнях, а тексты на старославянском языке разные. Сверху записан текст десятого песнопения, внизу – первого. Весь фрагмент за исключением последнего звука (7 знамен) входит в число внутригласовых инвариантов и встречается в текстах трех Октоихов 6 раз, а подчеркнутый фрагмент соответствует попевке «кулизма средняя», которая, как уже упоминалось выше, встречается 39

раз в 32 песнопениях Октоихов. Попевке в данном случае соответствует внутригласовый инвариант (5 знамен), встретившийся в трех Октоихах 26 раз.

Наряду с инвариантом, указанном на рисунке 2, «кулизма средняя» 11 раз представлена в первом гласе трех Октоихов в составе инварианта (e4f4g2) (e4f4e4d4) (c4d4c4H4) (A1), отличающегося заменой двух знамен одним составным . Кроме того, второе вхождение этой попевки в десятое песнопение не входит в множество внутригласовых инвариантов и выглядит следующим образом: (e4f4) (g2) (e4f4e4d4) (c4d4) (c4H4) (A1).

Фрагменты «кулизмы средней» содержатся в 7 инвариантах, среди которых, например, (e4f4) (g2) (e4f4), встречающийся в Октоихах 5 раз. И, наоборот, эта же попевка входит в состав 27 более длинных инвариантов, среди которых, кроме представленного на рисунке 2, фигурирует инвариант (d4H4c4H4) (A2c2) (H4H4A2) (c4H4) (c4d4) (e2.d4) (e4f4g2) (e4f4e4d4) (c4d4c4H4) (A1), входящий в Октоихи 4 раза.

Таблица 1. Сопоставление внутригласовых инвариантов разной длины с попевками

l	V_r	V_c	V_p	V_s	V_n	F_v
3	3	75	5	80	70	150
4	7	63	9	71	94	165
5	5	39	12	48	88	136
6	3	22	16	37	63	100
7	2	10	14	23	46	69
8	0	1	14	15	29	44
9	0	0	6	6	13	19
10	0	0	3	3	6	9
11	0	0	0	0	3	3
Σ	20	210	79	283	412	695

В таблице 1 представлено количество различных инвариантов (без учета частоты их встречаемости в текстах песнопений) длины от трех до 11 знамен (последний столбец F_v), и количество инвариантов, связанных с попевками. Здесь V_r – количество различных инвариантов, совпадающих с попевками, V_c – количество инвариантов, вложенных в попевки, V_p – количество инвариантов, содержащих попевки, V_s – количество инвариантов, связанных с попевками лишь в одном из перечисленных случаев (заметим, что в общем случае $V_s \le V_r + V_c + V_p$, т.к. один инвариант может совпадать с одной попевкой и быть вложенным или содержать другую попевку), $V_n = F_v - V_s$ – количество инвариантов, не связанных с попевками.

Анализ таблицы 1 показывает, что из 695 инвариантов первого гласа трех Октоихов 412 инвариантов формально не связаны с попевками отношением вложенности. Среди них 84 «нерасширяемых», 23 из которых заканчиваются четвертной нотой, 45 — половинной нотой, 16 — целой. Последние могут претендовать на роль вариантов попевочных структур. Например, внутригласовый инвариант (e4d4) (f2) (e2) (d2) (d4c4)(d4e4f2) (e1) отличается от «колчанца» всего одной вставкой и отсутствием каданса d1.

Внутригласовый инвариант (g4a4) (g4f4) (e4f4g4a4) встречается в песнопениях трех Октоихов 6 раз. Он нерасширяем, т.е. отсутствуют инварианты длины 4 и выше, в которые был бы вложен данный фрагмент. Этот же фрагмент обнаруживается и с помощью попарных сложностных разложений. Нотная составляющая инварианта вложена в попевки «дербица большая»: g4a4g4f4e4f4g4a4a4g2. и «дербица полная»: g4f4g4a4g4f4e4f4g4a4a4g2. Расширение влево и вправо фрагмента, соответствующего вхождению инварианта в песнопение «Слава» Октоиха Сол. 619/647 (№14), и сопоставление его с аналогичными фрагментами одноименных песнопений из Октоихов Сол. 618/644 и QI 188 демонстрирует совпадение первого звука при записи его разными знаменами, разную длительность (половинная с точкой, половинная и целая) последнего звука и вложенность попевки «дербица большая» в первый из фрагментов:

Сол. 619/647: (e1 ху)(g4a4 честь)(g4f4 и)(e4f4g4a4 сла)(a4g2. ва)
Сол. 618/644: (e1 ху)(g4a4 честь)(g4f4 и)(e4f4g4a4 сла)(a4g2 ва)
QI 188 : (e1 ху)(g4a4 честь)(g4f4 и)(e4f4g4a4 сла)(a4g1 ва)

При сравнении аналогичного фрагмента песнопения «Антифон второй» (Сол. 619/647, №15) с его одноименными аналогами наблюдается расширение инварианта влево до попевки «дербица полная», вставка слога в QI 188 и разная длительность последнего звука:

Сол. 619/647: (g4f4 си) (g4a4 мя)(g4f4 за)(e4f4g4a4ко)(a4g2. новъ)

Сол. 618/644: (g4f4 си) (g4a4 мя)(g4f4 за)(e4f4g4a4ко)(a4g2 новъ)

QI 188 : (g4f4 во)(v0121 а2знес)(g4a4мя)(g4f4за)(e4f4g4a4ко)(a4g1нъ)

Кроме того, можно видеть, что в песнопениях «Слава» из «Антифона 3» кроме различий в длительности последней ноты, два знамени заменены составным , поскольку распевается один слог текста, а не два:

Сол. 619/647: (g4f4 кло)(g4a4g4f4не)(e4f4g4a4ни) (a4g2. е)

Сол. 618/644: (g4f4 кло)(g4a4g4f4не)(e4f4g4a4ние)(a4g2 сла)

QI 188 : (g4f4 кло)(g4a4g4f4ня)(e4f4g4a4ни) (a4g2 е)

Столь незначительные отличия позволяют высказать предположение, что все перечисленные фрагменты являются вариантами попевки «дербица». Заметим, что знамя (дербица) встречается только в перечисленных фрагментах песнопений первого гласа трех Октоихов и совсем не встречается, например, в первом гласе Ирмология.

Каждый из повторяющихся фрагментов (мелодических оборотов), выявляемых с помощью сложностных разложений, сравнивается с каждой из 94 попевок в подборке Металлова. При этом выделяются случаи, когда фрагмент совпадает с одной из попевок, содержит попевку или является частью попевки. Из 2707 фрагментов (нерасширяемых повторов) длиной 10 или более нот, 378 выделенных фрагментов совпадают с попевками, 1277 содержат попевки, 704 являются частью какой-либо из попевок. Для 727 выделенных фрагментов связь с попевками не выявлена.

При увеличении порога r до 15 число выделенных фрагментов (компонентов разложений) сокращается до 625, из которых только 16 совпадают с попевками, 522 содержат попевки, 10 вложены в попевки и 85 длинных повторов не связаны с попевками.

В таблице 2 представлены результаты сравнения отдельных попевок с компонентами сложностных разложений, длина которых 10 нот или более. В первом столбце указан номер попевки из сборника Металлова В.М., во втором – название попевки и в третьем – ее нотное представление. В четвертом столбце (N_r) указано число компонентов, совпадающих с попевками, пятый столбец (N_c) –число компонентов, содержащих попевки, шестой столбец (N_p) – число случаев вложения компонента в попевку, седьмой столбец (F_t) – число песнопений, содержащих попевку, восьмой столбец (F_p) – общее число вхождений попевки в тексты трех Октоихов. В таблицу включены только попевки, встречающиеся не менее 6 раз в песнопениях первого гласа трех Октоихов.

Таблица 2. Сопоставление попевок с компонентами разложений

№	Название попевки	Попевка	N_r	N_c	N_p	F_t	F_p
5	Вознос последний	d2e4f4e4d4c4H4A1G1	0	55	0	12	12
11	Пригласка или киза малая	a2g4f4g2f2e1	0	61	0	17	25
12	Пригласка или киза средняя	f4g4a2g4f4g2f2e1	0	34	0	13	17
13	Пригласка большая	e4d4f4g4a2g4f4g2f2e1	21	10	0	8	9
18	Пригласка малая с тряскою	a8g8f4g4f4e4	0	13	0	15	16
25	Таганец малый	a2g4f4g2e1	0	27	0	15	15
26	Таганец средний	f4g4a2g4f4g2e1	0	27	0	13	13
27	Таганец большой	e4d4f4g4a2g4f4g2e1	0	23	0	11	11
31	Выплавка	c4H4c4d4c4H4c2d2e1	0	23	0	12	13
32	Переметка	c4d4e4d4e1	0	50	0	18	28
40	Возмер	e4d4e4f4g2f2e1	0	2	0	5	6
43	Хамила	e2d4e4d2	0	39	0	27	31
44	Паук	e4f4g2f2e2	0	23	0	15	17
59	Кулизма средняя	e4f4g2e4f4e4d4c4d4c4H4A1	14	609	0	32	39
64	Ометка малая	e2f4g4e2e2f4e4d2e2d1	23	22	0	10	10
68	Рожек светлый	e4d4e4f4g2f4e4f2e1	0	18	0	10	11

69	Задевец	c4H4c4d4e2f4e4d2e2d1	0	21	0	7	7
74	Колесо или шибок малый	e4f4d4e1d1	0	111	0	32	50
76	Колесо или шибок большой	d2e4f4g2e4f4d4e1d1	0	21	0	9	10
81	Долинка меньшая с ударкой и ломкой	c4e4e4d4d8c8H4c4H4A1G1	34	46	0	10	16
82	Долинка меньшая с качкой	d4e4e4d4d4H4c4H4A1G1	57	89	0	14	22
83	Долинка меньшая с качкой и ломкой	c4e4e4d4d4H4c4H4A1G1	6	13	0	7	8
85	Долинка средняя с ударкой и ломкой	d4H4c4e4e4d4d8c8H4c4H4A1G1	9	24	41	7	10
86	Долинка средняя с качкой	d4c4d4e4e4d4d4H4c4H4A1G1	37	48	108	12	16
93	Колчанец	e4d4f2e2d4c4d4e4f2e1d1	154	92	109	21	26
94	Пастела полная	e4f4g2f2e4d4f2e2d4c4d4e4f2e1d1	13	8	380	7	7

Наибольший интерес представляют столбцы N_c и N_r, т.е. компоненты, содержащие попевку или полностью с ней совпадающие. Обратим внимание, что поскольку при построении сложностных разложений каждое песнопение сравнивается с каждым, выполняется соотношение $\boldsymbol{N_c + N_r \leq C^2_{F_t}}$.

Поскольку среди попевок довольно много вложенных, то один компонент может содержать несколько попевок, например общий мелодический оборот песнопений «Слава» (№18) и «Слава и ныне» (№1) совпадает с попевкой «пригласка большая» и содержит попевки «пригласка или киза малая» (a2g4f4g2f2e1) и «пригласка или киза средняя» (f4g4a2g4f4g2f2e1):
18: (e4d4 сла)(f4g4 ва) (a2 и) (g4f4де) (g2f2 ржа) (e1 ва)
1: (-e4d4 то)(f4g4 рже)(a2 ство)(g4f4 днесь)(g2 бра)(f2 ти)(e1 е)

Как уже упоминалось выше, в песнопениях первого гласа трех Октоихов обнаружено 625 совпадающих мелодических оборотов, состоящих из 15 или более нот, из которых только 85 повторов не связаны с попевками из подборки Металлова. Некоторые мелодические обороты, формально не связанные с попевками, тем не менее обладают свойствами, сходными с попевочными структурами. Так, например, общий фрагмент песнопений «На стиховне. Богородичен» (№2) и «В неделю вечерня. Богородичен» (№20) заканчивается наиболее частым для попевок кадансом «палка - статья», соответствующим окончанию слов:
(e4f4g2 и)(f2 же)(e4d4 к те)(f2бе)(e2 при)(d4c4 бе)(d4e4f2 га)(e2 ю)(e2 щи)(d1 я)
(e4f4 те)(g2 бе)(f2воз)(e4d4 си)(-f2e2 я)(d4c4 вый)(d4e4f2 мир)(e2 про)(e2све)(d1ти)

То же самое наблюдаем для общих фрагментов песнопений «Стихиры восточные» (№8 и №9), отсутствующих в подборке Металлова, но соответствующих архетипу «какизы» [4]:
(e2от)(e2 де)(e4g4 ви)(e2 че)(f4e4 ска)(d2 чре)(e2)(d1 ва)
(e2 из)(e2ре)(e4g4 че)(e2нно)(f4e4му)(d2 тво)(e2 е)(d1 му)

Другие мелодические обороты, общие для разных песнопений, могут соответствовать распевам фиты (мелодиям без слов или длинным распевам одного слога текста [12]). Такие фрагменты видим, например, в песнопении «На литургии блаженна» из QI188 (№17) и «Стихира восточна» из Сол. 619/647 (№7):
17: (e2f2g1 мя)(f4e4)(f1)(e1)(d2.c4)(d4e4f2)(e1)
7: (f4e2)(f2g1) (f4e4)(f1)(e1)(d2.c4)(d4e4f2)(e1)

Общий фрагмент «Стихиры воскресна» (Сол. 619/647, №4) и «Стихира восточна» (Сол. 618/644, №9) заканчивается двумя четвертными «си» (H4H4) и формально не содержит ни одной попевки, однако в «Стихире восточна» распев знамени заканчивается нотой A1 и вместе с этим окончанием фрагмент содержит попевку «Долинка средняя с качкой и хамилой» (d4c4d4e4e4d4d4H4c4H4A2c2H4H4A1), а фрагмент песнопения «Стихиры воскресна» заканчивается A2, но поскольку записан тем же знаменем, его можно считать вариантом данной попевки:
9 : (e2 во)(d4c4спо)(d4e4 им)(e4d4 гла)(d4H4c4H4 го)(A2c2 лю)(H4H4A1ще)
4: (e4g4e2скре)(d4c4сше)(d4e4му)(e4d4из)(d4H4c4H4ме)(A2c2)(H4H4A2ртвыхъ)

4 Заключение

Общим свойством структурных единиц в различных языковых системах является их повторяемость в текстах конкретного языка, допускающая ограниченную вариативность. Основным типом структурных единиц знаменного пения, традиционно обсуждаемых в литературе, являются попевки – относительно короткие мелодические обороты, характерные особенности которых перечислены в разделе 2.1. Однако они составляют лишь часть полного спектра повторов разного типа, представленных в знаменных песнопениях. Некоторые типы повторов, не относящихся к категории попевок, уже систематизированы (таковы, например, лица и фиты знаменного распева [12].

В данной работе обсуждаются типы структур, выделяемые формальными методами непосредственно из текстов певческих рукописей, а именно внутригласовые инварианты и достаточно длинные повторяющиеся фрагменты из разных песнопений. Для формирования множества таких структур используются эффективные методы, основанные на вычислении *L*-граммных спектров и построении сложностных разложений.

Проведено сопоставление формально выделяемых единиц знаменного распева (повторов) с множеством попевок из авторского сборника В.М. Металлова. Выделены случаи, когда повтор совпадает с одной из попевок, вложен в попевку или содержит её в себе целиком. Показано, что порядка (70 – 80) % достаточно длинных повторов так или иначе связаны с попевками, т.е. в них реализуется один из указанных выше типов связи. В знаменных песнопениях присутствуют и достаточно длинные повторы с «непопевочной» структурой, например, серии стопиц в речитативных участках мелодии. Однако среди повторов, формально не связанных с попевками из подборки Металлова, некоторые имеют ярко выраженную попевочную структуру. Отсутствие у них явной взаимосвязи с подборкой Металлова можно объяснить ограниченностью ее объема. Возможна также ситуация, когда взаимосвязь имеется, но не обнаружена, поскольку представлена в завуалированной форме (например, в виде несовершенного повтора). Таким образом, формальные методы, использующие реальные певческие книги, помогают обогащать своды попевок, изложенных в теоретических руководствах.

5 Благодарности

Авторы выражают благодарность Титковой Татьяне Николаевне за многолетнее сотрудничество в области анализа и дешифровки древнерусских знаменных песнопений.

Работа выполнена при поддержке программы Фундаментальных Научных исследований РАН, проект № FWNF-2022-0015.

Литература

[1] Бражников М. В. Древнерусская теория музыки. По рукописным материалам XV–XVIII веков. Л.: Музыка, 1972. 423 с.

[2] Бахмутова И. В., Гусев В. Д., Титкова Т. Н. Компьютерный анализ древнерусских двознаменников: многозначность соответствия «знамя – нота» и «нота - знамя» // Искусственный интеллект и экспертные системы (Вычислительные системы), вып. 157, 1996, с. 68-100.

[3] Бахмутова И.В., Гусев В.Д., Мирошниченко Л.А. Количественные оценки допустимой вариативности в задаче нотолинейной реконструкции знаменных песнопений // Проблемы музыкальной науки / Music Scholarship. 2022, № 4 , С. 98-106. DOI: 10.17674/2782-3601.2022.4.098-106.

[4] Кручинина А.Н. Попевка знаменного роспева в русской музыкальной теории XVII века. // Певческое наследие Древней Руси: история, теория, СПб:Ut, 2002, С. 46-150.

[5] Металлов В.М Осмогласие знаменного распева (сборник нотолинейных попевок), М., 1899, 50 с.

[6] Aho A.V., Corasick M.J. Efficient string matching: an aid to bibliographic search Communications of the ACM. 1975, Vol. 18 (6): pp. 333-340.

[7] Бахмутова И.В., Гусев В.Д., Титкова Т.Н. Использование билингв типа «знамя–нота» для выявления инвариантных структурных единиц знаменного распева // Проблемы музыкальной науки. 2015, № 2 (19). – С. 5–11.

[8] Gusev V.D., Bakhmutova I.V., Miroshnichenko L.A., Titkova T.N. Possible approaches to deciphering Russian ancient Znamenny chant // Journal of New Music Research, 2020, Volume 49, Issue 3, Pages 298-306. doi:10.1080/09298215.2020.1762666.

[9] Гусев В.Д. Характеристики символьных последовательностей // Машинные методы обнаружения закономерностей. Вычислительные системы, вып. 88. Новосибирск, 1981, С. 112–123.

[10] Liubov A. Miroshnichenko; Irina V. Bakhmutova; Tatiana N. Titkova; Vladimir D. Gusev Complexity Decompositions in the Problems of Analysis of Ancient Russian Ecclesiastical Chants // 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), DOI: 10.1109/SIBIRCON48586.2019.8958232.

[11] Гусев В.Д., Мирошниченко Л.А. Использование сложностных разложений в задачах анализа символьных последовательностей // Доклады 8-й Международной конференции "Интеллектуализация обработки информации" (ИОИ-2010), Кипр, Пафос, 17-24 октября 2010, 469-472.

[12] Бражников М. Лица и фиты знаменного распева. Ленинград, «Музыка», 1984, 304с.

Знания-Онтологии-Теории (ЗОНТ-23)

Применение методов машинного обучения и ГИС для автоматической агроэкологической типизации земель

Павлова А.И., Осипов А.Л.

Новосибирский государственный университет экономики и управления, ул. Каменская, д. 56, г. Новосибирск, 630039, Россия

annstab@mail.ru, alosip@mail.ru

Аннотация. Работа посвящена автоматической классификации (агроэкологической типизации) сельскохозяйственных земель с применением ГИС и методов машинного обучения. Автоматическая классификация сельскохозяйственных земель реализуется с помощью ГИС (геоинформационных, географических информационных) систем, данных дистанционного зондирования Земли и машинного обучения. Суть методики автоматической классификации сельскохозяйственных земель заключается в создании цифровой пространственной модели землепользования, характеристике операционно-территориальных единиц (ОТЕ), создании модели машинного обучения и ее тестировании, отображении результатов классификации в виде тематической электронной карты типов земель. Исследования выполнены на примере землепользования ЗАО «Мирный» Коченевского района Новосибирской области, расположенного на территории Приобского плато. При создании цифровой пространственной модели землепользования использованы картографические (топографическая карта М 1:25000, почвенная карта-схема), информационно-справочные, космические снимки высокого пространственного разрешения ALOS PALSAR 2, Sentinel-2 A, результаты полевого обследования территории исследований. Использование векторной сетки ОТЕ позволило существенного сократить время, необходимое на создание модели машинного обучения, ее тестирование и составление электронной карты агроэкологической типизации земель. В результате исследований выполнена оценка точности классификации земель с помощью методов машинного обучения: наивный байесовский классификатор (Gaussian Naïve Bayes, GNB), метод ближайших соседей (k-nearest neighbors), деревья решений (Decision Tree, DT), метод случайного леса (Random Forest, RF), машина опорных векторов (Support Vector Machine, SVM), методы обучения многослойных нейронных сетей Adam и стохастического градиентного спуска SGD (Stochastic Gradient Descent). Результаты оценки точности показали, что ансамблевый метод машинного обучения и метод деревьев решений имеют более высокие показатели точности на обучающей и валидационных выборках данных.RF имеет более высокие показатели. С помощью разработанной методики автоматической агроэкологической типизации сельскохозяйственных земель на территорию хозяйства была составлена электронная карта, что позволило оценить пространственное распределение земель по агроэкологическим типам земель.

Ключевые слова: агроэкологическая типизация земель, ГИС, данные дистанционного зондирования, рельеф, морфометрия, классификация, машинное обучение

1 Введение

Результаты агроэкологической оценки и типизации сельскохозяйственных земель (классификации) востребованы при разработке адаптивно-ландшафтных систем земледелия, проектирования и организации севооборотов [7,10]. Для геоинформационного обеспечения типизации сельскохозяйственных земель необходимо создание пространственной базы данных с базовыми и

тематическими информационными слоями (агроэкологической типизации, морфометрические карты рельефа, внутрихозяйственного землеустройства, почвенная и др.). Применение традиционной методики составления карт агроэкологической типизации на основе оверлейного анализа информационных слоев носит субъективный характер установления пространственных границ групп, подгрупп или типов земель. Возникающие сложности визуального анализа большого количества информационных слоев при комплексной оценке сельскохозяйственных земель могут быть решены путем применения современных методов автоматической классификации данных.

В связи с этим актуальны исследования в области классификации сельскохозяйственных земель с применением геоинформационных систем (ГИС) и методов машинного обучения, позволяющие автоматизировать трудоемкий процесс оценки и картографирования сельскохозяйственных земель.

2 Теоретические проблемы

2.1 Постановка задачи

К числу значимых природных условий, определяющих функционирование природных ландшафтов и агроландшафтов относят рельеф, литологию, климат, влияние грунтовых вод, растительность, почвенный покров. При этом ведущую роль отводят рельефу. Рельеф оказывает непосредственное влияние на развитие почвообразовательных процессов путем неравномерного распределения солнечной радиации, жидких и твердых осадков, потоков энергии по поверхности [4]. Геоморфологические условия тесно связаны с особенностями структуры землепользования и пространственными характеристиками агроэкологических типов земель.

Согласно традиционной методике при оценке геоморфологических условий сельскохозяйственных земель часто применяют качественное описание рельефа в процессе картографирования и описания его форм и типов. Более современный подход количественного описания рельефа позволяет вычислить морфометрические величины (характеристики, показатели). Количественные показатели рельефа, описывающие его морфометрию необходимы при решении широкого научных и практических задач, связанных с прогнозным почвенным картографированием , изучением негативных экзогенных процессов рельефообразования и др. Получаемые в процессе геоморфометрического анализа электронные карты служат информационной основой для автоматической классификации форм и элементов рельефа [11,15-16,19,21,26]. Примером может служить работа [9], посвященная разработке автоматической классификации морфокомплексов рельефа на основе скользящего окна. При этом необходимо подобрать размер скользящего окна и отдельные морфометрические показатели.

При оценке сельскохозяйственных земель среди морфометрических показателей внимание уделяется крутизне и экспозиции склонов, горизонтальному и вертикальному расчленению рельефа. Однако современное развитие методов геоморфометрии и геинформационного анализа способствует расширению перечня морфометрических показателей рельефа, характеризующих особенности топографической поверхности. Существует необходимость в разработке методических подходов комплексной количественной оценки сельскохозяйственных земель на основе данных дистанционного зондирования Земли (ДДЗЗ) и методов машинного обучения.

Методы машинного обучения успешно применяют для составления карт земных покровов и мониторинга земель. Процесс классификации с привлечением ДДЗЗ реализуется созданием большого количества информационных растровых слоев. В числе таких слоев применяют вегетационные индексы. В работах [14,20,22,23,25] использован нормализованный разностный вегетационный индекс (NDVI), почвенные индексы (SAVI, OSAVI, TSAVI, PVI и другие), нормализованный индекс влажности (NDWI) для классификации почвенно-растительного покрова. Количество информационных слоев может достигать от нескольких десятков и более. Поэтому для практической реализации методики автоматической классификации земель на основе растровых моделей требуется значительно больший объем памяти компьютера и вычислительных ресурсов в сравнении с векторными моделями ГИС.

Научная новизна исследований состоит в разработке методического подхода автоматической агроэкологической типизации земель на основе комплексной оценки геоморфологических, почвенно-экологических условий территории и спектральных характеристик объектов земной поверхности.

Цель работы – агроэкологическая типизация сельскохозяйственных земель с применением ГИС и методов машинного обучения.

2.2 Материалы исследований

Исследования выполнены на примере хозяйства ЗАО «Мирный», расположенного в центральной части Коченевского района Новосибирской области (54,84° с.ш.-55,06° с.ш., 81,96° в.д.-82,26° в.д.) (рис.1). В хозяйстве имеется населенный пункт Мирный, удаленный на 5 км от административного центра Коченево и в 80 км от г. Новосибирска. Почвенный покров хозяйства отличается разнообразием. Территория исследований относится на границе перехода дренированной лесостепи Приобья к плоской заболоченной лесостепи. Восточная граница хозяйства начинается почти на самой высокой части Приобского увала у районного центра Коченево, в направлении на запад рельеф заметно понижается и переходит в заболоченную равнину, характерную для Барабинской низменности.

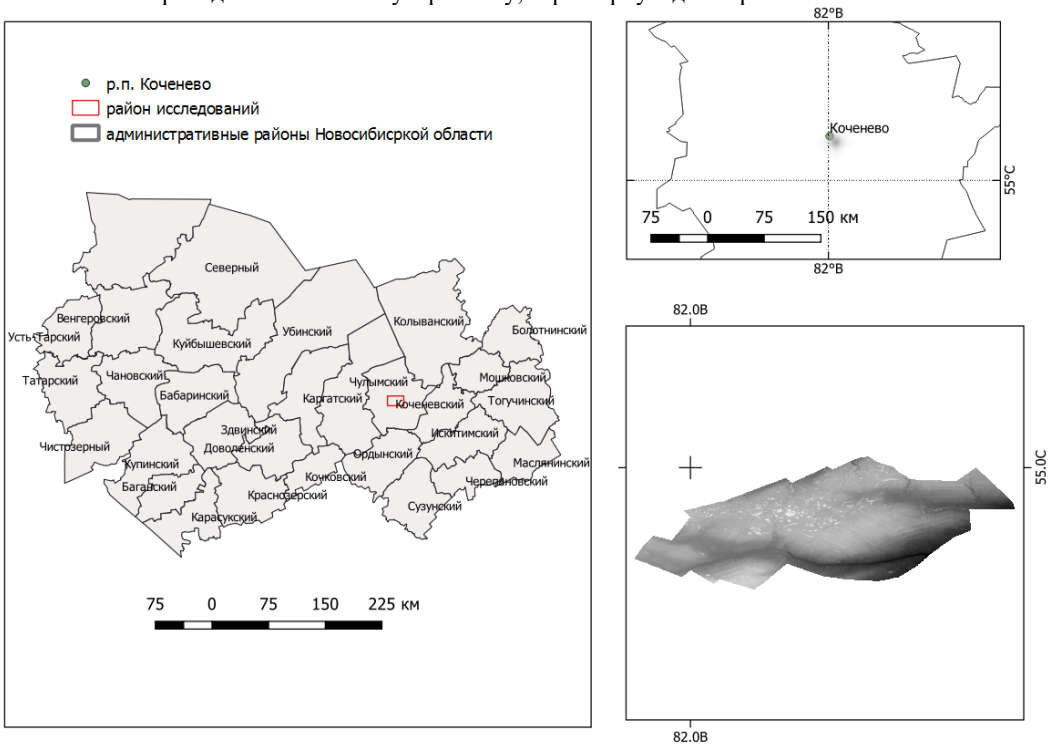


Рис. 1. Территория исследований ЗАО Мирный Коченевского района Новосибирской области

При создании цифровой пространственной модели землепользования использованы следующие материалы: топографическая карта (М 1:25000), Sentinel-2A, результаты полевого почвенного обследования и спутниковых измерений, почвенная карта (М 1:25000), космические снимки ALOS PALSAR-2, ALOS DSM (Digital Durface Model Японского аэрокосмического агентства Japan Aerospace Exploration Agency (JAXA), размер ячейки растрового изображения составляет 1 угловую секунду, что соответствует примерно 30 м. Космические снимки имеют пространственную привязку к эллипсоиду WGS 1984. Для исследований также использованы космические снимки интерферометрической съемки ALOS PALSAR-2 (Япония) с разрешением ячейки растровой модели 10 м/пиксел. Японский космический спутник JERS-1/SAR, оснащенный датчиком SAR L-диапазона был запущен в эксплуатацию в 1992 г. Позднее в 2006 г. введен в использование спутник ALOS/PALSAR и в 2014 гг. ALOS PALSAR-2 [1]. Цифровые модели рельефа имеют артефакты и для устранения искажений выполнена их коррекция [6].

2.3 Методы исследований

Методика автоматической классификации сельскохозяйственных земель заключается в создании цифровой пространственной модели землепользования, формировании векторной сетки операционно-территориальных единиц (ОТЕ), создании модели машинного обучения и ее тестировании, отображении результатов классификации в виде тематической электронной карты типов земель. Цифровая пространственная модель землепользования содержит атрибутивную и картографические составляющие (рельеф, агроклиматические параметры, почвы, сведения об использовании земель). Векторная модель данных для автоматической классификации сельскохозяйственных земель представляет собой сетку квадратов ОТЕ (размер стороны квадрата равен 2,5 м). Совокупность ОТЕ представляется двумерной матрицей «объекты-признаки», включающей морфометрические показатели, почвенно-экологические индексы, вегетационные индексы.

В проведенных исследованиях в процессе комплексной геоморфологической оценке сельскохозяйственных земель вычислены фундаментальные и комбинированные морфометрические показатели. Данные показатели согласно теории дифференциальной геометрии и теории топографической поверхности представляются функцией двух переменных плановых координат точки на топографической поверхности, описывающей ее свойства [8,15-16,24]. Комбинированные морфометрические показатели включают локальные и нелокальные переменные, и участвуют при характеристики геометрических особенностей территории и относительного положения некоторой точки на топографической поверхности [8,15-16,24]. В работе вычислены 31 показатель рельефа, в числе которых:

- топографический индекс влажности (TWI Topographic Wetness Index);
- топографический индекс расчлененности (устойчивости местности) TRI (Topographic Ruggedness Index);
- индекс оценки степени эрозионных процессов LSF (Length Steepness Factor);
- топографический индекс отношения уклона и длины склонов Slope-Length;
- топографический индекс Mid-Slope Position;
- составной топографический индекс TCI (Topographic Compound index).

Индекс оценки степени эрозионных процессов LSF (Length Steepness Factor) входит в качестве основного показателя универсального уравнения потерь почвы USLE (Universal Soil Loss Equation) и более современной версии для оценки RUSLE (Revised Universal Soil Loss Equation) и модифицированную версию MUSLE (Modified Universal Soil Loss Equation) [12].

Абсолютная высота согласно работе [8] не относится к группе локальных, региональных и комбинированных показателей, однако используется при вычислениях всех количественных параметров рельефа. Поэтому в работе при комплексной оценке рельефа территории хозяйства вычислены: абсолютная высота водосборного бассейна; глубина вреза речных долин; амплитуда высот; нормализованная высота; стандартизированная высота.

Почвенно-экологическая оценка земель реализована путем оценки агроклиматических и почвенных условий на основе методики [3].

Для классификации сельскохозяйственных земель использованы методы машинного обучения: наивный байесовский классификатор (Gaussian Naïve Bayes, GNB), метод ближайших соседей (k-nearest neighbors), деревья решений (Decision Tree, DT), метод случайного леса (Random Forest, RF) [13], машина опорных векторов (Support Vector Machine, SVM) [2], методы обучения многослойных нейронных сетей Adam [18] и стохастического градиентного спуска SGD (Stochastic Gradient Descent) [5]. При построении модели машинного обучения (ММО) реализованы процедуры наличия и заполнения пропущенных значений сплайн-методом, мин-макс нормализация данных.

2.4 Оценка точности классификаторов

Для машинного обучения исходный набор данных был разделен на три выборки: обучающую (70%), тестовую (15%) и валидационную (15%) случайным образом. Процедура оценки точности классификации сельскохозяйственных земель является важной частью валидации карт агроэкологической типизации земель. Полученные карты агроэкологической классификации земель могут служить основой для успешного планирования и рационального управления земельными ресурсами. Для оценки точности ММО использованы показатели: точность (Precision), полнота (recall), F-мера (F1 score), доля ложноположительных примеров (False Positive Rate, FPR), доля отрицательных примеров (False Negative Rate, FNR). Дополнительно при анализе результатов построения ММО служил визуальный анализ кривой ROC (receiver operating characteristic) и площадь под кривой ROC (Areas

under the curve, AUC). F-меру часто используют при оценке методов машинного обучения, в особенности для случае когда распределение исходных данных в обучающем наборе неравномерное [5, 17]. Квадратный корень средней квадратической ошибки (Root Mean Square Error, RMSE) является одним из основных показателей эффективности классификатора. Данный показатель использован для оценки правильности работы методов машинного обучения (классификаторов).

Результаты исследований

Результаты оценки точности моделей машинного обучения для обучающей и валидационной выборок данных приведены в таблице 1. Более высокие показатели точности получены для ансамблевого метода случайного леса RF и метода деревьев решений DT (для кросс-валидации изменяются от 91,2 до 95,0%). Значение F1 score, отражающее среднее гармоническое точности (Precisioin) и полноты (Recall) выше для RF на 2,9%, а значение RMSE ниже на 0,201.

Для методов KNN, SVM_radial, SVM_poly, Adam, SGD точность классификации сельскохозяйственных земель составила от 84,1 до 85,7%. Коэффициент каппа составил от 82,3 до 85,3%. Для метода SVM точность классификации сильно зависит от выбора разделяющего ядра и других гиперпараметров обучения. При использовании гауссова радиально-базисного ядра общая точность классификации и показатель F1 score существенно выше в сравнении с линейным или полиномиальным ядром. Градиентный метод Adam характеризуется более высокими показателями точности и количеством верно классифицированных примеров в сравнении с рассматриваемыми методами. Однако время, затраченное на обучение ММО существенно выше в сравнении с методами KNN и SGD. Для повышения точности классификации большое значение имеет оптимальный выбор гиперпараметров ММО. В проведенных исследованиях использованы следующие гиперпараметры моделей машинного обучения, вычисленные с помощью метода Random Reach (Таблица 2).

Таблица 1. Результаты оценки точности классификации земель

Алгоритм	Время обучения, сек	Precision train, %	Precision val,%	F1 score train, %	F1 score val, %	Kappa train,%	Kappa val,%	RMSE train	RMSE val
RF	3,1	99,9	95,0	99,7	95,3	100,0	94,6	0,001	0,124
DT	6,7	99,9	92,6	98,5	92,4	99,9	91,2	0,005	0,325
GNB	2,5	77,5	76,4	76,5	74,8	77,1	73,3	2,102	2,146
KNN	20,6	90,2	85,3	89,1	84,2	88,7	83,5	0,728	0,779
SVM_linear	315,8	77,5	76,9	76,8	75,4	75,1	73,2	2,130	2,418
SVM_radial	115,48	84,1	84,5	85,3	84,0	85,2	84,4	1,152	1,167
SVM_poly	416,48	83,7	82,5	84,8	83,6	84,3	83,5	1,235	1,380
Adam	454,18	86,5	85,7	89,4	85,1	86,2	85,3	0,424	0,525
SGD	58,19	84,4	83,3	82,2	83,5	81,9	82,3	0,907	0,951

Таблица 2. Гиперпараметры моделей машинного обучения

Модель	Гиперпараметры
Ансамблевый метод случайного леса (RF)	максимальная глубина дерева решений 20 количество деревьев решений 100 функция для измерения качества разбиения Gini минимальное количество экземпляров узле 2 минимальное количество экземпляров в листе 1
Метод деревьев решений (DT)	функция для измерения качества разбиения Gini минимальное количество экземпляров узле 2 минимальное количество экземпляров в листе 1
Наивный байесовский классификатор (GNB)	веса, применяемы к выборам приняты равным 1; вероятность появления –функция Гаусса
Метод ближайших соседей (KNN)	количество соседей 5 весовые коэффициенты относительно евклидова расстояния
Машина опорных векторов	параметр регуляризации 1,0 линейное ядро классификации;

	полиномиальное со степенью 3; гауссово ядро с радиальной базовой функцией
Многослойная нейронная сеть	с двумя скрытыми слоями, количество нейронов на скрытых слоях 20, 30; функция активации на скрытых слоях ReLu; количество циклов обучения 30; коэффициент скорости обучения 0,001. Метод обучения Adam и SGD

Невысокие значения показателей точности получены при использовании наивного байесовского классификатора и машины опорных векторов с линейным ядром классификации. Точность классификации для кросс-валидации ниже 18,9 и 18,1% соответственно в сравнении с методом RF. Метод GNB характеризуется высокой скоростью обучения с другими методами, но имеет низкие показатели точности классификации сельскохозяйственных земель.

Заключение

Выполненный анализ методов машинного обучения показал, что ансамблевый метод машинного обучения и метод деревьев решений имеют более высокие показатели точности на обучающей и валидационных выборках данных. Оптимальный выбор гиперпараметров машинного обучения оказывает существенное влияние на результаты классификации сельскохозяйственных земель. Для метода машина опорных векторов оптимальным разделяющим ядром классификации является гауссово радиально-базисное ядро. Высокие показатели точности классификации земель получены при использовании многослойной нейронной сети, обученной методом Adam. Разработанная методика автоматической классификации сельскохозяйственных земель позволила в интеграции с помощью ГИС, ДДЗЗ и машинного обучения составить электронную карту агроэкологических типов земель на примере хозяйства ЗАО «Мирный» Коченевского района Новосибирской области.

Литература

[1] ALOS PALSAR-2. 2023. URL: https://www.eorc.jaxa.jp/ALOS-2/en/about/palsar2.htm

[2] Вапник В. Н. Восстановление зависимостей по эмпирическим данным. М, Наука, 448 с, 1979.

[3] Карманов И.И. Методика и технология почвенно-экологической оценки и бонитировки почв для сельскохозяйственных культур. М.: ВАСХНИЛ, 114 с.,1990.

[4] Ласточкин А.Н. Системно-морфологическон основание наук о Земле (геотопология, структурная география и общая теория геосистем). Спб: НИИХ СПб, 762 с, 2002.

[5] Миронов А. Машинное обучение. М-Макс-пресс, Ч.1, 100 с., 2018.

[6] Павлова А.И. Анализ методов интерполирования для создания цифровых моделей рельефа // *Автометрия,* Т. 53, № 2, С. 86–94, 2017.

[7] Самовалова И.А., Мудрых Н.М., Каменский Н.Ю., Лобанова Ю.А. Агроэкологическая типизация земель как основа совершенствования систем севооборотов и удобрений. *Вестник Алтайского государственного аграрного университета*, № 5(103), С.45–50, 2013.

[8] Флоринский И.В. Иллюстрированное введение в геоморфометрию. *Электронное научное издание альманах Пространство и время*, Т.11, вып.1, 2016.

[9] Чупина Д.А., Зольников И.Д. Геоинформационное картографирование форм и типов рельефа на основе морфометрического анализа // Геодезия и картография. 2016. № 6. С. 35-43.

[10] Чурсин А.И., Крючкова Н.А. Агроэкологическая оценка земель в системе рационального природопользования, Пенза: ПГУАС, 192 с., 2016.

[11] Amelia V., Sinaga S., Bhermana A. Agroecological management at sloping land areas using land resources evaluation approach to achieve sustainable agricultural development (a case study of Gunung Mas Regency, Central Kalimantan Province, Indonesia). *IOP Conf. Series: Earth and Environmental Science*, vol. 782. article 032061. 2021. URL: https://iopscience.iop.org/article/10.1088/1755-1315/782/3/032061 (дата обращения 16.03.2023).

[12] Benavidez R., Jackson B., Maxwell D., Norton K.: A review of the (Revised) Universal Soil Loss Equation ((R)USLE): with a view to increasing its global applicability and improving soil loss estimates. *Hydrology and Earth System Sciences,* no. 22, p. 6059–6086, 2018.

[13] Breiman L. Random Forests. *Machine Learning,* vol. 45 (1), p. 5-32, 2001.

[14] Erinjery J. J., Singh, M.,Kent R. Mapping and assessment of vegetation types in the tropical rainforests of the western ghats using multispectral Sentinel-2 and SAR Sentinel-1 satellite imagery. *Remote Sensing of Environment,* vol. 216, p. 345–354, 2018.

[15] Florinsky I. V. Digital Terrain Analysis in Soil Science and Geology. 2nd ed. Amsterdam: *Elsevier Academic Press*, 486 p., 2016.

[16] Florinsky I. Geomorphometry on the surface of a triaxial ellipsoid: Towards the solution of the problem. *International Journal of Geographical Information Science*, vol. 32(8), p.1558-1571, 2018.

[17] Hastie T., Tibshirani R., Friedman J. The elements of statistical learning. Data mining, interference, and prediction. *Springer*, 764 p., 2017.

[18] Kingma D.P., Ba J.L. Adam: a method for stochastic optimization. *Publised as a conference at ICLR 2015*. URL: https://arxiv.org/pdf/1412.6980.pdf (дата обращения 1.06.2023). 2015.

[19] Lin S., Chen N., He Z. Automatic Landform Recognition from the Perspective of Watershed Spatial Structure Based on Digital Elevation Models. *Remote Sensing*, vol. 13, no 19, article 3926. 2021. URL:https://doi.org/10.3390/rs13193926 (дата обращения 10.05.2023).

[20] Ma L., Liuc Y., Zhang X.,Ye Y., Yin G., Johnson B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS J. Photogramm. Remote Sensing,* no. 152, p. 166–177, 2019.

[21] Mashimbye Z.E., Loggenberg K. A Scoping Review of Landform Classification Using Geospatial Methods. *Geomatics*, no. 3, p. 93–114, 2023.

[22] Pelletier C., Webb G. I., Petitjean F. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing,* no. 11, article 523, 2019.

[23] Rußwurm M., Körner M. Multi-temporal land cover classification with long short-term memory neural networks. *Int. Arch. Photogramm. Remote Sens. Spat.Inf. Sci.,* no. 42, article 551, 2017.

[24] Shary P. A., Sharaya L. S., Mitusov A. V. Fundamental quantitative methods of land surface analysis // Geoderma, no. 107, p.1-32, 2002.

[25] Thanh Noi, P.; Kappas, M. Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery. *Sensors*, no. 18, article 18, 2018.

[26] Xiong L., Li S., Tang G., Strobl J. Geomorphometry and terrain analysis: data, methods, platforms and application. *Earth-Science Reviews,* no. 233. article 104191. 2022. URL: https://doi.org/10.1016/j.earscirev.2022.104191 (дата обращения: 1.03.2023).

Метод распознавания эмоциональной окраски текстов, основанный на сочетаниях слов

Пахомов И. В.

Институт математики им. С.Л. Соболева СО РАН

iv-pahomov@yandex.ru

Аннотация. *В статье рассмотрен метод распознавания эмоциональной окраски текстов. Метод реализован с помощью библиотеки для построения нейронных сетей. Этот метод производит классификацию текстовых сообщений по категориям эмоции, используя в качестве входных данных разность векторов, полученных при векторизации входящего текста. Затем нейронная сеть относит сообщение к какой-то эмоциональной категории. Приведено описание модели и её топология. Основная особенность данной модели от других популярных моделей в том, что она проводит анализ по сочетаниям слов, то есть, проводит анализ по контексту, при этом, не требует больших словарей для обучения. Использование итеративного обучения позволяет дообучать модель в процессе использования.*

Ключевые слова: нейронная сеть, распознавание текста, семантические технологии, обработка естественного языка, теория предметной области, извлечение знаний, порождение знаний

1 Введение

Одним из популярных направлений в развитии технологий сфере искусственного интеллекта на сегодняшний день стоит задача распознавания текстовых диалогов. За последние несколько лет большую популярность приобрели чат-боты с применением искусственного интеллекта, используемые в поисковых системах, в некоторых мессенджерах и платформах для группового общения. Данные боты способны заменить живого собеседника в любой бытовой или профессиональной сфере. Например, они могут предоставить профессиональную консультацию по многим вопросам. Несмотря на то, что во многих узкоспециализированных направлениях процент точных ответов ещё не достиг приемлемого уровня, большой рост популярности подобных технологий и лёгкость в использовании делает подобные боты незаменимым помощником для большого количества людей[1].

Одним из вариантов решения данной проблемы является подбор иных категорий для обучения алгоритмов таких программных систем. Для определения смысла текстового сообщения можно использовать не только прямой этимологии ключевых слов в сообщении, но и установление эмоционального смысла в сообщении.

Для корректной работы чат-ботов, как и любых других систем, использующих смысловое распознавание текста, необходимо, чтобы бот корректно воспринимал входную информацию, для чего нужно не только восприятие смысла (значения) вводимых слов, но и понимал суть выражения в целом. Для понимания контекста сообщения желательно оценивать его эмоциональную окраску.

Выражение эмоций в записанных текстах на естественном языке играет важную роль в имитации естественного общения. Примеры подобных задач – отслеживание эмоций пользователей в отношении обзоров, событий или политики, выраженных в виде

комментариев в сети или новостях, управление взаимоотношениями с клиентами – всё это использует эмоциональную информацию[2,3].

Использование моделей распознавания эмоций под более гибкие языки также накладывает дополнительные сложности для применения простых моделей, использующих обычное семантическое определение отдельных слов.

2 Теоретические проблемы

2.1 Постановка задачи

Основные алгоритмы распознавания смысла текстовых сообщений и эмоциональной нагрузки уже определены, теперь главной проблемой остаётся лишь точность распознавания. Для увеличения точности распознавания эмоциональной окраски текстов необходимо подобрать другой классификатор, например, такой, который является модификацией существующих методов.

Ещё одной проблемой является поиск или формирование обучающей выборки. Задачи с применением обучающей выборки, являются более удобными и лёгкими для разработки и внедрения, но для полноценной работы таких систем необходима сама обучающая выборка значительного объёма. В большинстве случаев она представляет из себя огромный корпус текстов, в котором дополнительный объём также занимают тэги (разметка). Создание выборки удовлетворяющего объёма, покрывающего все необходимые варианты классификации, их возможные вариации и выбросы, является очень объёмной задачей, сильно затягивающий процесс разработки.

Несмотря на то, что существуют выборки находящиеся в открытом доступе, они не соответствуют полному перечню вариаций классов. Тем более, обучение модели по ограниченной выборке представляет собой малодинамическую технологию, в которой имеется необходимость переобучать заново модель при возникновении новых примеров.

Модели, основанные на обучении на примере конкретных токенов, тоже необходимо обучать, однако их обучение, при масштабировании на новые возникающие примеры, проходит легче, то есть, её достаточно дообучить, а не переобучать заново.

Наиболее успешные и часто применимые ныне модели, такие как GPT, например, работающие с распознаванием текста по смыслу, используют токенизацию, то есть, разбиение текста на мелкие составляющие, и уже работают с этими токенами. Для таких моделей, например, можно использовать, например, словари, или сочетания символов[13-15].

В данной работе предложен метод, распознающий эмоциональную окраску текста с помощью сочетания слов, а не отдельных слов. В данном случае, на вход такой модели подаётся разность векторов между двумя словами. Это поспособствует, распознаванию смысла не по отдельным словам, а по сочетаниям слов. Данная модель будет больше по объёму и сложности программной реализации, но при этом должна быть более точной, в силу того, что в ней для определения результата будет использоваться контекст, то есть, окружение одного слова другими словами. Для обучения модели используется итеративный метод обучения[16].

2.2 Разработка модели

Задачей исследования в рамках квалификационной работы является задача эмоционального распознавания текстов, что, по сути, представляет из себя частный случай системы смыслового распознавания текстов.

Общая схема модели следующая[4]:

1. Предварительная обработка входного текста для удобства дальнейшей работы модели с этим текстом
2. Обработка текста моделью для выяснения уникальных признаков сообщения.
3. Классификация текста с помощью различных методов распознавания, классификации и иных методов машинного обучения, формирующая в результате решение.

Входными данными для разработанной модели является текстовое сообщение. Входные данные должны пройти предварительную обработку. Целью такого процесса является придание тексту более формализованной формы, удобной для восприятия его программными средствами.

Предварительная обработка текстов.

Обычно используются следующие приемы нормализации текстов[5]:

1. Удаление шума и стоп-слов. Под шумом понимаются слова, не несущие никакой полезной информации, стоп-слова.

2. Стемминг – нахождение основной формы слова для заданного исходного слова, при этом основа слова не всегда может совпадать с морфологическим корнем слова. Нормальная форма исключает склонение слова, множественную форму, особенности устной речи, текстовые ошибки (опечатки) и сленговые искажения.

3. Лемматизация – преобразование слов в лемму, то есть в их первоначальную словарную форму.

4. Приведение регистра – преобразование всех символов к верхнему или нижнему регистру, в каком виде программному средству будет удобнее.

После нормализации текста все слова, либо их части или словосочетания, необходимо преобразовать в векторы. Слова-векторы – численные представления слов, сохраняющие семантическую связь между ними.

Для векторизации текста можно использовать уже существующие методы.

1. One-hot encoding

Пусть число различных слов равно K. Сопоставим слову с номером i вектор длины K, в котором i-тая координата равна единице, а все остальные — нулям. Недостатком этого метода является то, что по векторным представлениям нельзя судить о схожести смысла слов. Также вектора имеют очень большой размер, из-за чего их неэффективно хранить в памяти.

2. word2vec

word2vec принимает на вход большой текстовый корпус и сопоставляет каждому слову вектор. Сначала он создает словарь, а затем вычисляет векторное представление слов. Векторное представление основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами, то есть имеющие схожий смысл, в векторном представлении имеют высокое косинусное сходство.

3. fastText

Недостатком word2vec является то, что с его помощью не могут быть представлены слова, не встречающиеся в обучающей выборке. fastText решает эту проблему с помощью N-грамм символов. Модель fastText строит векторные представления N-грамм, а векторным представлением слова является сумма векторных представлений всех его N-грамм. Части слов с большой вероятностью встречаются и в других словах, что позволяет выдавать векторные представления и для редких слов[6,7].

Для разработки в рамках квалификационной работы была выбрана модель word2vec, тем более, что one-hot encoding и так в неё включен, а fastText является развитием word2vec. В данном случае, на вход модели подаётся разность векторов между двумя словами. Это поспособствует, в частности, разбиению смысла не по отдельным словам, а по сочетаниям слов.

2.3 Структура модели

От классических моделей распознавания текста, представленная модель отличается тем, что между частью, в которой происходит токенизация (и дальнейшая векторизация) входных данных, и непосредственно моделью распознавания первичных входных векторов, формируются новые векторы. Эти вторичные векторы z представляют из себя разности всевозможных сочетаний первичных векторов a:

$$zij = ai - aj$$

Данное усложнение увеличит вычислительную сложность модели, но оно позволит учитывать возможные сочетания слов. В дальнейшем можно будет масштабировать модель, применив, например, сочетания более чем двух первичных векторов.

Общая схема модели представлена на рисунке 1.

Рис.1 – схема модели.

В дальнейшем, на основании полученного массива вторичных векторов, можно использовать любую другую модель классификации, например, реализованную в любой существующей библиотеке для конструирования систем распознавания и классификации. В общем виде, предложенная модель является всего лишь модификацией известных моделей, с дополнительным компонентом, преобразующим исходные вектора входных данных в новые данные.

В качестве модели для классификации была взята модель, реализуемая библиотекой spaCy[10-12]. Её архитектура изображена на рисунке 2.

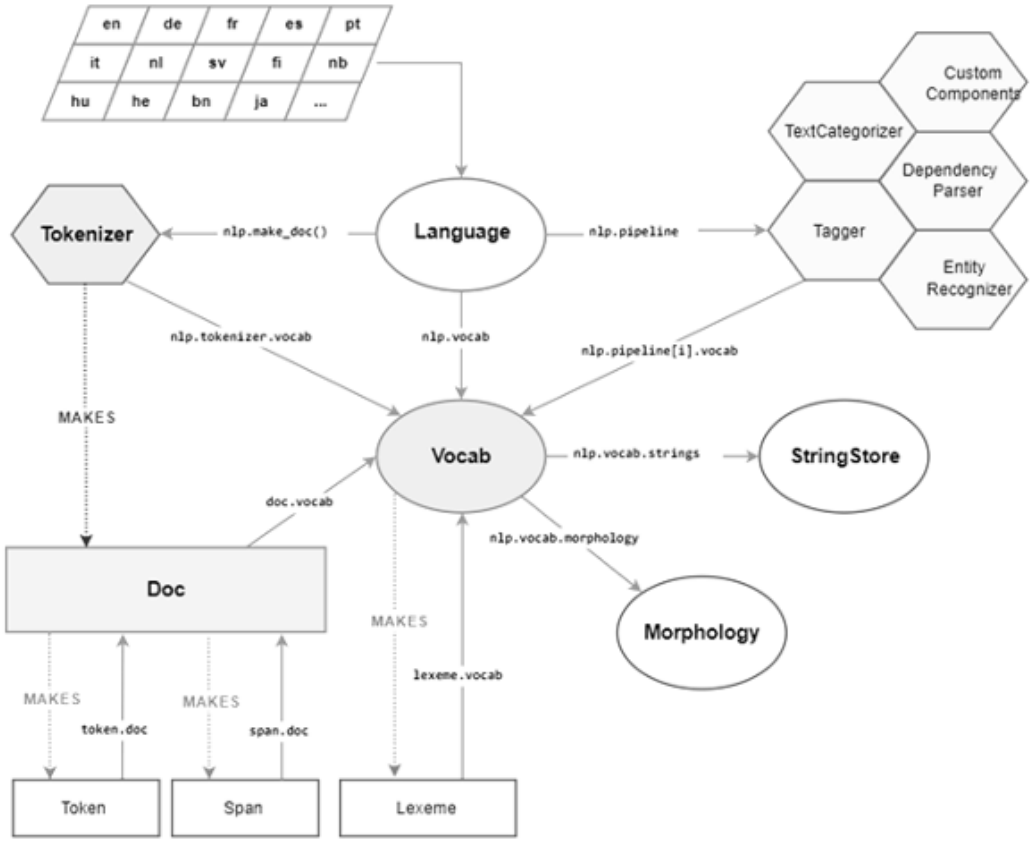


Рис.2 – архитектура модели, используемой библиотекой spaCy.

2.4 Обучение модели

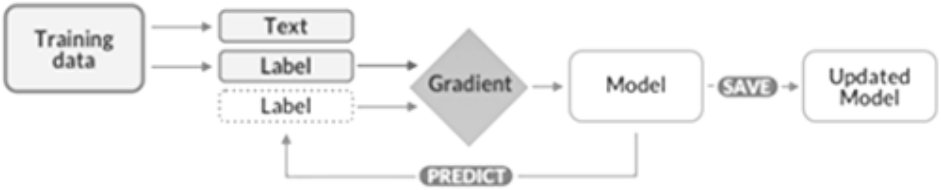


Рис.2 – алгоритм обучения модели из библиотеки spaCy[11].

Обучение сети происходит с помощью метода обратного распространения ошибки[9]. Тестовый набор для обучения нейронной сети должен содержать наиболее широкий набор используемых символов. Количество сообщений должно быть несколько для каждого необходимого типа эмоций.

В результате распознавания сообщения выдаётся результат, какой эмоциональной окраске оно соответствует. Затем, в тестовом наборе происходит сверка вышедшего результата с ожидаемым результатом, вычисляется разница (то есть – ошибка), и затем, с использованием данной ошибки происходит перерасчёт весовых коэффициентов нейронов.

$$\delta = y0 - y$$

где y0 – это ожидаемый вывод сети, а y – это реальный вывод слоя. Помимо выходного элемента ошибки можно определить и для всех элементов скрытого слоя нейронной сети, об этом мы поговорим чуть позже.

Дельта-правило заключается в следующем – изменение величины весового коэффициента должно быть равно:

$$\Delta wjk = \eta \delta kxj$$

Где η – норма обучения. Это число мы сами задаем перед началом обучения. xj – это сигнал, приходящий к элементу k от элемента j. А δk – ошибка элемента k.

Таким образом, в процессе обучения нейронной сети на вход мы подаем образец за образцом, и в результате получаем новые значения весовых коэффициентов. Обычно обучение заканчивается, когда для всех вводимых образцов величина ошибки станет меньше определенной величины. После этого сеть подвергается тестированию при помощи новых данных, которые не участвовали в обучении. И по результатам этого тестирования уже можно сделать выводы, хорошо или нет справляется сеть со своими задачами[8,9].

Заключение

В результате был разработан метод классификация текстов по эмоциональной окраске. Её преимуществом является то, она учитывает сочетания слов. Была предложена библиотека для возможности программной реализации разработанного метода. Имеется возможность масштабирования модели от работы с парами слов до работы с множественными сочетаниями. Ещё одним преимуществом является итеративное обучение, для чего не требуется обучающая выборка больших объёмов.

Литература

[1] D. Palchunov and A. Vaganova. Methods for Developing Digital Twins of Roles Based on Semantic Domain-Specific Languages, 2021 IEEE 22nd International Conference of Young Professionals in Electron Devices and Materials (EDM), 2021, pp. 515-519, doi: 10.1109/EDM52169.2021.9507716.

[2] D. Palchunov, A. S. Tregubov. Semantic methods of intelligent assistant developing. In: 2021 International Symposium on Knowledge, Ontology, and Theory (KNOTH), 2021, pp. 30-35. doi: 10.1109/KNOTH54462.2021.9686335.

[3] 24.Solovyev, V., V. Polyakov, V. Ivanov, I. Anisimov and A. Ponomarev, 2013. An approach to semantic natural language processing of Russian texts. Research in Computing Science, 65: 65-73.

[4] Пикалёв Я. С. РАЗРАБОТКА СИСТЕМЫ НОРМАЛИЗАЦИИ ТЕКСТОВЫХ КОРПУСОВ // Проблемы искусственного интеллекта. 2022. №2 (25).

[5] Stratis K., Use Sentiment Analysis With Python to Classify Movie Reviews. URL: https://realpython.com/sentiment-analysis-python/ (Дата обращения - 08.06.2023).

[6] Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space // In Proceedings of Workshop at ICLR. — 2013a.

[7] Mikolov T., Yih W.-T., Zweig G. Linguistic Regularities in Continuous Space Word Representations // In Proceedings of NAACL HLT. — 2013b.

[8] Хайкин С. Нейронные сети: Полный курс. Пер. с англ. Н. Н. Куссуль, А. Ю. Шелестова. 2-е изд., испр. — М.: Издательский дом Вильямс, 2008, 1103 с.

[9] Обработка естественного языка с помощью spaCy - шаги и примеры. URL: https://qwertybox.ru/articles/194347/ (Дата обращения - 09.06.2023)

[10] Обработка естественного языка с помощью spaCy - шаги и примеры. URL: https://qwertybox.ru/articles/194347/ (Дата обращения - 09.06.2023)
[11] Industrial-Strength Natural Language Processing URL: https://spacy.io (Дата обращения - 09.07.2023).
[12] spaCy 101: все, что вам нужно знать. Часть 1. URL: https://webdevblog.ru/spacy-101-vse-chto-vam-nuzhno-znat-chast-1/ (Дата обращения - 09.06.2023).
[13] Лучшие чат-боты с искусственным интеллектом: ChatGPT и альтернативы. URL: https://trashexpert.ru/software/best-chatbots/ (Дата обращения - 08.07.2023).
[14] GPT-4 Technical Report. [cs.CL] 27 Mar 2023.
[15] Жеребцова Юлия Андреевна, Чижик Анна Владимировна СРАВНЕНИЕ МОДЕЛЕЙ ВЕКТОРНОГО ПРЕДСТАВЛЕНИЯ ТЕКСТОВ В ЗАДАЧЕ СОЗДАНИЯ ЧАТ-БОТА // Вестник НГУ. Серия: Лингвистика и межкультурная коммуникация. 2020. №3.
[16] Новиков Д.А. Закономерности итеративного научения. М.: Институт проблем управления РАН, 1998. – 77 с.

Программная архитектура типовых узлов научных данных и вычислений по атмосферным наукам, основанная на их онтологических моделях

Привезенцев А.И. [1,2]

[1]*ИОА СО РАН, пл. Академика Зуева, 1, г. Томск, 634055, Россия.*
[2]*ФИЦ ИВТ, пр-т Академика Лаврентьева, 6, г. Новосибирск, 630090, Россия.*

alexey@privezentsev.ru

Аннотация. *В данной работе представлено описание программной архитектуры, которая создана с целью облегчения процессов разработки, развертывания и поддержки небольших типовых научных информационно-вычислительных систем, способных явно взаимодействовать друг с другом. Построение научной семантической распределенной децентрализованной сети, состоящей из малых типовых информационно-вычислительных систем для обработки научных данных и выполнения вычислений, является эффективным способом автоматизации ряда задач компьютерной поддержки комплексных сложносоставных научных исследований. Ключевым артефактом семантической распределенной децентрализованной сети являются автономные программные агенты, которые ситуативно взаимодействуют с небольшими поставщиками научных данных и вычислений. В современном мире такие провайдеры данных и вычислений обычно являются узконаправленными научными информационно-вычислительными программными системами, которые поддерживаются, как правило, небольшими научными командами. Программная архитектура, рассматриваемая в данном контексте, формируется на общих концепциях использования онтологических моделей в программной инженерии, таких как разработка, управляемая онтологией, разработка с возможностями онтологий, архитектура на основе онтологий, архитектура с возможностями онтологий. Перечисленные идеи опираются на базовое представление онтологий в рамках информационного пространства Semantic Web и отражают возможности использования онтологических моделей на всех стадиях жизненного пути программной системы, которые должна в том или ином виде проходить любая научная информационно-вычислительная система.*

Ключевые слова: онтология, OWL, онтологическая модель, семантическая модель, программная разработка, программная архитектура, информационная система, научная информационно-вычислительная система, узел научных данных, узел научных вычислений, компьютерная поддержка научных исследований, атмосферные науки

1 Введение

Автоматизация с помощью компьютера ряда процессов информационной деятельности научных сотрудников является важной проблемой в рамках комплексной компьютерной поддержки научных исследований. Современные компьютерные информационные платформы всё больше стремятся достигнуть качественного удовлетворения не только общих, но и более узких профессиональных информационных потребностей человека в пределах заданной предметной области путем предоставления необходимых информационных ресурсов и услуг. Идёт явный переход в парадигме компьютерной поддержки научных исследований от удовлетворения примитивных простых информационных проблем пользователей к удовлетворению сложных составных информационных потребностей исследователей

посредством использования программных автономных интеллектуальных ассистентов. В физических науках необходима явная интеллектуальная обработка больших научных числовых массивов данных из различных специализированных информационных систем, связанных с помощью интеллектуальных программных агентов. В таком случае такие узкоспециализированные научные профессиональные информационные системы представляются открытыми для взаимодействия не только с пользователем человеком, но открытыми к взаимодействию с пользователем автономным программным агентом. Для обработки сложносоставных информационных запросов от интеллектуальных программных агентов необходимо создавать семантические модели каждой узкопрофильной научной информационно-вычислительной системы. Коллекции метаданных информационных систем с семантическими моделями фиксируются и предоставляются в виде публичных реестров для поиска открытых к межмашинному взаимодействию типовых поставщиков научных данных и вычислительных услуг. Формирование решений по достижению удовлетворения новых информационных запросов исследователей наиболее вероятно приведет к необходимости построения всё большего количества оригинальных разнообразных интеллектуальных агентов и параллельного построения семантических моделей. Такие семантические модели от информационных провайдеров данных и вычислительных услуг необходимы для онтологической связи таких запросов с базой знаний этих самых интеллектуальных программных агентов. Следовательно, эффективным путем удовлетворения сложных составных информационных потребностей исследователей является использование обобщающих и стандартизирующих информационных платформ, позволяющих открытое и гибкое межмашинное взаимодействие разнообразных интеллектуальных автономных программных агентов через интеграцию и согласование множества семантических моделей научных узкопрофессиональных предметных областей. Это позволяет обеспечить более оптимальную обработку данных и удовлетворение всего спектра разнообразия информационных запросов научного сообщества в рамках не только простых, но и комплексных исследований.

Ключевой проблемой с точки зрения построения автономных интеллектуальных помощников в научной среде по физическим наукам является необходимость переориентации существующих научных проблемно-ориентированных информационно-вычислительных систем в части прикладных программных интерфейсов с исключительно человека [1] на программных агентов, требующих открытые спецификации не только для интерфейсов взаимодействия, но также для различных вариантов служб и структур представления данных, также используемых в технологиях интеллектуальной открытой межмашинной программной коммуникации. Текущая тенденция перехода к интеллектуальному межмашинному взаимодействию профильных специализирующихся на узкой научной предметной области информационных систем создает потребность удовлетворения информационных требований в первую очередь программных агентов в качестве основных пользователей таких информационных систем. Необходимость и требование современного научного мира не только к локальной, но и к глобальной кооперации формируют запрос на такие информационные системы, которые предоставляют ясные открытые спецификации для своих прикладных программных интерфейсов. Подобные информационные системы с профильными функциональными возможностями, реализующие разнообразные варианты служб для взаимодействия посредством согласованных форматов представления и описания информационных ресурсов, позволяют более эффективно быть использованными в совместных не просто интеграционных, а более масштабно в инфраструктурных проектах, связанных с доступом к многочисленным качественным числовым массивам научных данных в физических науках. Следовательно, актуальной становится необходимость формирования общего подхода к программной разработке таких информационных систем, ориентированных прежде всего на информационные потребности программных агентов и предоставляющих ясные открытые однозначные спецификации для всего спектра программных артефактов интеллектуального взаимодействия от прикладных программных интерфейсов и структур представления данных до множества предоставляемых этими системами функциональных служб. Общая программная платформа позволит обеспечить более эффективное межмашинное взаимодействие и предоставит более гибкое решение для удовлетворения информационных потребностей программных агентов прежде всего в рамках кооперирования и работы над разнообразными совместными проектами в физических науках.

Для научной общественности является заманчивой концепция интеллектуальной компьютерной поддержки научных исследований с делегированием решений ряда рутинных прикладных информационных задач от высококвалифицированных ученых к автономным персонализированным интеллектуальным программным ассистентам. Эта идея в рамках развития всемирной глобальной паутины Веб изначально была изложена в работе о

семантической сети (Semantic Web) [2], где описывались автономные программные агенты, взаимодействующие с различными поставщиками данных и вычислительных услуг. Основной идеей данного исследования является рассмотрение подхода к обобщению и типизации небольших прикладных информационных систем с целью формирования децентрализованной распределенной информационно-вычислительной сети, состоящей из семантически связанных узлов научных данных и вычислений, явно специализирующихся на отдельных областях знаний в физических науках. Предлагаемая децентрализованная распределенная сеть в рамках общей концепции Semantic Web предполагает построение общего семантического информационного пространства взаимодействия программных агентов и поставщиков научных данных и услуг. Такое программное решение позволит более гибко подходить к множеству вероятных прикладных проблем взаимодействия с учетом необходимости использования не просто качественных данных, но и обеспечивая должный профессиональный уровень предметной экспертизы информационных ресурсов и оказываемых услуг в рамках всего научного сообщества. Таким образом, за счет повышения общей эффективности в области компьютерной поддержки научных исследований должен произойти качественный скачок в научных возможностях малых научных коллективов в сфере физических наук, предоставляя им доступ к ресурсам посредством автономных интеллектуальных программных агентов, которые ранее могли себе позволить только крупные научные сообщества и кооперации.

В научном мире уже давно появились подходы, направленные на систематизацию научных ресурсов в глобальной паутине Веб, например, подходы e-Science и CyberInfrastructure. Вышеизложенные идеи в том или ином виде представлены в подходе e-Science [3], где предложено систематизировать научные информационные ресурсы глобальной паутины Веб по трем уровням. Первый уровень это сырые данные, второй уровень это интерпретируемые данные, являющие уже информацией, и наконец, на третьем уровне находится информация пригодная для целевого использования в прикладных задачах, являющая собой прикладные знания. Этот подход также основан на базовых идеях информационного пространства Semantic Web и использует разработанные там технологии и формальные языки представления информации и знаний. В рамках этого подхода существующие информационные системы, обладающие возможностями и ресурсами всех трех вышеуказанных уровней, являются так называемыми семантическими информационными системами. Такие семантические информационные системы обязаны обладать функциональными возможностями интеллектуальной машинной коммуникации и интеграции. Однако, разработка таких научных информационно-вычислительных систем, открытых для межмашинного взаимодействия, является сложной и затратной задачей, что обуславливает относительно небольшое количество подобных систем. Поэтому важной задачей является повышение эффективности жизненного цикла (пути) [4] прикладных семантических научных информационно-вычислительных систем, существующих в глобальном информационном пространстве Веб. Хорошая программная архитектура таких информационных систем играет важную роль в эффективности процессов работы, происходящих на разных этапах их жизненного цикла. В данной работе архитектура программной системы [5] определяется как структурная форма системы, получаемая выделением компонентов системы с учетом множества вариантов их организаций и представлений, а также явным указанием способов программных взаимодействий этих компонентов между собой. Как правильно указано в работе [5] о программной архитектуре, назначение архитектуры программной системы состоит в упрощении процессов разработки, развертывания и поддержания системы, при этом ключевая стратегия такого упрощения заключается в сохранении как можно большего количества вариантов развития на как можно дольше временном отрезке с учетом ограничений оптимальности и эффективности. В результате эффективность выполнения работы в рамках процессов жизненного цикла информационной системы растёт за счет уменьшения затрачиваемых человеческих ресурсов на разработку и сопровождение таких программных систем.

2 Прикладные проблемы компьютерной поддержки научных исследований

В сфере компьютерной поддержки научных исследований существует значительная проблема, заключающаяся в необходимости быстро и экономично создавать и долгосрочно эволюционно сопровождать семантические узкоспециализированные научные информационно-вычислительные системы, предназначенные для использования небольшими коллективами с числом пользователей менее 100 в виде людей или программных ботов. При решении этой проблемы необходимо уделять внимание не только временным и качественным аспектам разработки программных решений, но также финансовым аспектам. Как правило, малые научные коллективы ожидают в современном информационном мире от любых используемых

программных систем адекватного решения широкого списка информационных задач, на реальную реализацию которых могут потребоваться большие трудозатраты. Даже для обладания приемлемыми функциональными возможностями при создании относительно простой информационной системы могут потребоваться значительные ресурсы. В настоящее время малые научные коллективы в атмосферных науках часто финансируют инженерно-практические программные решения исключительно в рамках короткого срока своего грантового финансирования. При этом, как правило, финансировании инженерных разработок прикладных программных решений не является приоритетной задачей. Основная цель грантов состоит в выполнении заявленных научных исследований в физических науках, а не разработке инженерных программных решений для автоматизации информационных процессов. Тем не менее переход к массовым инфраструктурным решениям, которые могут взаимодействовать друг с другом через интернет, позволит перейти на качественно новый уровень глобальных интеграционных проектов. Такие проекты в будущем будут выполняться небольшими научными коллективами с помощью персонализированных интеллектуальных программных агентов в качестве автономных помощников. В современном российском научном сообществе финансирование разработки программных решений не поощряется и считается инженерной работой, поэтому на них выделяются исключительно небольшие бюджеты. Такие бюджеты используются либо для покупки недорогих лицензий на типовое программное обеспечение, либо на оплату технической поддержки и сопровождения программных систем, но будут маленькими для оплаты трудозатрат по индивидуальной разработке сложных программных систем.

Для решения данной проблемы можно рассмотреть два основных варианта. Во-первых, можно проводить индивидуальные убеждения и договоренности о необходимости разработки индивидуальных инфраструктурных решений, увеличивая финансирование до необходимых размеров. Во-вторых, можно переходить на использование сторонних универсальных шаблонных программных инструментов, способных предоставлять возможности быстрой и гибкой настройки под конкретные требования исследователей и при этом позволять внедрение при наличии маленьких бюджетов. Однако, на данный момент такие решения отсутствуют в научном сообществе по атмосферным наукам. Сроки процессов разработки и дальнейшего сопровождения научных информационно-вычислительных систем в физических науках являются ключевыми параметрами оценки эффективности решения вышеизложенной проблемы. Подобные информационные системы изначально должны обладать для членов научных коллектив и их исследовательских проектов необходимыми функциональными возможностями, которые при этом должны непрерывно с течением времени совершенствоваться. Появляется необходимость в сокращении сроков, затрачиваемых на разработку таких систем, и максимизации сроков сопровождения и эксплуатации этих систем. На практике научные информационно-вычислительные системы обычно находятся в стадии разработки лишь при наличии финансирования от гранта, а затем эти системы либо переходят в состояние затухающей эксплуатации, либо закрываются сразу по завершении грантового проекта. Для решения данной проблемы также можно рассмотреть два варианта. Во-первых, можно искать дополнительные источники финансирования и периодически вручную самим или привлекая сторонних подрядчиков актуализировать программные решения, доводя их до необходимого обновленного состояния. Во-вторых, можно переходить на использование сторонних универсальных шаблонных программных инструментов, позволяющих проводить бесшовное автоматические обновления научных информационных систем, создаваемых на базе выбранной общей технической программной платформы. Такие программные решения должны разрабатываться отдельно специализирующимися на них техническими командами, имеющими высокую квалификацию в области компьютерных наук.

Второй ключевой большой прикладной проблемой компьютерной поддержки научных исследований является возможность ситуативной автоматической организации взаимодействия программных агентов с необходимыми научными узлами в рамках интеграционных проектов. Под научным узлом понимается информационная веб-система, поддерживаемая небольшим научным коллективом, которая предоставляет узкоспециализированные научные информационно-вычислительные возможности для множества пользователей. В рамках интеграционных проектов для обеспечения межмашинного взаимодействия посредством интеллектуальных программных агентов необходимо создание и использование соответствующих инфраструктурных программных решений, имеющих механизмы автоматизации представления открытых прикладных программных интерфейсы для доступа к ресурсам и услугам, создаваемых на их основе. Формирование подобных инфраструктурных программных решений в виде семантических научных информационно-вычислительных сетей, состоящих из узлов-поставщиков данных и вычислений может быть либо на основе

централизованного либо децентрализованного подходов. Централизованный подход позволяет достичь большего контроля и управляемости за счёт простоты в разработке и сопровождении. Если же рассматривать преимущества децентрализованного подхода, то тот предполагает повышенную отказоустойчивость, гибкость горизонтального масштабирования и необходимую автономность узлов с возможностью независимого внедрения новых технологий на различных узлах в разное время.

Третьей ключевой большой прикладной проблемой компьютерной поддержки научных исследований, которую требуется решить, является необходимость соответствия информационных ресурсов, публикуемых в семантических научных информационно-вычислительных сетях общим универсальным принципам FAIR [6]. Цель поддержки соответствия этим принципам заключается в необходимости эффективного использования качественных научных массивов данных и экспертных вычислительных услуг, предоставляемых различными коллективами и организациями в качестве их независимых поставщиков. Данные, соответствующие принципам FAIR, должны быть обнаруживаемыми, доступными, совместимыми и многоразовыми. Следует отметить психологический аспект, наблюдаемый с практической стороны, что многие исследователи в атмосферных науках имеют страхи и сомнения по поводу необходимости открыто публиковать и делиться своими числовыми результатами исследований или своими вычислительными программами. Это нежелание и неуверенность порождают проблему наличия большого слоя данных, информации и знаний, потенциально скрытых от программных агентов. В результате исследователь, выступающий в роли конечного потенциального потребителя научных данных и услуг может лишиться значимой части существующих, но недоступных знаний научного сообщества. В первую очередь внедрение этих принципов обусловлено необходимостью расширения спектра открытых качественных научных данных и вычислений, предоставляемых различными исследователями сейчас на мизерном уровне, таким образом, скрывая многие свои наработки от публичного изучения. Полнота и качество открытых опубликованных и доступных для научной общественности информационных ресурсов имеет важнейшее значение при построении рассматриваемой интеллектуальной научной информационно-вычислительной сети. Расширение публичного доступа к таким информационным ресурсам в веб-пространстве с использованием защищенной программной инфраструктуры является одним из ключевых направлений данной исследовательской работы.

3 Семантическая научная распределенная децентрализованная информационно-вычислительная сеть

Программная инженерия на данный момент продолжает активно развиваться. Существуют различные варианты применения декларативных моделей. Один из наиболее известных и старых таких подходов MDA (Model Driven Architecture) [7], разработанный консорциумом OMG (Object Management Group), эволюционно сформировался из языка унифицированного моделирования (UML). Ключевой идеей этого подхода было желание, чтобы модель явно участвовала в разработке программных систем. В этом подходе модель из прообраза вручную создаваемого исходного кода становится важным входным артефактом для поверхностной кодогенерации посредством инфраструктурных программных систем. В результате такого подхода на основе трансформации изначальной модели посредством инфраструктурного программного инструмента моделирования происходит создание планируемой программной системы. Развитием уже этого подхода стала работа [8], предложенная одной из рабочих групп организации W3C, работая над подпроектом Semantic Web, где по аналогии с подходом MDA формой представления изначальных моделей стали онтологии, определяя подход ODA (Ontology Driven Architecture). Практическое обобщение применения онтологических моделей в программной инженерии есть в работе [9], описывая четыре разных подхода. Управляемая онтологией программная разработка (Ontology-Driven Development) предполагает использование онтологических моделей во время разработки программной системы по аналогии с подходом MDA. Программная разработка с возможностями онтологий (Ontology-Enabled Development) предполагает разнородные описания программных артефактов в виде онтологических моделей с целью дополнительной поддержки разработчиков. Программная архитектура на основе онтологий (Ontology-Based Architecture) предполагает использование онтологической модели в качестве базового основополагающего артефакта программной архитектуры всей создаваемой системы. Программная архитектура с возможностями онтологий (Ontology-Enabled Architecture) предполагает использование онтологической модели для дополнительной инфраструктурной поддержки пользователей создаваемой системы.

В представленной работе предлагается комплексный подход к проблемам, связанным с обеспечением компьютерной поддержки научных исследований в атмосферных науках. Этот подход, основанный на явном использовании онтологических (семантических) моделей, предполагает создание типовых программных решений, предназначенных для обработки и анализа научных данных и оказания вычислительных услуг. Эти типовые программные решения в виде узлов научных данных и вычислений в работе рассматриваются в виде информационных систем с целевой аудиторией, являющейся профессионалами в области атмосферных наук. Ключевая идея таких информационных систем соответствует вышеизложенному подходу основанной на онтологии программной архитектуры. В основанной на онтологии программной архитектуре онтологическая модель являясь базовым основополагающим артефактом такой архитектуры программной системы. Такой подход требует создания согласованной с предметной онтологической моделью, построенной схемы данных и постоянной синхронизацией с работающим программным кодом модулей информационной системы.

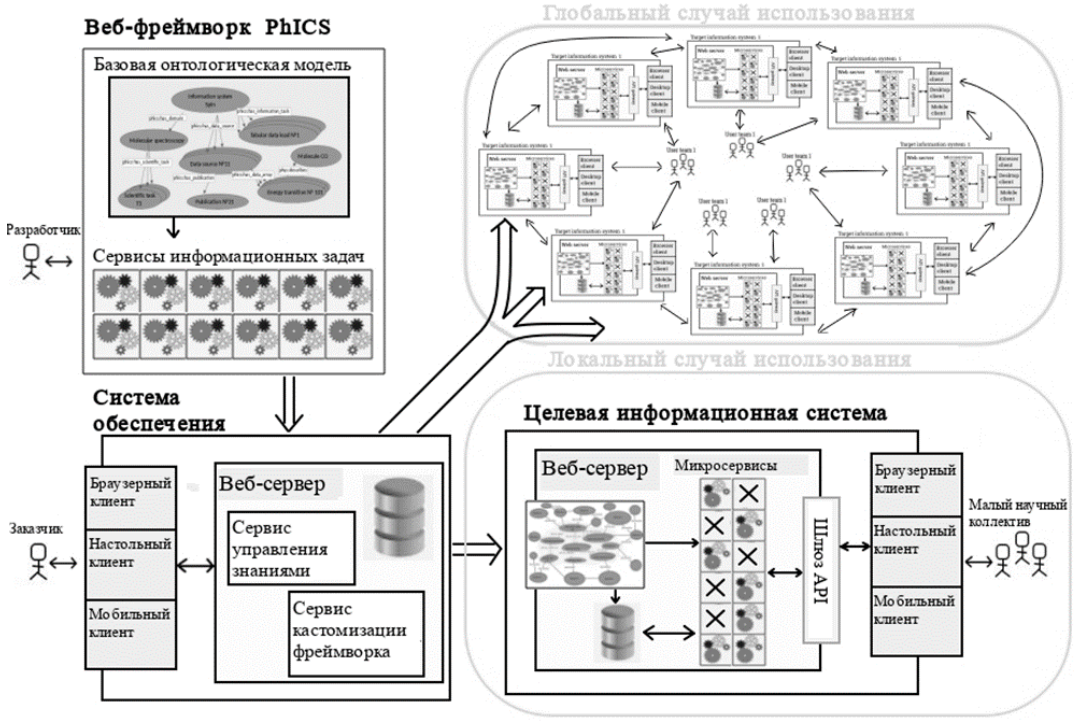


Рис.1. Концептуальная схема семантической децентрализованной научной информационно-вычислительной сети.

Рассматриваемая идея семантической научной сети в физических науках концентрирует в себе множество уже имеющихся в программной инженерии наработок, в общих чертах, представленных на оригинальной концептуальной схеме семантической децентрализованной научной информационно-вычислительной сети. Эта схема, изображенная на рисунке 1, объединяет типовые научные узлы в сеть с целью автоматически обрабатывать сложносоставные научные информационные запросы. В этой сети узлы данных и вычислений можно связывать по семантическим моделями предметных областей, используя публичные реестры открытых к межмашинному взаимодействию информационно-вычислительных систем. Подобная семантическая децентрализованная научная информационно-вычислительная сеть позволит множеству небольших и узкоспециализирующихся научных коллективов решать всё более сложные задачи с использованием знаний и ресурсов потенциально большого числа участников такой сети. Учитывая глобальную тенденцию к совместным интеграционным проектам в научном сообществе уже давно сложился запрос на необходимость интегрировать существующие данные и программные решения информационных систем. В данной работе делается акцент на программной архитектуре типовых узлов научных данных и вычислений, составляющих ключевые элементы планируемой сложной семантической децентрализованной научной информационно-вычислительную сети. Такие типовые узлы позволят достаточно

просто осуществлять публикацию и интеграцию существующих, но скрытых от широкой аудитории, информационных ресурсов и вычислительных услуг, имеющихся в научном сообществе по физических наукам.

4 Программная архитектура типового узла научных данных и вычислений

Обычно, любая программная система, в том числе информационная система, проходит через различные этапы своего развития, начиная с анализа и проектирования, далее через этап реализации к этапу развертывания, затем начинается этап эксплуатации с параллельным сопровождением и поддержкой. На любом этапе такого жизненного пути могут применяться специализированные дополнительные системы обеспечения (enabling system). Обеспечивающая система является дополнением к основной системе на протяжении всех стадий ее жизненного цикла, однако она необязательно имеет прямое влияние на функционирование основной системы. Конвейерный взгляд к одновременной потоковой разработке и поддержке нескольких целевых информационных систем с использованием одной специализированной системы обеспечения предполагает каркасный подход, часто используемый в программной инженерии. Потоковое создание небольших типовых научных информационно-вычислительных систем с уклоном на автоматизацию большого количества рутинных задач на разных этапах жизненного цикла хорошо согласуется с практиками каркасного подхода. Программный фреймворк (каркас) определяет базовые структуры данных и программные прикладные интерфейсы, закладывая прочный фундамент для эффективного наращивания в дальнейшем требуемых функциональных возможностей. Такой программный фундамент является ключевым артефактом формируемой программной системы, используя соглашения по наименованию и другие инженерные практики для легкой и гибкой настойки желаемой конфигурации системы. Необходимость постоянного эволюционного изменения и расширения фреймворка дополнительными программными модулями является важным аспектом, учитываемым при формирования требуемых функциональных возможностей. Программный фреймоворк позволяет гибко подходить к разработке типовых (шаблонных) программных систем, задавая правила и соглашения не только для дальнейшей конфигурации системы, но и по соединению во время разработки различных программных взаимосвязанных компонентов создаваемого сложного комплексного решения. Такая инженерная практика, как соглашения по конфигурации позволяет автоматизировать многие технические рутинные действия, ускоряя не только процесс разработки, но и дальнейшее сопровождение программной системы. Описываемая архитектура типового узла научных данных и вычислений опирается на идею возможности использования программного решения по управлению разработкой и сопровождению целевых информационных систем в виде общей обеспечивающей системы. Такая система обеспечения позволит гибко дорабатывать и донастраивать обобщенный универсальный специализированный программный фреймоворк посредством изменений онтологической модели, заданной целевой информационной системы. При этом прикладная онтологическая модель целевой информационной системы представляет собой наиболее полную спецификацию концептуализации заданных аспектов предметной области, достаточной для формирования актуального словаря концептов и отношений между ними с целью использования при создании и управлении такой целевой информационной системой.

Предлагаемая программная архитектура, изображенная на рис. 1, включает в себя несколько инженерных практик использования онтологий в программной инженерии, описанных выше. Ключевая идея рассматриваемой программной архитектуры основана на подходе основанной на онтологии программной архитектуры, акцентируя внимание на первоначальной необходимости построения онтологической модели целевой информационной системы, являющейся ключевым основополагающим артефактом архитектуры такой программной системы на всем ее жизненном пути. Подход управляемой онтологией программной разработки предлагает немного иной взгляд на практическое использование онтологической модели целевой информационной системы на этапе разработки непосредственно этой программной системы. В рассматриваемой программной архитектуре предлагается использовать онтологическую модель целевой информационной системы для создания и управления схемами баз данных предметной области этой системы с соответствующими связанными согласованными программными артефактами, представляющими в том числе и необходимый обрабатывающий программный код модулей этой системы. Подход программной разработки с возможностями онтологий в рассматриваемой программной архитектуре обеспечит возможность семантического описания широкого круга разнообразных артефактов в виде дополнительных согласованных онтологических моделей, необходимых для расширенной поддержки деятельности

разработчиков этой системы. Подход к программной архитектуре с возможностями онтологий для рассматриваемой программной архитектуры позволит использовать дополнительные онтологической модели для расширенной инфраструктурной семантической поддержки пользователей таких создаваемых программных систем. В соответствии с этими инженерными практиками использования онтологических моделей возможно для построения широкого круга узкоспециализированных интеллектуальных подсистем поддержки принятия решений, например, для подсистемы формирования персонализированных числовых массивов предметных данных в соответствии с ситуативными информационными потребностями пользователей. Для пользователей программных агентов онтологические модели целевой информационной системы предоставят возможности открытого семантического представления данных и вычислительных услуг каждого узла такой семантической децентрализованной научной информационно-вычислительной сети, позволяя открытое межмашинное взаимодействие с целью автономного выполнения сложносоставных потоков работ.

Предложенная архитектура открытых научных информационно-вычислительных систем по атмосферным наукам объединяет в себе классические программные архитектуры. В рассматриваемой программной архитектуре целевой информационной системы предлагается задействовать идеи клиент-серверной архитектуры, являющейся базовой идеей дистанционного взаимодействия разрабатываемой программной системы с пользователем в виде человека или программного агента. Клиентская часть такой системы может иметь несколько различных воплощений, например, реализация графического интерфейса пользователя может иметь воплощение в виде настольного приложения, веб-приложения или мобильного приложения. Серверная часть такой системы должна иметь открытый прикладной пользовательский интерфейс для внутреннего взаимодействия с клиентской частью создаваемых узлов данных и вычислений, а также для предоставления возможности непосредственного межмашинного взаимодействия при решении штатных типовых информационных задач внешних клиентов программных агентов. С точки зрения организации научной совместной распределенной коллективной работы рассматриваемая программная архитектура предлагает реализацию идей классической распределённой децентрализованной архитектуры. Ключевой идеей является построение общего информационного пространства для семантически связанной распределённой коллективной работы. Такое общее информационное пространство представляет собой семантическую распределенную информационно-вычислительную сеть, состоящую из множества автономных узлов в виде небольших узкоспециализированных информационных систем, существующих в рамках глобальной сети Интернет. Каждый узел данных и вычислений обладает собственным узкоспециализированным хранилищем научных данных и соответствующим внутренним потребностям набором функциональных возможностей, обеспечиваемым модульным решением шаблонных информационных задач. Оригинальные репозитории данных и репозитории программных реализаций вычислительных алгоритмов являются ключевыми информационными ресурсами формируемой семантической распределенной информационно-вычислительной сети для научного сообщества, специализирующегося на физических науках. При необходимости формирования централизованной структуры узлов, например, в отдельном взятом интеграционном проекте в качестве центрального узла выбирается и назначается один наиболее подходящий под требования этого проекта узел среди множества равноправных узлов семантической сети по физическим наукам. Подобный выбор центрального узла происходит в рамках договоренностей с помощью достижения взаимных соглашений между участниками определенного научного коллективного проекта. При таком централизованном подходе центральный узел может выполнять технические функции общего назначения, например, предоставлять функции единой точки аутентификации и авторизации, то есть использоваться в качестве хранилища учетных записей пользователей, ролей и групп в дополнение к локальным возможностям каждого отдельно взятого узла. В дополнение к распределенным узлам каждый отдельно взятый центральный узел при достаточных технических возможностях может быть источником быстрого доступа и резервным хранилищем определенного количества опубликованных информационных ресурсов по физическим наукам. Подобное полное резервное хранилище открытых публичных числовых массивов данных в отличие от частного локального хранилища каждого узла позволяет перенести нагрузку на вычислительные мощности центрального узла, наиболее подходящего по техническим параметрам. Равноправный подход предполагает, что каждый узел семантической распределенной научной информационно-вычислительной сети может импортировать в свое локальное хранилище любые открытые или доступные по определенным правилам числовые массивы данных и оригинальные программные реализации вычислительных решений от других равноправных узлов с целью оптимизации локального выполнения определенных информационных задач. Общий подход к работе распределённой

системы предполагает возможности межмашинного выполнения комплексных потоков работ с учетом их дистанционного запуска и мониторинга на разных узлах такой семантической децентрализованной распределенной сети. Рассмотренная программная архитектура, основанная на онтологических моделях целевых информационных систем с базовыми возможностями открытого межмашинного взаимодействия небольших разнородных узкоспециализированных научных информационно-вычислительных систем по атмосферным наукам должно привести к формированию более эффективной научной коллективной совместной работы научного сообщества.

Рассматриваемая программная архитектура, основанная на онтологических моделях, целевой информационной системы предполагает задействовать идеи микросервисной архитектуры для гибкого развертывания, горизонтального масштабирования и расширяемой технической поддержки. С этой точки зрения серверная часть рассматриваемой архитектуры предполагает программную реализацию в виде множества автономных веб-сервисов, состоящих из связанных между собой одной общей специализированной предметной задачей в виде разных программных наборов базовых модулей или модулей расширения. Микросервисная архитектура одного узла небольшой научной информационно-вычислительной системы в простом варианте предполагает устойчивую поддержку безопасного сетевого периметра. Такой сетевой периметр, как правило, организуется в рамках одного узла с одной точкой входа и взаимодействия, представляющего небольшую локальную сеть, связанных прикладными программными интерфейсами сервисов. Каждый сервис может программно реализовываться на своем технологическом стеке, предоставляя прикладной программный интерфейс для взаимодействия с ним. С другой стороны, очевидно, что сервисы будут в той или иной степени использовать общие разделяемые ресурсы, например, одну систему управления базой данных или один сервер очередей для параллельного выполнения продолжительных по времени информационных задач. Рассматриваемая комбинация нескольких классических программных архитектур в описываемой программной архитектуре небольших узкоспециализированных научных информационно-вычислительных систем по атмосферным наукам позволяет решать в том числе и современные информационные сложносоставные задачи, стоящие перед научным сообществом в целом и малыми научными коллективами в частности, как правило, в рамках их совместных проектов. Это позволяет обеспечить техническую поддержку и масштабирование системы, а также удовлетворить потребности разработчиков в различных технологических.

Предметная область рассматриваемых в работе подмножеством класса информационных систем ограничена физическими атмосферными науками. Основной упор в описании предметной области делается на возможность работы с большими опубликованными числовыми массивами значений взаимосвязанных наборов физических величин, накапливаемых в процессе научной деятельности небольших научных коллективов при решении своих прикладных научных задач. Ключевым артефактом описания целевых научных информационных систем является базовая онтологическая модель, предоставляющая терминологическую основу и структуру ожидаемых связей при моделировании предметной области такой целевой информационной системы. Наличие базовых терминов и базовых взаимосвязей необходимо для построения машиннообрабатываемой и интерпретируемой семантической модели, используемой далее системой обеспечения с целью создания и управления целевой информационной системой. Базовая онтологическая модель формируется на основе концептов, представляющих совокупности описаний, решаемых прикладных научных задач для выбранных моделей физических систем, ситуативно характеризуемых заданными наборами физических свойств. В рамках каждой научной задачи фиксируются и описываются исключительно необходимые и достаточные для представления результата решения такой задачи характеристики выбранной физической системы. В такой концептуализации физическая система описывается как единый концепт с набором важных в рамках выбранной научной задачи физических характеристик и общей структурой в виде упорядоченной совокупности взаимосвязанных выделенных элементов, также обладающих набором изучаемых физически важных свойств. Каждый элемент физической системы при описании может быть связан с детально описанной другой физической системой. При необходимости концептуализации физических процессов акцент описания можно сместить на сами протекающие в физической системе процессы через описание взаимодействий разных физических систем. Подобный подход базовой концептуализации позволит в зависимости от изучаемых аспектов предметной области формировать различные семантически согласованные модели одних и тех же объектов, и процессов реальности, описываемые в виде взаимодействующих физических систем с разными наборами рассматриваемых физических характеристик. Накапливаемые опубликованные числовые значения результатов решения таких научных задач позволяют формировать узкоспециализированные профессиональные хранилища

данных, представляемые в онтологической модели концептом источника данных. Концепт источника данных представляет собой опубликованный в научной работе результат решения научной задачи, описываемые одним или несколькими числовыми массивами значений физических величин. Для общей шаблонной работы с источниками данных в целевой научной информационно-вычислительной системе предполагается создание ряда программных реализация типовых информационных задач, подключаемых и настраиваемых в виде модулей такой системы посредством системы обеспечения.

Общая система обеспечения предназначена для автоматизации ряда рутинных задач, возникающих на различных этапах жизненного цикла каждой типовой целевой узкоспециализированной научной информационно-вычислительной системы. В процессе анализа и проектирования целевой информационно-вычислительной системы формируется её прикладная онтологическая модель, позволяющая описать ключевые научные понятия и их взаимосвязи в заданной предметной области с точки зрения ключевых аспектов проводимых исследований и решаемых прикладных задач. Основным описываемым аспектом таких научных узкоспециализированных прикладных задач является выявление и фиксация структуры опубликованных числовых массивов данных в заданной предметной области. После получения прикладной онтологической модели целевой информационной системы появляется возможность автоматической генерации целевого прототипа научной информационно-вычислительной системы посредством использования специализированных программных инструментов системы обеспечения. В процессе генерации целевой информационной системы её прикладная онтологическая модель является по сути конфигурационным файлом для программных инструментов системы обеспечения, предназначенная прежде всего для построения и актуализации реляционной схему данных выбранной предметной области. Базовые функциональные возможности для работы с данными целевой информационной системы, реализуются в виде модулей универсальных шаблонных информационных задач, настраиваемых с помощью той же самой прикладной онтологической модели. В рамках каждой заданной целевой научной информационно-вычислительной системы возможно расширение её узкоспециализированных функциональных возможностей с помощью дополнительной программной реализации оригинальных алгоритмов обработки и представления имеющихся информационных ресурсов посредством соответствующих программных инструментов общей системы обеспечения. Подобные расширения также требуется фиксировать в прикладной онтологической модели целевой информационной системы, используя терминологию базовой онтологической модели.

5 Заключение

В данной работе предлагается использовать онтологические модели целевой информационной системы для гибкого и быстрого создания и сопровождения типовых программных систем в рамках существующих идей программной инженерии. Создание и сопровождение с помощью онтологических моделей на всех этапах жизненного пути небольших узкоспециализированных научных информационно-вычислительных систем позволит более эффективно использовать небольшие ресурсы малых научных коллективов. Предлагаемые рассматриваемой программной архитектурой возможности открытого семантического межмашинного взаимодействия позволят повысить эффективность, сложность и масштабность решаемых небольшими научными коллективами задач. Предлагаемая программная архитектура направлена на удешевление, ускорение и гибкость создаваемых решений посредством упрощения работы на различных этапах процессов разработки, тестирования, развертывания и сопровождения целевых информационных систем в физических науках. Использование онтологических моделей целевых информационных систем позволит точнее, полнее и эффективнее использовать научным сообществом имеющиеся научные информационные ресурсы посредством упаковки их в типовые узлы научных данных и вычислений. Создание и поддержка качественных узкоспециализированных информационных систем и хранилищ данных является чрезвычайно трудозатратной работой, а попытка объединения таких программных систем является ещё более масштабной и сложной проблемой. Представленная концепция семантической научной распределенной информационно-вычислительной сети является одним из вариантов решения этой проблемы посредством предоставления программного инструмента для компьютерной поддержки научных исследований в атмосферных науках. Подобное решение позволит более эффективно решать сложносоставные информационные проблемы небольших научных коллективов. Изложенные идеи находятся в стадии воплощения программной реализации фреймворка на основе рассмотренной программной архитектуры типовых узлов научных данных и вычислений по физическим наукам.

Литература

[1] Калиниченко Л.А., Вольнова А.А., Гордов Е.П., Киселева Н.Н., Ковалева Д.А., Малков О.Ю., Окладников И.Г., Подколодный Н.Л., Позаненко А.С., Пономарева Н.В., Ступников С.А., Фазлиев А.З. Проблемы доступа к данным в исследованиях с интенсивным использованием данных в России // Информатика и ее применения. 2016. Т. 10, № 1, С. 3-26.

[2] Berners-Lee T., Hendler J., Lasssila O. The Semantic Web // Scientific American. 2001.

[3] De Roure D., Jennings N.R., Shadbolt N.R. The Semantic Grid: A Future e-Science // Grid Computing: Making the Global Infrastructure a Reality. 2003. pp. 437–470.

[4] Мацяшек Л. Анализ и проектирование информационных систем с помощью UML 2.0. М.: Вильямс, 2008. 816 с.

[5] Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. СПб.: Питер, 2019. 352 с.

[6] Jacobsen A.; de Miranda Azevedo R.; Juty N., et al. FAIR Principles: Interpretations and Implementation Considerations // Data Intelligence. 2020. pp. 10–29.

[7] Poole J.D. Model-Driven Architecture: Vision, Standards And Emerging Technologies // Workshop on Metamodeling and Adaptive Object Models. ECOOP, 2001. 15 pp.

[8] Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering. 2006. URL: https://www.w3.org/2001/sw/BestPractices/SE/ODA/ (дата обращения 29.09.2019).

[9] Happel H., Seedorf S. Applications of ontologies in software engineering. // Proceedings of 2nd International Workshop on Semantic Web Enabled Software Engineering (SEWE 2006). Athens, GA, USA. 14 p. URL: https://km.aifb.kit.edu/ws/swese2006/final/happel_full.pdf (дата обращения 29.09.2019).

Философские концепции в приложениях анализа данных и искусственного интеллекта

В.М. Резников

Институт философии и права СО РАН, пр. Николаева, д. 8, г. Новосибирск, 630090, Россия.

mathphil1976@gmail.com

Аннотация. *Продемонстрирована значимость применения философских концепций каузальности, объяснения и понимания в приложениях анализа данных и экспертных систем. Так, в случае исследования изучаемых данных с помощью причинных отношений упрощается введение и верификация свойства независимости, которое часто используется в математической статистике. Кроме того, применение причинных интерпретаций в байесовских сетях открывает новые возможности для понимания логики решения проблем компьютерными программами в области искусственного интеллекта. Показано, что критика философами «узких мест» статистического анализа данных, в частности понятия группирования данных, способствует прогрессу, так как в науке не принято критиковать методы, если нет альтернативных подходов. Исследована модификация задачного подхода применительно к искусственному интеллекту. Показана адекватность концепций понимания для исследований в области искусственного интеллекта. Обоснованы гипотетические перспективы плодотворного участия философов, имеющих подготовку в области формального анализа в качестве консультантов корректного использования философских концепций в науке.*
Ключевые слова: независимость, распределение, причинное отношение, объяснение, понимание, задачный подход

1 Введение

Известно, что успешное развитие и эффективное функционирование науки предполагает участие методологов. Методология науки занимается исследованием оснований методов, изучением перспектив их развития, анализом адекватности приложениям. Нередко исследователи успешно совмещают научные и методологические исследования. Однако по отношению к применению математики вне математики картина не столь отрадная. Дело в том, что существует несколько серьезных препятствий для участия математиков в методологической деятельности, связанной с применением математики вне этой науки. Во-первых, далеко не все математики интересуются приложениями, так например, чистые математики нечасто интересуются приложениями. Они небезосновательно утверждают, что успехи в чистой математике связаны с красотой математических объектов, и, по их мнению, красота не присуща приложениям математики. Во-вторых, математика изучает абстрактные отношения, и успешно решает проблемы, связанные с дедуктивными выводами на основе аксиом и ранее доказанных утверждений. Однако она не столь успешна при исследовании не столь абстрактных объектов и в решении проблем, связанных с индуктивными рассуждениями. Какое это имеет отношение к участию математиков в методологической деятельности? Дело в том, что эта работа предполагает еще и критику недостатков методов, однако в науке не принято критиковать, если нет альтернативных подходов. В докладе дано описание ряда недостатков классической математической статистики, например трудоемкость верификации законов распределения вероятностей для изучаемых данных и независимости результатов экспериментов в случае продолжающихся испытаний. Эти и другие несовершенства математической статистики способствуют созданию потенциальных возможностей для участия философов в исследованиях в области анализа данных в качестве критиков.

Имеются ли философские концепции, представляющие интерес для анализа данных? Ответ на этот вопрос положительный, так одна из целей работы заключается в демонстрации

уменьшения трудоемкости верификации независимости в случае описания изучаемых связей с помощью причинных отношений. Вторая цель работы, связанная с первой, заключается в описании естественных предпосылок для плодотворного участия философов, имеющих подготовку в формальных науках для исследований в области анализа данных как критиков «узких мест» анализа данных и консультантов по правильной интерпретации причинных отношений в задачах анализа данных. Кроме того, изучается адекватность современных философских новаций, в частности концепции понимания для исследований в области экспертных систем.

2 Актуальность методологических исследований применения математики

Судя по литературе, научное сообщество в целом положительно оценивает влияние классической философии от античной философии до И. Канта на интеллектуальное развитие и культурный уровень исследователей. Однако по отношению к значимости философии для текущих научных исследований не существует консенсуса. Так, известный физик нобелевский лауреат С. Вайнберг отрицает значимость философии и посвятил этому вопросу главу в своей книге [1]. Однако выдающийся физик А. Эйнштейн утверждал об особом влиянии Э. Маха и Д. Юма в контексте окончательного формулирования частной теории относительности [2]. Современный математик Г. Шейфер писал, что значимость философских исследований ограничена классическими вопросами науки и не затрагивает современную науку. Однако по нашему мнению в современной науке имеются потенциальные вакансии для участия философов в научной деятельности, связанные с критикой несовершенств научного знания, с консультированием исследователей не философов по корректному использованию философских концепций, а также с привлечением концепций объяснения и понимания для анализа научного знания. Начнем с критической деятельности, дело в том, что по нашему мнению, в науке имеется “ничья земля”, это область критики текущего научного знания в контексте его применимости.

Отметим, что проблема применимости знания до сих пор не является особо популярной ни у математиков, ни у философов. Хотя еще в прошлом столетии выдающиеся математики и философы обосновали значимость применимости знания. Так, независимо друг от друга два известных философа Г. Райл и А. Полани продемонстрировали, что понимание изучаемых объектов достигается в результате применения знания. Выдающийся математик А.Н. Колмогоров отмечал, что успехи в основаниях математики заслонили собой самостоятельную проблему применимости математики. Почему проблема применимости знания не оказывается популярной? Так у чистых математиков проблема применимости не особенно популярна, так как по признанию чистых математиков, их исследования существенно связаны с красотой математических объектов, однако, по их мнению, красота не присуща приложениям математики. Отметим, что в целом проблема применимости формальных методов не является популярной и у философов. Так, они предпочитают изучать теории, а не их использование. Знаменитый, недавно ушедший из жизни философ Я. Хакинг, подробно описал причины непопулярности проблемы применимости знания в философском сообществе [3].

Отметим, что со временем интерес к проблеме применимости математики возрастает. В настоящее время в философском сообществе достаточно популярны две проблемы, связанные с применением математики. Во-первых, проблема принципиальной неустранимости математики из наиболее обоснованных современных теорий. Во-вторых, сформулированная нобелевским лауреатом Е. Вигнером, проблема непостижимой эффективности математики в естествознании. Отметим, что эти оба направления исследований связаны с применением формальных теорий, однако в подавляющем множестве научных дисциплин знание не является полностью формализованным, и математика применяется непосредственно к изучаемым данным.

Применение знания, и в частности применение формальных методов является областью потенциально интересной для философских и методологических исследований. Почему область применимости знания оказывается потенциально интересной для философских исследований? Дело в том, что знание в целом и формальные методы в частности не являются совершенными. Кроме того, в науке не принято критиковать используемые методы, если нет альтернативных подходов. Критика философами несовершенств математических методов может способствовать улучшению существующих методов и созданию новых более совершенных методов, поэтому философские исследования представляют интерес и для математиков. Известный философ Кайберг отмечал, что проблема применимости знания интересует исключительно методологов и бизнесменов, добавим к этому перечню и прикладных математиков, участвующих в

ответственных испытаниях. Однако в основном они публикуются в специальных журналах, а что касается монографий, то в известной отечественной литературе последняя солидная монография методологического характера вышла в 80 гг. [4]. Поэтому проблема применимости математики оказывается актуальной.

3 Трудности формальной верификации по данным их независимости и законов распределения

Обратимся к математической статистике, эта дисциплина наиболее популярна в приложениях, и покажем, что ее методы в контексте приложений не свободны от некоторых недостатков. Напомним, что для того чтобы решить большинство проблем классической математической статистики необходимо предварительно решить две задачи, во-первых описать распределения случайных величин, описывающих результаты исследований и во-вторых, определить как эти случайные величины связаны, в простейшем случае они независимы.

3.1 Проблемы формального определения законов распределений по изучаемым данным

В современных исследованиях законы распределения вероятностей определяют с помощью специальных компьютерных программ на основе построения гистограмм по результатам испытаний, а также с помощью специальных функций. Если распределения определяются для продолжающихся испытаний с большими объемами результатов, то исследования такого плана оказываются весьма трудоемкими. В результате исследований определяются гипотетические распределения вероятностей, и заключительным этапом определения распределений оказывается проверка соответствия этих распределений изучаемым данным с помощью специальных критериев согласия. Одним из популярных критериев является критерий Пирсона хи-квадрат. Его популярность обязана универсальности, так как он подходит для верификации как дискретных, так и непрерывных распределений. Критерий был предложен К. Пирсоном, знаменитым математиком, генетиком, философом в 1900 году. Критерий не является в полной мере адекватным для приложений, так как в его реализации предполагается группирование данных. Дело в том, что в случае бесконечного объема данных не имеет значения, каким образом произведено группирование, однако для реальных конечных объемов специфика группирования может иметь значение, так как не исключено, что при определенном разбиении данных проверяемая гипотеза не будет отвергнута, а в случае другого разбиения она будет фальсифицирована. Проблема заключается в том, что в статистике нет понятия объективного группирования. Известно, что специалисты по распознаванию образов обратили внимание на эту проблему, и возможно, удастся получить ее решение без использования группирования. Трудоемкость определения распределений на основе эмпирических испытаний и недостатки формальных критериев приводят к естественному вопросу о возможности верификации распределений на основе рассуждений по аналогии или соображений общего характера.

3.2 О верификации распределений на основе неформальных рассуждений

Рассмотрим на примерах обоснованность неформальных подходов по определению распределений вероятностей. Начнем с рассуждений общего характера. Известно, что Распределение Гаусса или как его часто называют ‘нормальное’ распределение наиболее часто встречается как в теоретических исследованиях теории вероятностей, так и в приложениях математической статистики. Тогда возникает вопрос, является ли рациональным использование нормального распределения или другого часто встречающегося в конкретном исследовании распределения на основании их популярности? В целом такое предположение не является основательным, так как для многих положений дел было доказано, что для исследуемых феноменов адекватными оказывались распределения абсолютно негауссова типа. Теперь рассмотрим обоснованность принятия законов распределения на основе рассуждений по аналогии. Предположим, что для изучения поведения некоторого сообщества животных оказалось адекватным конкретное распределение вероятностей. Тогда возникает вопрос об адекватности этого распределения для описания поведения близкородственных животных. По нашему мнению, если первоначальное распределение было получено теоретически, тогда аналогия вполне допустима. Однако если оно было получено в результате подбора распределения по данным, тогда это распределение в лучшем случае имеет гипотетический статус для использования в других ситуациях.

3.3 Проблемы формального определения независимости результатов испытаний

Теперь перейдем к анализу независимости в изучаемых данных. Отметим, что исследование независимости многократно сложнее верификации распределений по данным. В чем заключаются сложности исследования независимости? Во-первых, не существует унифицированного индикатора независимости, так известные индикаторы независимости имеют контекстуальный характер, в частности их использование определяется распределением случайных величин. Если исследуемые случайные величины имеют совместное нормальное распределение, тогда адекватным индикатором независимости оказывается попарный коэффициент корреляции. Во-вторых, неизвестны надежные индикаторы независимости. Так, например если случайные величины независимы, то попарный коэффициент корреляции равен нулю, однако если он равен нулю, то случайные величины не обязательно оказываются независимыми. В-третьих, методы определения независимости связаны с количеством случайных величин, по отношению к которым определяется характер их связанности. Так, для верификации независимости в случае двух случайных величин адекватны таблицы сопряженности. Однако для исследования независимости результатов бросания монеты, таблицы сопряженности не подходят, так как результаты описываются не двумя, а одной случайной величиной. И, наконец, последнее по порядку, но не по степени значимости. Трудоемкость определения независимости значительно превосходит сложность верификации распределений вероятностей. Дело в том, что если получены одинаковые распределения вероятностей для данных на некоторых выборках, то итоговое распределение для всех данных обычно совпадает с распределениями на выборках. Однако в случае исследования независимости некоторой популяции данных, результаты о независимости различных частей этой популяции не имеют никакого значения. Впервые пример такого плана был предложен известным математиком С.Н. Бернштейном. В предложенном примере Бернштейна случайные величины являются вершинами треугольника, причем попарно все случайные величины независимы, однако все вместе они оказываются зависимыми величинами. В дальнейшем много раз было продемонстрировано, что свойство независимости, которым обладают подмножества множества данных, не обязательно имеет место для всего объема данных. Таким образом, независимость оказывается принципиально интегральной характеристикой. Современные математики Г. Шейфер и В. Вовк отмечали чрезвычайную сложность определения независимости в случае продолжающихся испытаний [5].

3.4 О неформальной верификации независимости

Так как формальная верификация независимости оказывается чрезвычайно сложной, то возникает вопрос о возможности принятия независимости на основе неформальных соображений. Рассмотрим на примерах убедительность использования независимости на основе неформальных рассуждений, опирающихся на интуицию. В качестве первого примера рассмотрим принятие независимости на основе контроля фоновых условий проведения экспериментов. Отметим, что ряд ученых и философов возражали против апелляции к фоновым условиям. Так, например, известный философ, одновременно успешно занимавшийся наукой, П. Суппес полагал, что апелляция в рассуждениях к фоновым условиям часто оказывается несостоятельной, т.к. они практически никогда полностью неизвестны. Специалист в области прикладной математики Ю.И. Алимов утверждал о несостоятельности контроля фоновых условий за пределами физики [6]. Однако и в физике возникают проблемы с контролем условий проведения экспериментов. Так, из истории физики известно, что при открытии радона в лаборатории Резерфорда все приборы зашкаливали. Только после того, как догадались, что открытый элемент является газом, и снова провели рафинирование условий проведения экспериментов, то приборы вернулись в нормальное состояние.

В качестве второго примера, апеллирующего к интуиции, рассмотрим принятие модели независимых испытаний для случайных величин с ничтожной корреляцией. Пример взят из работы Эльясберга, известного теоретика в области космонавтики [7]. В этом примере рассмотрены два множества случайных величин, имеющих нормальное распределение, мощности этих множеств равны 1000. В первом множестве для всех случайных величин коэффициент попарной корреляции равен 0,01, а во втором множестве все случайные величины попарно независимы. Для каждого множества определялась сумма случайных величин, и для нее оценивалась дисперсия. Оказалось, что дисперсии отличались больше чем на порядок, а в случае, если мощности множеств равны 10000, то дисперсии разнятся более чем на два порядка. И, таким образом переход к модели независимых испытаний оказывается полностью

несостоятельным. Фактически нами обоснована необходимость проверять условия применимости стохастической математики исключительно формальным образом.

3.5 О целесообразности разработки нормативных требований к применению стохастической математики

Отметим, что в настоящее время требования к применению математической статистики имеют рекомендательный, а не нормативный характер. А так как формальная верификация условий применимости математической статистики сложна, то часто ее игнорируют, ограничившись неформальным анализом. В связи с этим для философов, обладающих знаниями методов математической статистики в контексте приложений, открывается потенциальная возможность участия в научной деятельности совместно с математиками, которая связана с анализом известных рекомендаций по применению методов математической статистики с целью возможной их модификации и приданию им нормативного статуса, а также участие в мониторинге за выполнением этих требований.

4 Философские концепции в научных исследованиях

4.1 Введение и использование свойства независимости на основе принципа общей причины

Среди верификаций условий применения классической математической статистики проверка независимости случайных величин оказывается наиболее сложной проблемой. Как известно, А.Н. Колмогоров отмечал особую значимость свойства независимости для возникновения, становления теории вероятностей и ее отделимости от теории меры, однако, по его мнению, в рамках математики невозможно определить условия, при которых целесообразно использование моделей независимых экспериментов [8]. А.Н. Колмогоров предполагал, что эта проблема скорее относится к философии естествознания [Там же].

По существу, предположение Колмогорова является вызовом философскому сообществу. В России идея Колмогорова вызвала отклик у Ю.В. Сачкова, он показал значимость идеи независимости для ряда наук и философии, при этом он отметил, что проблема независимости не была замечена философским сообществом.

Возникает вопрос: “Имеются ли в философии ресурсы для развития понятия независимости?” По-нашему мнению, причинные отношения являются естественным инструментом для исследования независимости. Действительно, если А причина В, то А не зависит от В. Это элементарный ход, более серьезный подход к описанию независимости в рамках причинных отношений был предложен Г Рейхенбахом. Он сформулировал принцип общей причины, согласно этому принципу, если любые два события А и В связаны, то либо А причина В, либо В причина А. Однако, если это не так, то существует событие С, которое является их общей причиной, при этом А и В становятся независимыми. Принцип общей причины использовали философствующие физики и философы, изучающие квантовую физику, в частности П. Суппес для исследования условий, при которых скрытые причины имеют вероятностный или детерминистский характер [9]. Современные философы и философствующие физики продолжают исследования, связанные с принципом общей причины [10].

Отметим, что в ряде направлений современной науки, например в экспертных системах, популярно использование причинных отношений. Однако прежде чем описывать используемые в науке причинные отношения, уделим некоторое внимание их значимости в тех научных областях, где они редко используются или где они не используются, но могут потенциально быть использованными. Так, например, в анализе данных причинные отношения не часто используются, так как не любые положения дел допускают причинные формализации. Предположим, что для исследуемой проблемы в рамках анализа данных получено формальное описание с помощью причинных отношений. Многие проблемы в рамках статистического анализа данных исследуются с использованием независимости результатов испытаний. В эмпирических продолжающихся испытаниях при получении новых результатов верификация независимости осуществляется для совокупного объема данных, включающего как прошлые, так и вновь полученные результаты. В случае же определения независимости на основе теоретических описаний верификация независимости существенно упрощается. Так как причинные описания имеют теоретический статус, то при обнаружении, что результаты испытаний в рамках причинных отношений оказываются независимыми, нет необходимости в случае продолжающихся испытаний и получении новых результатов обязательно проверять

наличие независимости для всего объема данных. Проверка независимости имеет значение в случае изменения причинных отношений.

Причинные отношения использовались современным математиком Г. Шейфером при конструировании обобщенной вероятностной интерпретации [11], в работе известного философа Н. Картрайт была дана критика концепции Шейфера [12]. Так как, по мнению Картрайт его концепция верна только в частных случаях.

4.2 Современные концепции понимания де Регта и К. Халифы

Мы показали значимость участия философов в научных исследованиях, на примере анализа данных, в качестве критиков несовершенств научных методов и консультантов по корректному применению философских концепций в науке. Другая возможная деятельность философов в науке связана с участием вместе с математиками по разработке нормативных рекомендаций с целью корректного применения математических методов, а также участие в мониторинге за соблюдением этих требований в приложениях математики. Получается, что в целом деятельность философов в науке имеет критический характер. Возникает вопрос: ” Имеются ли в арсенале философов позитивные новации для науки?” Ответ на этот вопрос, положительный, так как одним из направлений деятельности философов являются работы по проблемам объяснения и понимания, которые представляют интерес для науки. Отметим, что публикации по проблемам объяснения, охватывающие дедуктивно - номологическую схему К. Гемпеля, схему практического силлогизма Е. Анскомб, а также теологическую и кибернетическую схему объяснения достаточно хорошо представлены в литературе [13].

В тоже время, проблема понимания в известной литературе не получила должного внимания, поэтому остановимся на этой проблеме. Раньше проблема понимания не имела самостоятельного статуса, так как считалось, что понимание возникает в результате сведения изучаемой проблемы к уже ранее решенной проблеме или в результате успешных объяснений. Со временем философское сообщество пришло к заключению, что проблема понимания представляет самостоятельный интерес и вне объяснений, кроме того, если в философии вполне естественной является апелляция к объяснениям, то в науке исследователи скорее говорят о понимании на основе проведенных экспериментов или в результате применения научных методов, а не на основе объяснений. В настоящее время в философии науки интенсивно разрабатываются концепции понимания без объяснений. Я кратко остановлюсь на концепциях де Регта и К. Халифы, т.к. первая из них в большей степени ориентируется на науку, а во второй разрабатывается идея уровней понимания, которая значима для нашей работы. Сначала обратимся к концепции де Регта, который получил престижную премию имени Лакатоса за работы по данной теме и книгу [14].

Основные идеи де Регта заключаются в том, что значимые научные результаты достигаются с помощью теорий, а чаще посредством модифицированных моделей этих теорий на основе изучаемых феноменов, которые являются понятными для исследователей. Свойство понятности не является внутренним свойством теорий, им его наделяют ученые. Так, теория оказывается понятной, потому, что способности и интеллектуальные навыки исследователя подходят для того чтобы изучить и в совершенстве овладеть этой теорией. Согласно де Регту, изучаемые положения дел становятся понятными на основе применения понятных теорий. Ключевой составляющей теорий понимания являются критерии понимания. Они формулируются в предположении, что изучаемая проблема получила решение. Один из критериев де Регта основан на том, что решение проблемы получено на основе применения формальных методов. Пусть на основе модели теории составлено уравнение, описывающее поведение изучаемых явлений. Тогда эта модель оказывается понятной, и, соответственно изучаемые на ее основе явления оказываются понятными, если исследователь способен получить качественное решение уравнения, до того, как оно фактически решено. Отметим, что базовая идея этого критерия была ранее сформулирована известным физиком Р. Фейнманом, даю ссылку по [14, P. 102]. Дополним концепцию де Регта понятием уровня понимания, эта понятие разрабатывается в концепции К. Халифы [15]. Будем считать, что получение сильного идеального решения задачи свидетельствует о сильном ее понимании. Поясним понятие уровня понимания на примерах из физики. В нашей работе критерий сильного понимания совпадает с ранее описанным критерием Фейнмана. Фейнман не дал описания ни характера решенного им уравнения, ни особенностей полученных им решений этого уравнения, которые привели его к пониманию изучаемых физических явлений природы. Возникает предположение, что решаемые в этом случае уравнения допускали или полное аналитическое решение или такое решение в значимом частном случае. Наша гипотеза подтверждается соображениями современного известного физика и электронщика Д. Абботта. Он утверждал, что когда физика занималась изучением

макроэлектронных явлений, то они были глубоко поняты, и для уравнений, описывающих эти явления, были получены аналитические решения, при этом не было необходимости в использовании компьютеров. И, далее он продолжал, сейчас физики изучают микроэлектронные явления, которые являются очень сложными, и они до сих пор не столь глубоко изучены, при этом уравнения, описывающие явления в микроэлектронике не удается решать аналитически, и они допускают только численные решения [16]. Критерии Фейнмана и Абботта естественно считать критериями сильного понимания. Критерий среднего, вполне достойного понимания в области точного естествознания поясним на примере. Предположим, что изучаемые положения дел описываются уравнениями, которые допускают итеративное решение. Если исследователь на основе нескольких итераций приходит к качественно правильному решению, до того, как уравнения фактически решены, то естественно считать, что получено решение хорошего качества, что свидетельствует о весьма достойном, не хуже среднего уровня понимания изучаемой проблемы. Теперь комбинированную концепцию понимания на основе концепций де Регта и Халифы попробуем использовать для концепций в области искусственного интеллекта.

4.3 Искусственный интеллект новая область науки для применения философских идей и участия в ней философов

В области искусственного интеллекта используются философские концепции для различных целей, например принцип общей причины, в ряде приложений он играет техническую роль для получения независимых переменных. В тоже время в некоторых системах искусственного интеллекта, например в байесовских сетях, причинные интерпретации, по мнению их разработчиков, в частности Перла, играют сущностную роль. Он отмечает три фактора, которые препятствуют компьютерным программам достижению уровня интеллекта, сопоставимого с интеллектом человека, это отсутствие адаптивности, умения объяснять полученные предсказания и понимания причинно-следственных связей [17]. Перл пишет: «Этот признак человеческого познания, на мой взгляд, является необходимым (хотя и недостаточный) ингредиент для достижения интеллекта человеческого уровня» [17, P. 54]. Отметим, чем больше философских концепций используют исследователи, которые не являются философами, тем больше создается потенциальных условий для работы философов в науке, т.к. не философы далеко не всегда осваивают философский аппарат на должном уровне.

4.3.1 Концепции понимания применительно к искусственному интеллекту

Цель исследования заключается в изучении адекватности базовых идей концепции понимания и в предварительном оценивании пользы от их применения к решению проблем на основе систем искусственного интеллекта. В качестве пробного шага была выбрана схема функционирования системы искусственного интеллекта, построенная в рамках задачного подхода [18]. Концепция задачного подхода была разработана в связи с исследованием влияния ограничительных теорем К. Геделя на работающую математику в совместных работах логиков, известного математика академика Ю.Л. Ершова и доктора философских наук К.Ф. Самохвалова [19]. Они показали, что корректность каждой отдельной математической дисциплины обосновывается независимо от этих теорем, поэтому теоремы Геделя не имеют особого значения для используемой математики. Кроме того, они продемонстрировали, что значимость математической теории определяется множеством эффективно решаемых задач на ее основе. Оказалось, что концепция задачного подхода, ориентированная на исследование проблем теоретического характера, имеет прагматическую значимость. Так, она была успешно использована для приложений в области семантического программирования [20]. Кроме того, в совместной работе Е.Е. Витяева, С.С. Гончарова и Д.И. Свириденко предложена адаптация задачного подхода для исследований в области искусственного интеллекта [18]. В этой работе была предложена схема решения проблем общего характера, охватывающая многие области знания, включая искусственный интеллект.

Опишем значимые для нашего исследования, составляющие этой схемы, и дадим им краткое пояснение. Первый блок носит название потребность, здесь не нужны особые объяснения. Второй блок называется противоречие, здесь речь о несоответствии между доступными способами удовлетворения потребности и желаемыми способами. Третий блок называется задача. Это базовый блок схемы, задача формулируется таким образом, что в случае ее решения потребность будет удовлетворена способом, полностью или, по крайней мере, частично удовлетворяющим заказчика. Для дальнейшего изложения является значимым понятие

минимального понимания задачи, сформулированное Ершовым и Самохваловым: «…мы *понимаем* задачу только тогда, когда ей сопоставили обоснованное чувство уверенности в том, что всякое состояние нашего сознания мы сумеем убедительным и безошибочным образом распознать как такое, когда решение найдено, или как такое, когда решение не найдено» [19].

Для применения концепции понимания к схеме решения проблем в области искусственного интеллекта необходимо осуществить согласование базовых терминов этой концепции и схемы. Напомним центральные термины концепции понимания. Это понятная теория или модель и понимание изучаемых явлений на основе понятной модели. Вполне естественно полагать, что решение задачи на основе понятной модели приводит к пониманию изучаемых явлений. Так как в концепции понимания нет понятия минимального понимания, то для более полного согласования концепции решения проблем в искусственном интеллекте на базе задачного подхода и комбинированной концепции понимания включим в последнюю понятие минимального понимания из задачного подхода. Имеют ли рассмотренные нами концепции прагматическую значимость? Необходимо подчеркнуть, что создатели этих концепций не преследовали прагматические цели, однако отсюда не следует, что эти концепции, априори, не представляют интереса для исследователей из различных областей знания. Поэтому предварительное оценивание значимости концепций для науки представляется небесполезным делом. Некоторые аргументы в пользу значимости концепций понимания таковы. Во-первых, знакомство исследователей с критериями понимания в различных областях знания может способствовать созданию критериев понимания в интересующей исследователей области знания. Во-вторых, знание критериев понимания в определенной области знания, которые учитывают уровень понимания, а также определение интеллектуальных, материальных и д.р. трудозатрат для достижения соответствующего уровня понимания, способствуют разработке рациональных планов по решению проблем определенного качества, соответствующего искомому уровню понимания решаемой проблемы.

5 Заключение

Показана значимость применения философских концепций причинности и понимания для исследования некоторых проблем в области анализа данных и искусственного интеллекта. Так, в случае решения задач, в области анализа данных, допускающих причинную интерпретацию, существенно уменьшается трудоемкость верификации законов распределения и особенно независимости результатов испытаний. Кратко описаны философские концепции понимания и показана адекватность этих концепций для исследования некоторых проблем в области искусственного интеллекта. Так как в науке не принято критиковать недостатки используемых методов, если им нет альтернативных подходов, а также в связи с использованием в анализе данных и искусственном интеллекте концепции причинных отношений, обоснованы перспективы участия философов, которые владеют научными методами, в этих областях знания в качестве критиков несовершенств этих наук и консультантов по корректному применению концепции каузальности.

Литература

[1] Вайнберг С. Мечты об окончательной теории. – М.: Едиториал УРСС, 2004.

[2] Norton J. How Hume and Mach helped Einstein Find special relativity // Discourse on a new method. Edited by Domsky M. and Dicrson, M. – Chicago: Open Court, 2010.

[3] Hacking I. Representing and intervening. – Cambridge University Press, 1983.

[4] Блехман И.И., Мышкис А.Д., Пановко Я. Г. Прикладная математика: предмет, логика, особенности подходов. – Екатеринбург: УРСС, 2018.

[5] Shafer G., Vovk V. The Sources of Kolmogorov's Grundbegriffe // Statistical Science. – 2006. – Vol. 21, № 1. –P. 70–98.

[6] Алимов Ю.И. Альтернатива методу математической статистики. – М.: Знание, 1980.

[7] Эльясберг П.Е. Вычислительная информация: Сколько ее нужно? Как ее обрабатывать? – М.: Наука, 1986.

[8] Колмогоров А.Н. Основные понятия теории вероятностей. – М.: Наука, 1974.

[9] Suppes P. Probabilistic metaphysics. – Oxford: Basil Blackwell Publisher Ltd, 1984.

[10] Hofer-Szado G., Redei M., Szado L. The principle of the common cause. – Cambridge University Press, 2013.

[11] Shafer G. The Art of Causal Conjecture. – Cambridge MA: MIT Press, 1996.

[12] Cartwright N. The Dappled World. – Cambridge University Press, 2003.

13] Вригт Г. Х. Логико-философские исследования. – М: Прогресс, 1986.

[14] De Regt X. Understanding scientific understanding. – Oxford University Press, 2017.

[15] Khalifa K. Understanding, explanation and scientific knowledge. – Cambridge University Press, 2017.

[16] Abbott D. The Reasonable Ineffectiveness of Mathematics // Proceedings of the IEEE. – 2013. – Vol. 101, № 10. – P. 2147-2153.

[17] Pearl J. The Seven Tools of Causal Inference, with Reflections on Machine Learning // Communications of the ACM. –2019 – Vol.62, № 3. P. 54-60.

[18] Витяев Е.Е., Гончаров С.С., Свириденко Д.И. О задачном подходе в искусственном интеллекте // Сибирский философский журнал. – 2019. – Т. 17. – № 4. – С. 5–25. DOI 10.25205/2541-7517-2019-17-4-5-25.

[19] Ершов Ю.Л., Самохвалов К.Ф. Современная философия математики: недомогания и лечение. – Новосибирск: Параллель, 2007.

[20] Гончаров С. С., Свириденко Д. И. Семантическое моделирование и искусственный интеллект // Сибирский философский журнал. –2018 – Т. 16 , № 4. – С. 5–25. DOI 10.25205/2541-7517-2018-16-4-5-25.

Знания-Онтологии-Теории (ЗОНТ-2023)

Процессная онтология и квантование информации

Суров Илья Алексеевич

Университет ИТМО, Кронверкский пр., д. 49 лит. А, г. Санкт-Петербург, 197101, Россия.

ilya.a.surov@itmo.ru

Аннотация. *Современные методы построения онтологий, семантических и знаниевых карт ограничены узкоспециализированными приложениями, за рамками которых задача структурирования информации остаётся нерешённой. Причина этой проблемы состоит в ограниченности существующих алгоритмов статической логикой сущностей, в основе которой лежит операция классификации. В докладе представлен подход к развитию онтологического моделирования на основе процессной логики. При этом первичным элементом модели является циклический процесс, разворачивающийся в последовательность трёх функциональных этапов: распознавание новости и постановка по отношению к ней субъективных целей (1), действия и мероприятия по достижени этих целей (2), а также внедрение, апробация и оценка полученного результата (3). Такая последовательность, известная в системотехнике как жизненный цикл, используется в качестве шаблона, наполняемого конкретными информационными блоками. Эта структура позволяет упорядочивать информацию в виде процессных единиц, содержащих не только объективно данные информационные блоки, но и субъективно установленные между ними причинно-следственные связи. Показано, что этот подход фактически используется в кибернетике, сценаристике и режиссуре, образовании и осмыслении событий в повседневном мышлении. Представленная онтология и её математическая формализация позволяют ввести задачу процессного моделирования в научное поле. Сопряжение процессной и статической логик открывает возможности для создания алгоритмов и систем анализа данных на принципах естественного мышления.*

Ключевые слова: процесс, онтология, причинность, жизненный цикл, семантика, квант

1 Введение

Для использования растущих объёмов информации необходимо её упорядочивание. В настоящее время эта задача решается путём построения онтологий предметных областей [1]–[3], карт знаний и семантических сетей [4]–[8], в том числе в задачах поиска и анализа данных [9]–[13]. За многолетнюю историю развития, однако, алгоритмы этих типов не достигли эффективности, необходимой для широкого практического использования [14]. Причина этого состоит, по всей видимости, не в недостатке вычислительных мощностей или несовершенстве программных реализаций. Для решения проблемы желательно выявить коренной недостаток отмеченных подходов.

Подход к такому решению намечен в работах [15]–[17], исследующих различные логики в мышлении человека. Среди них важное место занимает т.н. *логика существования,* которая представляет мир как множество сущностей. Стержневой операцией этой логики является классификация предметов по определённым признакам ([17], стр. 23): мяч, игрушка, спортивный снаряд, предмет круглой формы и т.д. Именно эта операция подведения под класс лежит в основе цитированных методов упорядочивания, например, при размещении информации в многоуровневой категориальной системе – онтологии, на карте знаний или в семантической сети. Как показано авторами (там же), этот выбор обусловлен установкой разработчиков на пространственно-временное осмысление мира, рассматривающее его как

статический контейнер инертных сущностей. Эта логика (характерная для западного, греческо-европейского мышления) обусловила принципы построения и архитектуру современных информационных систем.

Другим важнейшим типом логики является *процессная логика*, основанная на понятии действия [16]–[18]. В отличие от статических сущностей С-логики, в процессной логике первично действие – непрерывный процесс, который разворачивается в пространстве-времени и наполняется событиями по мере развития. В повседневном мышлении процессная П-логика применяется примерно также часто, как и С-логика [17], что свидетельствует о её практической ценности. Этот вывод подтверждается результатами когнитивных исследований, согласно которым основной задачей естественного мышления является предсказательное моделирование деятельности организма [19]–[21]. При этом способность процессной логики связыть события и факты причинно-следственными отношениями отвечает задаче прагматичного структурирования информации.

В современных системах анализа данных, однако, процессная логика используется редко. Примером такого подхода является метод моделирования технологических и бизнес-процессов, в котором события и потоки информации структурируются в виде графовых и стрелочно-блочных схем [22]–[26]. Недостатком этого подхода является неограниченное многообразие и сложность процессов, затрудняющие его масштабирование и автоматизацию. Нехватка достаточно универсального формализма для описания разнородных процессов и соответствующих правил рассуждения обуславливает фактическое отсутствие процессной логики в научном поле. В настоящей статье представлено решение этой проблемы. Используемая для этого онтология процесса [27] описана в разделе 2. Вытекающий из неё метод структурирования информации и его свойства рассмотрены в разделе 3.

2 Онтология процесса

2.1 Универсальные процессные состояния

Процесс обычно определяется как последовательность переходов между состояниями системы. На практике такое определение малополезно из-за неопределённости того, какие состояния следует рассматривать: в одном процессе это могут быть эмоциональные состояния человека, в другом — агрегатные состояния вещества, в третьем — состояния экономики государства. Кроме того, такие последовательности состояний не имеют ни начала ни конца – одна непрерывная целостность, бесконечные потоки событий, мало помогающие в задаче упорядочивания информации.

Для единообразной формализации процессов в приведённом определении нужно уточнить состояния так, чтобе обе отмеченные проблемы не возникли. Такие состояния должны быть, во-первых, достаточно универсальными, т.е. применимыми к широкому спектру процессов. Во-вторых, они должны ограничивать потоки событий практически целесообразным образом. Наиболее простой набор таких состояний это *начало*, *середина* и *конец*. Эти состояния легко выделимы если не во всех, то в оцень многих практически важных процессах: жизнь организма, научно-исследовательский проект, семинар, экспедиция или поход, чаепитие и т.д. и т.п.

Интуитивная понятность этих состояний указывает на их ценность в естественном мышлении. Эта ценность, разумеется, не сводится к формальной хронометрии. Знание о том, что процесс находится в своём *начале*, *середине* или *конце*, значимо из-за функций этих состояний:

1. *Начало*: новость
   В начале любого процесса различается новое, изучается предметная область, определяются субъективные мотивы и проблемы, ставятся цели и задачи, разрабатываются планы по их решению.
2. *Середина*: действие
   В середине процесса происходит основное действие, проводятся мероприятия по решению поставленных задач, выполняются планы по достижению намеченной цели, преодолеваются непредвиденные препятствия.
3. *Конец*: результат
   В конце процесса происходит оформление, апробация и оценка полученных результатов, внедрение их в предметную область, делаются выводы о проделанной работе.

Например, знание о том что человек находится в *начале* аспирантуры говорит о том, что сейчас формулируются задачи работы, соискатель скорее всего знакомится с научным коллективом, предметной областью и осваивает методы исследования. На этом этапе уместно сделать обзор литературы, тогда как спрашивать о полученных результатах ещё рано. Аналогичную смысловую нагрузку несут *середина* и *конец* процесса. Функциональное своеобразие этих состояний позволяет кратко кодировать большие объёмы значимой информации, что и объясняет широту их использования в естественном мышлении.

Описанная функциональность состояний *начало*, *середина* и *конец* соответствует последовательностям

- проектирование - реализация - рефлексия [28], [29] и
- мышление (think) - действие (act) - чувство (sense) [30], [31]
- анализ – творчество – синтез [32]

задающим трёхактную структуру процессов управления, дизайна, взаимодействия индивида с окружающей средой, и других [27].

Таблица 1. Соответствие процессных категорий в различных предметных областях.

Предметная область / контекст	***Начало*** ***Новость, проблема***	***Середина*** ***Действие, мероприятие***	***Конец*** ***Результат, итог***
Кибернетика [30]	Мышление	Действие	Чувство
Дизайн [32]	Анализ	Творчество	Синтез
Промышленный цикл [33]	Постановка задач, планирование	Исполнение, производство	Обратная связь, оценка, выводы
Сложная деятельность [28]	Проектирование	Реализация	Рефлексия

Возможны и другие разбиения процесса на этапы. Одним из наиболее известных являются различные вариации т.н. модели Деминга, согласно которым процесс (промышленной деятельности) разбивается на 4 фазы: разработка и планирование (plan), исполнение и производство (do), продажи и оценка (check), рефлексия и выводы (act) [33]. Последние два этапа в этом списке, однако, сходны по содержанию; очень близки также глаголы do и act (уместность последнего для обозначения соответствующего этапа сомнительна). В этой связи указанные этапы не делят процесс на равнозначные и уникальные части. Это может быть оправдано технологически (например, для выделения продаж в отдельную фазу), однако для универсальной задачи структурирования информации такая асимметрия нежелательна. Минимальный набор функционально разнокачественных частей процесса содержит именно три фазы, описанные выше. В промышленности они соответствуют (позже преобразованной Демингом) модели Шеварта, состоящему из определения целей (specification, 1), производства (production, 2) и оценки результатов (inspection, 3) [33]. При необходимости более подробного описания число этапов удваивается с сохранением функциональной симметрии [27].

2.2 Функциональное замыкание

На практике последовательность этапов обычно повторяется так, что после оценки результатов (этап 3) следует постановка новых целей (этап 1). В промышленности таким образом идёт работа по непрерывному улучшению старых или созданию новых продуктов. Эта эффективная на практике функциональная логика отражена в циклической структуре упомянутых выше моделей промышленного процесса [33].

То же самое верно для произвольного процесса. Оценка полученного результата подразумевает его сравнение с ожиданием, выраженным в целях на начальном этапе процесса. Вывод на основе такой оценки (этап 3) обычно предшествует началу (этап 1) следующего аналогичного цикла: рабочего дня, исследовательского проекта, следующего матча спортивной команды и т.д. При этом линейная последовательность процессных этапов замыкается в кольцо как показано на Рис. 1.

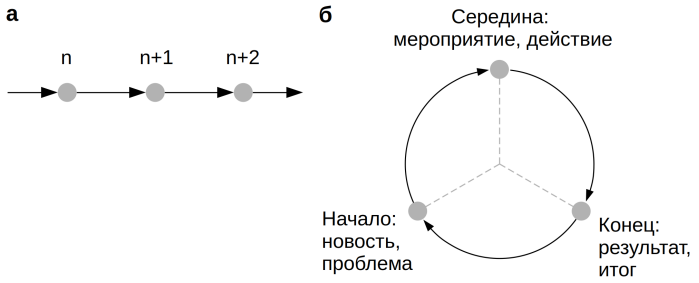


Рис. 1. а: Линейная модель процесса как бесконечная последовательность состояний.
б: Круговая модель процесса на основе циклического замыкания функциональных состояний.

Функциональность элементов такой модели аналогична структуре «Если C, сделай D, чтобы G», представляющей собой минимальное описание целесообразного поведенческого алгоритма [34]. В отличие от линейной структуры C → D → G (там же), однако, целевое состояние G (этап 3) направлено не просто куда-то в будущее, а *отвечает* на исходное событие – новость C (этап 1). Замыкающая цикл на Рис. 1(б) нижняя стрелка отражает эту процессно-диалектическую природу естественного мышления [35]. Переход от линейной модели процесса к круговой соответствует переходу от отстранённо-описательного к субъективно-целесообразному структурированию информации.

Круговая модель процесса известна как *жизненный цикл*, используемый для моделирования динамики научно-исследовательских, производственно-технологических, биологических, экономических, социальных, информационно-психологических и других процессов [36], [37]. Все эти процессы проходят через одни и те же функциональные состояния, названия которых определяются своеобразием соответствующих систем. Простейшими примерами такого повторяющегося процесса являются гармонические колебания, в также суточные и годовые циклы, в которых начало соответствует ранней весне, действие – лету, а результат – поздней осени [27]. Из этих примеров видно, что цикличность процессной структуры не означает пассивного просмотра одних и тех же событий как в повторяющемся кино: этапы проектов, времена года и суток повторяются, однако их содержание может меняться.

2.3 Математическая модель

Плоскость на Рис. 1(б), на которой изображён процесс, образована семантическими факторами Ч. Осгуда *сила* и *активность* [38] следующим образом [27]:

1. Фактор *активности* есть вертикальная ось Y, различающая процессные состояния по степени их внешней активности. Наибольшая активность наблюдается при достижении поставленных целей на этапе действия в *середине* процесса. Наименьшая активность имеет место между окончанием предыдущего и началом следующего процессного цикла, когда деятельность ограничена пассивным восприятием информации.

2. Фактор *силы* есть горизонтальная ось X, различающая процессные состояния по открытости и потенциалу, т.е. возможности влияния на последующее течение процесса как показано на Рис. 2. Наибольшее такое влияние имеет постановка субъективных целей в крайней левой точке траектории. Наименьшее влияние возможно на этапе получения финального результата в крайней правой точке. Таким образом фактор *силы* различает *начало* и *конец* процесса, близкие по уровню *активности*.

Из этих двух факторов обычно непосредственно наблюдаем лишь первый, определяющий явную динамику процесса (экономическая активность, интенсивность коммуникаций, возбуждение нейронного ансамбля и т.д.) [36], [39]. В этой связи процессную динамику иногда моделируют гармонической зависимостью такой величины от времени типа

$$y(t) = A\sin(\omega t + \phi_0), \tag{1}$$

что есть линейная проекция кругового движения на фазовой плоскости (рис. 1б) с радиусом A и частотой ω. Фаза колебания

$$\phi = \omega t + \phi_0 \pmod{2\pi}, \qquad 0 \le \phi < 2\pi \tag{2}$$

при этом соответствует фазе процесса как показано на Рис. 2.

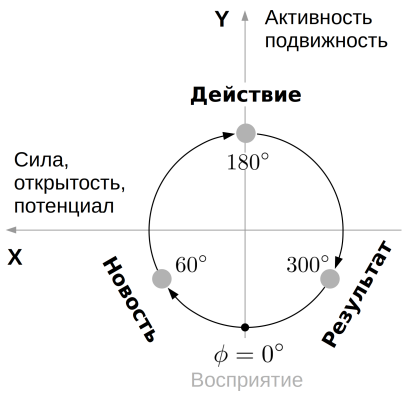


Рис. 2. Математическая формализация процесса на семантической (фазовой) плоскости, образованной факторами *сила* (X) и *автивность* (Y) классической семантики. Фаза процесса соответствует угловой координате ϕ.

Начало цикла в нижней точкетраектории по смыслу соответствует нулевой фазе $\phi = 0$, когда имеет место пассивное *восприятие* прошлых результатов. Следующие далее *новость*, *действие* и *результат* в простейшем симметричном варианте делят фазовую траекторию на три равных сектора. В программной реализации нулевую точку и направление обхода можно задать произвольно для удобства расчётов. Таким образом качественные модели жизненного цикла [24], [36] получают интерпретируемый математический аппарат, дополняющий существующие методы процессных вычислений и анализа данных [23], [40]–[43] .

3 Процессное квантование информации

Описанная структура процесса представляет собой шаблон, позволяющий структурировать информацию субъективно-целесообразным образом. Заполнение процессных состояний в этом шаблоне образует целостную информационную единицу, то есть *информационный квант*. Рассмотрим примеры такого квантования.

3.1 Языковая информация

Формально, текст есть линейная последовательность знаков — событий (букв, слов, предложений, разделов и глав), аналогичная показанной на Рис. 1(а). Практической ценностью, однако, обладает не любая последовательность такого типа, а лишь структурированная определённым образом. В случае научной статьи такая структура даётся классическим порядком разделов: описание проблемы, постановка задачи, методы решения, результаты и обсуждение [44]. Эта последовательность соответствует универсальным процессным состояниям как показано в Таблице 2.

В художественной литературе структура произведений не регламентирована. Тем не менее установлено, что в основе любой увлекательной истории – будь то сказка, миф, сюжет романа или киносценарий – лежит метафорическое путешествие героя, внутреннее или внешнее [45]. Типичная структура такого сюжета состоит из трёх частей: завязка-знакомство, испытание-действие и разрешение-развязка [46], функционально аналогичных трём универсальным процессным состояниям. Эти схемы, при необходимости подразделяемые на более мелкие этапы, показаны на Рис. 3.

Рис. 3. Процессная стуктура кванта текстовой информации. Сюжетная схема справа [45] приводится в соответствие с процессной моделью на Рис. 2 поворотом на 180 градусов.

То же самое верно для устных докладов и презентаций. В системах ведущих консалтинговых компаний те же три этапа названы ситуация-подход, возможности-трудности-находки и решения-выводы [47]. Во всех случаях эта стуктура задаёт строение процессного кванта языковой информации в соответствии с моделью, представленной в разделе 2. Подход к использованию этой схемы в задаче сценарного анализа текстов представлен в работе [48].

Таблица 2. Продолжение таблицы 1 в различных методах квантования информации.

Метод квантования информации	***Начало***	***Середина***	***Конец***
Научная публикация [44]	Введение и постановка задачи	Методы и решение	Результаты и обсуждение
Художественное произведение [46]	Завязка, контекст	Действие, борьба	Разрешение, финал
Образовательный курс [49]	Знания	Умения	Навыки

3.2 Знания — Умения — Навыки

В ходе обучения тройке процессных состояний соответствуют элементы дидактической цепочки знания — умения — навыки [49], [50] как показано в Таблице 2. Знания обеспечивают способность распознавания и объяснения явлений в изучаемой предметной области. Умения суть владение совокупностью приемов, обеспечивающих возможность выполнять целевую деятельность. Конечной точкой обучения являются навыки, обеспечивающие выполнение целевой деятельности в автоматическом режиме без сознательного контроля. В профессии водителя (лошади, автомобиля, самолёта, корабля), например, необходимо

- Знания: типичные дорожные или полётные ситуации, правила движения, устройство транспортного средства и основы соответствующих законов физики.
- Умения: приёмы управления и обслуживания транспортного средства, оказания первой медицинской помощи, починки лёгких неисправностей.
- Навыки: автоматизм распознавания дорожных ситуаций, автоматизм вождения.

Аналогичные элементы можно выделить в любой работе: анализ данных, разведение костра, приготовление еды, управление компанией и т.д. Во всех случаях для успеха необходимы все три элемента; в частности, владение только навыком (например, езды на велосипеде) не позволит перестроить деятельность в новых обстоятельствах в силу отсутствия соответствующих понятий на сознательном уровне психики. Таким образом определённые знания, умения и навыки образуют квант информационного обеспечения различных видов деятельности.

3.3 Сборка информационных блоков

Рассмотренная в предыдущем разделе процессная структура в языке есть проявление фундаментального свойства естественного мышления, организующего опыт, рассуждения и память в форме историй и повествований [51]–[54]. Рассмотрим пример такого структурирования новой информации на основе трёхэтапной модели процесса.

Пусть имеется событие «Компания А купила завод», типичное для сводки экономических новостей. Это событие можно интерпретировать как действие, отвечающее на некоторую новость-проблему. Такой новостью может быть, например, получение долгосрочного заказа, требующее расширения производственной мощности. Соответственно можно ожидать, что в результате этой сделки компания А сможет выполнить взятые на себя обязательства. Таким образом информационный блок «Компания А купила завод» размещён в минимальной причинно-следственной структуре, образующей единицу процессного мышления как показано на Рис. 4(а).

В этой модели, однако, из трёх информационных блоков достоверно известен лишь один – второй. Причина этого действия – блок 1 – является предположением, проверить которое часто не представляется возможным в силу, например, непубличности контракта. Следствие этого действия – блок 3 – также есть лишь более или менее вероятный прогноз, осуществление которого может зависеть от многих других факторов. Таким образом, построенная модель объекивно данного события носит субъективно-гипотетический характер.

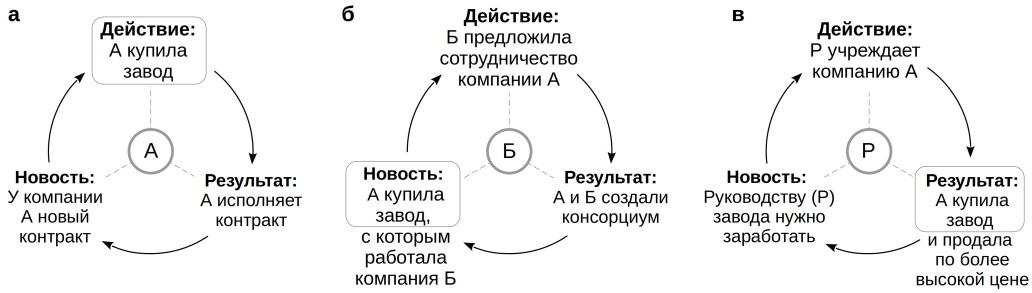


Рис. 4. Возможные процессные модели информационного блока - события «А построил дом». а: С точки зрения субъекта А. б: С точки зрения субъекта-семьи. в: С точки зрения субъекта Б.

Гипотетических моделей такого типа неограниченно много. Например, компания А могла купить завод для вложения средств или для подавления конкурентов. Кроме того, это событие может принимать и другие процессные состояния. Например, факт покупки завода компанией А может стать новостью для компании Б, которая также рассчитывала на этот завод. В ответ на это Б может отменить эти планы или предложить сотрудничество компании А. В последнем случае результатом может стать создание консорциума для ведения совместных проектов. В этой модели процесса, показанной на Рис. 4(б), субъектом является компания Б.

В другом случае та же самая покупка может быть итогом мошенничества, проводимого руководством завода с обычной целью собственного обогащения. Для решения этой проблемы может быть учреждена подставная компания А и разработан план приватизации. Рассматриваемая сделка тогда может быть проявлением этой схемы. В такой модели процесса, Рис. 4(в), субъектом является руководство компании (Р).

Как и предыдущие модели, эта гипотетическая история также составляет минимальное представление информационного блока в процессном мышлении. При этом полученный результат всегда отвечает на исходную проблему, обеспечивая функциональное замыкание процессного цикла (раздел 2.2). Эта структура эквивалентна рассмотренной выше структуре «Если C, сделай D, чтобы G», а также структурам *want — react — effect* [55] и *субъект — предикат — объект* в языке описания онтологий OWL [14], [56]. Эти методы частично реализуют процессную, причинно-следственную логику анализа данных, хотя и без функционального замыкания.

3.4 Субъективность процессного квантования

Процессное квантование информации, таким образом, характерно своей многовариантностью. Как и любое связное повествование, такие модели содержат не только «сырые» факты, но и их субъективную интерпретацию в виде логических связей с другими информационными блоками. В отличие от «объективной» информации, такие связи не содержатся в данных, а накладываются на них по произволу автора модели. Также произволен и выбор информационных блоков, включаемых в процессно-смысловые модели авторами альтернативных историй [57], [58].

Эта субъективность процессно-смыслового мышления [27], [59] представляет основное препятствие для его машинной имитации. Она обусловила, в частности, провал проекта W3C, предполагавшего возможность построения сети объективно истинных смысловых связей между информационными блоками. Реальное содержание интернета, напротив, отличается сосуществованием различных интерпретаций одних и тех же событий, естественно возникающих при взгляде на них с точек зрения различных субъектов. Эта смысловая относительность и многомерность, по всей видимости, не совместима с попытками построения всеобщей истины на основе «взгляда из ниоткуда» [60].

Субъективность процессного мышления, однако, имеет важнейшее прикладное значение в силу того, что реконструкции прошлых и прогнозы будущих событий, необходимые для принятия решений, обычно носят гипотетический характер. При этом на деле всегда подтверждается лишь малая часть от возможных процессных моделей; остальные оказываются ложными с более или менее тяжёлыми последствиями для их пользователей. Таким образом опыт позволяет совершенствовать качество процессно-смыслового мышления, приводя субъективизм индивидов в соответствие с объективными закономерностями природы.

4 Заключение

Как видно из примеров в разделе 3, процессное моделирование и структурирование информации на его основе не являются чем-то новым; напротив, этот способ представления данных широко используется в естественном повседневном мышлении. Его новизна в научном поле связана трудностью единообразной формализации достаточно широкого спектра процессов, допускающей их математическое описание. Эта задача подразумевает определение единицы процессного моделирования – «процессного кванта», организующего минимальный набор информационных блоков в целостную логическую структуру.

Представленная модель решает эту задачу. Размещение процесса в фазовой плоскости с интерпретируемыми координатами делает возможной формализацию процессной логики на основе элементарной алгебры угловых координат. В результате процессное структурирование информации становится доступным для алгоритмической имитации. Это в свою очередь открывает возможности для моделирования смысловой структуры психики, напрямую связанной с процессной логикой [17], [27], [59]. Сопряжение статически-классовой логики сущностей и процессно-смысловой логики действий позволит существенно повысить качество поисковых и прогнозно-аналитических систем.

Литература

[1] А. Н. Копайгородский, «Применение онтологий в семантических информационных системах», *Онтологии проектирования*, т. 4, вып. 14, сс. 78–89, 2014.

[2] Н. М. Боргест, «Роль онтологии в проектировании информационных систем», в *Open Semantic Technologies for Intelligent Systems*, 2014, т. 514.

[3] Н. М. Боргест и М. Д. Коровин, «Онтологии: Современное Состояние, Краткий Обзор», *Онтология проектирования*, вып. 2, сс. 49–55, 2013.

[4] A. V. Samsonovich, R. F. Goldin, и G. A. Ascoli, «Toward a semantic general theory of everything», *Complexity*, т. 16, вып. 4, с. NA-NA, 2009, doi: 10.1002/cplx.20293.

[5] A. Sheth, «Semantics Scales Up: Beyond Search in Web 3.0», *IEEE Internet Comput.*, т. 15, вып. 6, сс. 3–6, 2011, doi: 10.1109/MIC.2011.157.

[6] L. Feigenbaum, I. Herman, T. Hongsermeier, E. Neumann, и S. Stephens, «The Semantic Web in Action», *Sci. Am.*, т. 297, вып. 6, сс. 90–97, 2007, doi: 10.1038/scientificamerican1207-90.

[7] O. Brazhnik, «Databases and the geometry of knowledge», *Data Knowl. Eng.*, 2007, doi: 10.1016/j.datak.2006.05.005.

[8] I. A. Bessmertny, «Knowledge visualization based on semantic networks», *Program. Comput. Softw.*, т. 36, вып. 4, сс. 197–204, 2010, doi: 10.1134/S036176881004002X.

[9] А. А. Лебедев, А. С. Гаврилкина, Н. В. Максимов, О. Л. Голицына, и К. В. Монанков, «Онто-графовые механизмы глубинного семантического поиска», *Научно-Техническая Информация Серия 2. Информационные Процессы И Системы*, вып. 7, сс. 1–17, 2022, doi: 10.36535/0548-0027-2022-07-1.

[10] К. И. Костенко, «Диаграммы процессов и правила синтеза знаний из элементов онтологий», в *Знания-Онтологии-Теории*, 2021, сс. 131–140.

[11] А. Г. Додонов и В. Р. Сенченко, «Онтологический подход при формировании сценариев аналитики», в *Знания-Онтологии-Теории*, 2021, сс. 62–71.

[12] В. Е. Ложников и В. А. Маренко, «Семантическая сеть как информационная модель медиасферы», в *Знания-Онтологии-Теории*, 2019, сс. 227–232.

[13] И. Л. Артемьева, «Основанные на онтологиях инструментальные и прикладные интеллектуальные системы», в *Знания-Онтологии-Теории*, 2019, сс. 16–23.

[14] Д. А. Хворенков, «Веб-редактор процессных онтологий», *Интеллектуальные системы в производстве*, т. 18, вып. 3, сс. 72–76, 2020, doi: 10.22213/2410-9304-2020-3-72-76.

[15] В. И. Шалак, «Сравнительный логический анализ субстанциальной и процессуальной онтологий», *Логические исследования*, т. 26, вып. 2, сс. 58–86, 2020, doi: 10.21146/2074-1472-2020-26-2-58-86.

[16] В. И. Шалак, «Онтология и логика процессов», *Философия науки*, т. 62, вып. 6, сс. 138–150, 2019, doi: 10.30727/0235-1188-2019-62-6-138-150.

[17] А. В. Смирнов и В. К. Солондаев, *Процессуальная логика*. М: Садра, 2019.

[18] G. Bateson, «From classification to process», в *Mind in Nature. A necessary unity*, New York: E.P. Dutton, 1979, сс. 187–202.

[19] А. В. Брушлинский, *Мышление и прогнозирование*. М: Мысль, 1979.

[20] A. Bubic, D. Yves von Cramon, и R. I. Schubotz, «Prediction, cognition and the brain», *Front. Hum. Neurosci.*, т. 4, 2010, doi: 10.3389/fnhum.2010.00025.

[21] P. F. Greve, «The role of prediction in mental processing: A process approach», *New Ideas Psychol.*, т. 39, сс. 45–52, 2015, doi: 10.1016/j.newideapsych.2015.07.007.

[22] T. R. G. Green, «Pictures of programs and other processes, or how to do things with lines», *Behav. Inf. Technol.*, т. 1, вып. 1, сс. 3–36, 1982, doi: 10.1080/01449298208914433.

[23] D. Fahland, C. Ghidini, J. Becker, и M. Dumas, Ред., *Business Process Management*, т. 12168. Cham: Springer, 2020.

[24] Н. В. Максимов и А. А. Лебедев, «Онтологическая система „знания-деятельность“», *Онтология проектирования*, т. 11, вып. 2, сс. 185–211, 2021, doi: 10.18287/2223-9537-2021-11-2-185-211.

[25] Л. Р. Черняховская, Н. О. Никулина, А. И. Малахова, Ш. Г. Гарайшин, и Т. Р. Нагимов, «Проектирование системы управления бизнес-процессами на основе онтологического анализа и имитационного моделирования предметной области», *Информационные и математические технологии в науке и управлении*, т. 3, вып. 15, сс. 18–30, 2019, doi: 10.25729/2413-0133-2019-3-02.

[26] В. Н. Кучуганов, «Ассоциативная семантика ситуаций и сюжетов», *Искусственный интеллект и принятие решений*, вып. 2, сс. 42–51, 2014.

[27] И. А. Суров, «Жизненный цикл: смысловая матрица процессного моделирования», *Онтология проектирования*, т. 12, вып. 4, сс. 430–453, 2022, doi: 10.18287/2223-9537-2022-12-4-430-453.

[28] М. В. Белов и Д. А. Новиков, «Структура методологии комплексной деятельности», *Онтология проектирования*, т. 7, вып. 4, сс. 366–387, 2017, doi: 10.18287/2223-9537-2017-7-4-366-387.

[29] М. В. Белов, «Проблемы управления жизненными циклами организационно-технических систем», *Управление в социально-экономических системах*, сс. 117–172, 2018.

[30] M. Siegel, «The sense-think-act paradigm revisited», 2003, doi: 10.1109/ROSE.2003.1218700.

[31] E. Gat, «On Three-Layer Architectures», *Artif. Intell. Mob. Robot.*, сс. 195–210, 1997, doi: 10.1.1.165.5283.

[32] J. Doblin, «A Short, Grandiose Theory of Design», *STA Des. Journal, Anal. Intuit.*, т. 1, сс. 6–16, 1987.

[33] R. D. Moen и C. L. Norman, «Circling Back: Clearing up myths about the Deming cycle and seeing how it keeps evolving», *Qual. Prog.*, вып. 11, сс. 22–28, 2010.

[34] В. И. Шалак, «Логика в онтологии процессов», *Логические исследования*, т. 27, вып. 2, сс. 48–65, 2021, doi: 10.21146/2074-1472-2021-27-2-48-65.

[35] N. Rescher, *Dialectics: a classical approach to inquiry*. De Gruyter, 2007.

[36] Д. Б. Берг, Е. А. Ульянова, и П. В. Добряк, *Модели жизненного цикла: учебное пособие*. Екатеринбург: Издательство Уральского университета, 2014.

[37] «ГОСТ 57193-2016. Процессы жизненного цикла систем». Стандартинформ, М, 2016.

[38] C. E. Osgood, «Studies on the generality of affective meaning systems», *Am. Psychol.*, т. 17, вып. 1, сс. 10–28, 1962, doi: 10.1037/h0045146.

[39] Ю. Е. Вагин и К. В. Судаков, «Колебательная модель системоквантов поведения», *Известия РАН. Теория и системы управления*, т. 6, сс. 166–176, 2008.

[40] G. Grambow, R. Oberhauser, и M. Reichert, Ред., *Advances in Intelligent Process-Aware Information Systems. Concepts, Methods, and Technologies*, т. 123. Cham: Springer, 2017.

[41] W. Van der Aalst, *Process mining: Data science in action*, 2-е изд. Heidelberg: Springer, 2016.

[42] М. Н. Бухаров, «Информационная система для онтологического моделирования предметных областей», *Информатика*, сс. 85–99, 2022, doi: 10.37661/1816-0301-2022-19-2-85-99.

[43] В. В. Грибова и Е. А. Шалфеева, «Онтология диагностики процессов», *Онтология проектирования*, т. 4, вып. 34, сс. 449–461, 2019, doi: 10.18287/2223-9537-2019-9-4-449-461.

[44] L. B. Sollaci и M. G. Pereira, «The introduction, methods, results, and discussion (IMRAD) structure: A fifty-year survey», *J. Med. Libr. Assoc.*, т. 92, вып. 3, сс. 364–367, 2004.

[45] К. Воглер, *Путешествие писателя. Мифологические структуры в литературе и кино (пер. с англ.)*. М: Альпина нон-фикшн, 2015.

[46] S. Field, *Screenplay: The Foundations of Screenwriting*. New York: Delta, 2005.

[47] D. McKinsey, *Strategic storytelling: how to create persuasive business presentations*. 2014.

[48] I. A. Surov, «Process-Semantic Analysis of Words and Texts», в *Artificial Intelligence in Models, Methods and Applications*, O. Dolinina, I. Bessmertny, A. Brovko, V. Kreinovich, V. Pechenkin, A. Lvov, и V. Zhmud, Ред. Cham: Springer, 2023, сс. 247–260.

[49] А. Р. Камалеева, «Технологическое наполнение дидактической цепочки „Знания → Умения и навыки“ с позиции когнитивной дидактики», *Педагогический журнал Башкортостана*, т. 95, вып. 1, сс. 41–52, 2022, doi: 10.21510/18173292_2022_95_41_52.

[50] Н. А. Селивёрстова, «Знание, умения и навыки как элементы образовательного опыта молодежи», *Научные Труды Московского Гуманитарного Университета*, вып. 4, сс. 28–34, 2015, doi: 10.17805/trudy.2015.4.3.

[51] J. Bruner, «The narrative construction of reality», *Crit. Inq.*, т. 18, 1991.

[52] T. Akimoto, «Cogmic space for narrative-based world representation», *Cogn. Syst. Res.*, т. 65, сс. 167–183, 2021, doi: 10.1016/j.cogsys.2020.10.005.

[53] N. Szilas, «Towards narrative-based knowledge representation in cognitive systems», в *6th Workshop on Computational Models of Narrative (CMN 2015)*, 2015, сс. 133–141.

[54] C. León, «Narrative as a fundamental information unit in the mind», *AAAI Fall Symp. - Tech. Rep.*, т. FS-17-01, сс. 408–413, 2017.

[55] D. Fedyukin, D. Gromozdov, и Y. Gapanyuk, «Creating a Metagraph Dataset Based on the ATOMIC Dataset», в *Artificial Intelligence Methods and Applications*, O. Dolinina, I. Bessmertny, A. Brovko, V. Kreinovich, V. Pechenkin, A. Lvov, и V. Zhmud, Ред. Springer, 2023, сс. 207–218.

[56] «RDF 1.1 Primer», *W3C Working Group Note 24*, 2014. http://www.w3.org/TR/rdf11-primer.

[57] А. К. Гуц, «Вариации истории и модальная логика», *Математические структуры и моделирование*, т. 4, вып. 64, сс. 122–128, 2022, doi: 10.24147/2222-8772.2022.4.122-128.

[58] A. Meroño-Peñuela *и др.*, «Semantic technologies for historical research: A survey», *Semant. Web*, т. 6, вып. 6, сс. 539–564, окт. 2014, doi: 10.3233/SW-140158.

[59] И. А. Суров, «Какая разница? Прагматическая формализация смысла», *Искусственный интеллект и принятие решений*, вып. 1, сс. 78–89, 2023, doi: 10.14357/20718594230108.

[60] T. Nagel, *The View From Nowhere*. Oxford University Press, 1986.

Когнитивные Возможности и Модели Коллективных Субъектов

В.А. Филимонов[1]

[1]*Институт математики СО РАН, г. Новосибирск, пр. Академика Коптюга, 4, 630090, Россия.*

filimonov-v-a@yandex.ru

Аннотация. *Областью исследования является деятельность субъектов и коллективов субъектов. Ставится задача исследования специфики коллективной деятельности. Рассматриваются минимальные логические структуры, соответствующие субъектам и коллективам субъектов. В рамках кросс-технологий ситуационного центра используется авторский подход «4К» (коллективный, когнитивный, конфигураторный, конвергентный), включающий модели К.Г. Юнга и В.А. Лефевра, а также философию информатики А.А. Берса. Предлагаются новые инструменты: когнитивная осведомлённость, абстрактный конечный результат, метафорические когнитивные карты, рефлексивный театр с функциональными местами Комментатора и АнтиКомментатора. Приводятся результаты экспериментов с коллективами, в том числе включающими системы искусственного интеллекта. Результатом являются два варианта постановки задачи исследования коллективных субъектов.*

Ключевые слова: коллективные субъекты, подход «*4К*», метафоры, эксперименты, обучение

1 Введение

Исследование коллективных субъектов встречается достаточно редко. Однако рассмотрение коллектива субъектов, а также субъекта как коллектива квази-субъектов, позволяет получать нетривиальные результаты. Стало актуальным рассмотрение коллективов систем искусственного интеллекта (ИИ), таких как многоагентные системы, рои беспилотников, а также смешанные команды из людей и систем ИИ. Интеллектуальное поведение демонстрируют даже системы, которые не являются субъектами в обычном понимании этого термина как носитель деятельности, сознания и познания. Примером является модельный организм *Physarum polycephalum,* который фактически является колонией (коллективом) клеток. При этом он обладает памятью, способностью к оптимизации траекторий, и даже демонстрирует подверженность «*эффекту асимметричного превосходства*» (decoy effect) [1].

Как известно, свойства системы не всегда могут быть выведены из свойств элементов, которые её образуют. Свойства отдельных субъектов исследованы гораздо лучше, чем свойства коллективов. Отдельный интерес представляют свойства коллективных субъектов, аналогичные свойствам субъектов-индивидов.

В ходе исследований, связанных с образовательными процессами, появлялись новые инструменты, которые здесь впервые описываются в комплексе. Эти инструменты интегрированы в подход «4К» [2, 3], созданный на основе кросс-технологий ситуационного центра, ранее представленных автором на данной конференции [4]. Среди этих инструментов важное место занимают когнитивные компоненты, связанные с художественным творчеством, такие, как рефлексивный театр ситуационного центра.

Взрывное проникновение обученных нейросетей в качестве подручных средств для индивидуального изготовления текстов и графики также требует рассмотрения перспектив их развития.

2 Теоретические компоненты

2.1 Замечание о постановках задач

Областью исследования является деятельность субъектов и коллективов субъектов. Назначение статьи - рассмотрение двух возможных постановок задач. В первом случае коллективные субъекты (КС) рассматриваются аналогично субъектам, входящим в их состав. Примером первого направления является процесс коллективной экспертизы, где в качестве критерия выступает точность оценки. Второе направление фокусируется на свойствах КС, которые отсутствуют у составляющих их субъектов-подсистем. Примером может являться система из людей и роботов, где сенсорные свойства людей дополняются техническими возможностями роботов (например, оценкой радиоактивности), а люди выполняют функции коммуникации, связанные с эмпатией, при контактах с одушевлёнными системами.

Добавим, что построение определений связано с постановкой и решением конкретной задачи. Ниже рассматривается фундаментальная схема А.А. Зиновьева, в рамках которой могут осуществляться возможные постановки задач.

2.2 Фундаментальная схема «Субъект – Логика – Онтология»

Как правило, в научных публикациях не упоминаются субъекты, которые будут интерпретировать тексты эти публикаций. Корректное описание научного исследования, согласно [5], должно включать в себя триаду «*Субъект – Логика - Онтология*», где *Логика* - это конструктор, с помощью которого *Субъект* (Исследователь, Наблюдатель и т.п.) формирует *Онтологию* - представление исследуемого объекта (процесса) в терминах своей *Логики*.

Пояснить это для технической сферы можно следующим примером. Для трёхмерной детали (объекта) вариантом её *Онтологии* будет программа изготовления этой детали с помощью 3D принтера, написанная на языке программирования с использованием G-кода (который является компонентом *Логики*, как и прошивка принтера RepRap или Marlin), а *Субъектом* будет компьютер, который встроен в принтер, и интерпретирует команды/коды. Для гуманитарной сферы примером может являться работа психоаналитика (*Субъекта*), который интерпретирует процесс диалога с клиентом как *Онтологию* навязчивых состояний клиента с помощью теории Фрейда (*Логики*).

Одним из следствий бессубъектности описания моделей, помимо проблем с их интерпретацией, является отсутствие рефлексивного управления исследуемыми одушевлёнными объектами, хотя теория В.А. Лефевра [6-9] позволяет получать нетривиальные результаты.

Можно заметить, что отдельного индивида можно рассматривать как коллектив взаимодействующих суб-субъектов. Примером могут служить взаимодействие систем мышления 1 и 2 в модели Д. Канемана [10], психиатрические расстройства и искажения типа «реальность как ошибка» [11]. В [12] предлагается определение как целостного организма улья (*точнее – «роя» В.Ф.*) , в котором отдельные пчёлы выполняют функцию клеток организма.

В данном тексте мы будем считать, что Субъектом является система, обладающая следующими свойствами:

- одушевлённостью в смысле В.А. Лефевра [6], т.е. способностью различать негативный и позитивный полюса в своей картине мира, и осуществлять выбор полюса (варианта реакции);
- альгедоничностью (способностью ассоциировать полюса с удовольствием и страданием), и/или её имитацией [13];
- способностью оперировать с интеллигибельными истинами (например, преподаваемыми в технических вузах).

Данному определению будет соответствовать как человек, так и достаточно обученная нейросеть типа GP4. Варианты дополнительного рассмотрения одушевлённости рассмотрены в [14].

2.3 Подход «4К»

В данный подход входят четыре компонента: коллективный, когнитивный, конфигураторный и конвергентный [2, 3]. Компонент «коллективный» реализует выполнение следующих операций:

- Создание профилей (психологических, компетентностных и т.п.) индивидов, входящих в коллектив.
- Формирование и оптимизация характеристик коллектива на основе профилей индивидов и отношений между ними.

Среди множества моделей взаимодействия индивидов в группах особое место принадлежит рефлексивному анализу в трактовке В.А. Лефевра [7,9].

Компонент «когнитивный» предполагает максимально возможное включение в деятельность всех сенсорных возможностей человека, в том числе его подсознания. В первую очередь это предпочтения человека относительно каналов работы с информацией: ведущее полушарие мозга (левое или правое) и репрезентативная система (видео, аудио, кинестетическая, тактильная).

Компонент «конфигураторный» является инструментом для сборки системы из готовых компонентов в соответствии с правилами системного анализа, где конфигуратором называется комплекс языков и представлений объектов с позиций различных дисциплин. В трактовке В.А. Лефевра конфигуратор требует наличия модели субъекта сборки. Вопросы конфигурации детально рассмотрены в публикациях А.А. Берса [15, 16].

Компонент «конвергентный» обеспечивает сходимость процесса групповой работы к оптимальному в определённом смысле результату, такому как инженерное решение, бизнес-проект, вердикт присяжных и т.п. В процессах обучения оптимальным для субъектов может быть отрицательный результат в виде провала коллективного проекта. В этом случае необходим рефлексивный анализ проекта («разбор полётов»).

2.4 Когнитивная осведомлённость и абстрактный конечный результат

В военном деле используется термин «*ситуационная осведомлённость*» (*situational awareness*), который подразумевает знание обо всех обстоятельствах, критических для определённой деятельности в конкретной ситуации. Предлагается термин «*когнитивная осведомлённость*» (*cognitive awareness*), под которым понимается рассмотрение ситуации в более широком контексте, в частности, с позиций антихрупкости [17]. Когнитивная осведомлённость предполагает использование в комплексе системного и рефлексивного анализа, а также эвристики.

Одним из фундаментальных достижений известной Теории Решения Инженерных Задач (ТРИЗ) является понятие Идеального Конечного Результата (ИКР), когда система выполняет заданные действия без затраты ресурсов, в том числе пространства. Однако методологически идеализации, т.е. устремлению определённого свойства к пределу, должна предшествовать абстракция: выделение этого обобщённого объекта и свойства (качества). В ТРИЗ используются предельные абстракции: время, пространство, вещество, поле, противоречие. При конструировании ИКР обычно не осознаётся и не фиксируется построение абстракций. Предлагается дополнительно к ИКР формировать абстрактный конечный результат (АКР). Например, для ИКР «*билеты проверяются сами без контролёра*» АКР будет сформулирован так: «*объекты контролирования контролируются без системы контролирования*». «Клеточку» общей схемы можно представить как «*Объект1 / Процесс / Объект2*» либо «*Процесс1 / Объект / Процесс2*». Во многих случаях абстрагирование выполняется за счёт обобщения (яблоки / фрукты / плоды /объекты). Понятию «*идеальный маятник*» предшествует абстракция «*маятник*».

3 Компоненты технологий обучения

Описанные ниже компоненты могут быть использованы для конструирования синоптических (целостных, обзорных) учебных курсов, дополняющих существующие курсы образовательных учреждений. Подчеркнём, что такие курсы реализуются в подходе «4К» на основе ресурсов, доступных в конкретной ситуации.

3.1 Метафорические когнитивные карты

В когнитивных науках много результатов по исследованию структуры и функционирования мозга, поведения человека в различных условиях, однако весьма редко изучаются когнитивные аспекты, связанные с искусством, а также с подсознательным восприятием информации. Автором предложены «метафорические когнитивные карты» [18], позволяющие дополнить информацию о рассматриваемом предмете (теме) ассоциативными связями. На Рис. 1 представлен вариант метафорической карты для когнитивной сферы. Все картины, входящие в состав этой карты, снабжены комментариями относительно смысла конкретной картины в данном тексте (Таблица 1).

Такие карты в учебном процессе можно формировать методом мозгового штурма. Эксперименты со студентами показали, что такая работа вызывает интерес и улучшает понимание ситуации в целом.

Рис.1. Метафорическая когнитивная карта когнитивистики

Таблица 1. Интерпретация компонентов метафорической когнитивной карты

№	Условное название	Предлагаемая когнитивная интерпретация
1	Атмосфера	Новое знание основано на старом
2	Вавилонская башня	Пример коллективного проекта
3	Операция глупости	Рефлексия: незнание и невежество
4	Римские числа	Важность базиса системы
5	Наскальные рисунки	Схема организации коллективной работы
6	Пенсионеры	Пример когнитивной графики
7	Станцуй диссертацию	Синтез наук и искусства
8	Остров Лапута	Технопарк
9	Чемодан на колёсах	Когнитивная осведомлённость
10	Невозможная фигура	Конфигуратор. Противоречие образа и логики
11	Спасатель	Потенциал деятельности и ограничения
12	Руки человека и робота	Содружество интеллекта

3.2 Модели из людей и рефлексивный театр

Для объяснения студентам финансового университета того, как работают нейронные сети, был использован следующий процесс [19].

Этап 1. Студенты в компьютерном классе командами по два человека делали рисунки с помощью нейросетевых сервисов Autodraw.com и Quickdraw. В первом сервисе можно нарисовать изображение, а нейросеть его завершает так, как она его распознала. Во втором сервисе идёт игра и обучение нейросети. Программа пытается за 15 секунд угадать, что именно нарисовано. На этом этапе у студентов формируется представление о ключевых компонентах системы (в данном случае, изображения). В частности, студентам почти всегда удавался зонт, и не удавался бульдозер. В дальнейшем это помогло объяснить понятие «основы *чего-либо*», например, управления.

Этап 2. С помощью логического автомата решалась задача определения, является ли объект на определённой фотографии человеком. Использовалось классическое определение: «*Человек – это двуногое без перьев*». Логический автомат был сформирован из трёх студентов и работал

следующим образом. Два студента быди сенсорами, первый из которых отвечал за наличие на фотографии двуногого существа, а второй – за наличие у существа перьев. В случае ответа «ДА» сенсор поднимал руку с открытой ладонью, а в противном случае руку со сжатым кулаком. Решение принимал студент, который выполнял функцию логического элемента «И».

3. Эту же задачу решала нейросеть, в которой был дополнительный слой (три сумматора=студента, суммирующие сигналы от сенсоров с весовыми коэффициентами); в этом случае сигнал «ДА» от сенсоров интерпретировался как +1, а «НЕТ» как 0. Решение принимал студент=элемент «Если»: в случае превышения заданного порога суммарным значением сигнала от сумматоров он принимал решение «ДА» и «НЕТ» в противном случае. Обучение нейросети состояло именно в подборе весовых коэффициентов, в которых признак двуногости был более значим, чем признак отсутствия перьев. Для ускорения эксперимента соответствующие весовые коэффициенты были вычислены заранее, напечатаны на карточках и вручены сумматорам, которые выбирали нужную карточку в зависимости от реакции сенсоров. Элемент «Если» суммировал значения трёх поднятых карточек.

Для эксперимента были приготовлены 4 фотографии: Гарри Поттер, индейский вождь в головном уборе с перьями, курица и робот-андроид. Теоретически логический автомат должен был признать человеком Гарри Поттера и робота, а нейросеть ещё и индейского вождя. Ошибка по признанию робота человеком позволила детальнее рассмотреть используемое определение. Непредвиденным результатом был абсолютно оправданный отказ сенсора считать Гарри Поттера двуногим, поскольку на использованном фото изображение его ног отсутствовало.

Студенты проявили достаточный энтузиазм в реализации данных простых моделей ИИ, что является необходимым условием освоения учебного материала.

Для инклюзивных вариантов обучения в рефлексивном театре ситуационного центра предложен способ конструирования т.н. «драматических представлений» по системе Я.А. Коменского [20], дополнительно использующий метод Ильи Франка. Метод заключается в дроблении текста (или видео) на небольшие фрагменты, которые демонстрируются дважды, причём в первый раз с рефлексивным комментарием. Ещё один вариант заключается в оппозиционном комментировании, которое осуществляют Комментатор и Анти-комментатор.

Специфика учебного процесс в Финансовом университете при Правительстве РФ (Омский филиал) обусловила рассмотрение влияния систем ИИ на финансовую сферу. Соответствующее рассмотрение [21, 22] выявило тенденцию к увеличению финансового взаимодействия между системами ИИ.

4 Заключение

Приведённая постановка задачи исследования коллективных субъектов позволяет актуализировать соответствующие исследования с учётом развития систем искусственного интеллекта. Отмечены два направления исследований. В первом направлении коллективные субъекты рассматриваются аналогично субъектам, входящим в их состав. Второе направление фокусируется на эмерджентных свойствах коллективных субъектов, которые отсутствуют у составляющих их субъектов-подсистем.

Ссылки на модели В.А. Лефевра и К.Г. Юнга с наибольшим исследовательским потенциалом дополнены описаниями авторских инструментов, в числе которых когнитивная осведомлённость, абстрактный конечный результат, метафорические когнитивные карты, а также рефлексивный театр с функциональными местами оппозиционных комментаторов.

5 Благодарности

Работа выполнена в рамках государственного задания Института математики СО РАН, проект FWNF-2022-0016.

Литература

[1] Элленберг Дж. Как не ошибаться. Сила математического мышления. М.: МИФ, 2018. — 576 с.

[2] Filimonov V.A., Burmistrova N.A., Chernyavskaya V.S., Malakhova V.R. Collective Development of Cognitive Abilities Using the "4C" Approach // *Proceedings - 2021 IEEE Ural-Siberian Conf. on Computational Technologies in Cognitive Science, Genomics and Biomedicine (CSGB).* – PP. 60–63. DOI: 10.1109/CSGB53040.2021.9496021

[3] Филимонов В.А. Пандемия и спецоперация: многодисциплинарный анализ с использованием подхода «4К» // *Математические структуры и моделирование*.– 1(61), 2022.– С. 82-91. EDN VQJRBY.
[4] Филимонов В. А. Сума технологии: три пятилетки многодисциплинарного проекта // *Знания - Онтологии - Теории (ЗОНТ-2015) : Материалы Всероссийской конференции с международным участием, Новосибирск, 06–08 октября 2015 года* / Том 2. – Новосибирск: ООО "Технотрейд", 2015. – С. 180-188. – EDN UOBRVB.
[5] Зиновьев А.А. Логический интеллект – М.: Изд-во Московского гуманитарного ун-та, 2005. – 284 с.
[6] Лефевр В.А. Что такое одушевлённость? - М: Когито-Центр, 2017.–122 с.
[7] Лефевр В.А. Алгебра совести. – М.: Прогресс, 2003. – 408 с.
[8] Лефевр В.А. Конфликтующие структуры.–М.:Советское радио», 1973.– 158 с.
[9] Лефевр В.А. Лекции по теории рефлексивных игр.– М.: Когито-центр, 2009.– 218 с.
[10] Канеман Д. Думай медленно, решай быстро.– М.: АСТ, 2020.- 653 с.
[11] Руднев В. П. Реальность как ошибка. М.: Гнозис, 2011. – 320 с.
[12] Watson L. The Romeo Error: A Matter of Life and Death.- Hodder and Stoughton, 1974, P. 254.
[13] Филимонов В. А. Концепция альгедонической нейронной сети на основе гомеостатики и рефлексивного управления // *Математическое и компьютерное моделирование : Сборник материалов V Международной научной конференции, Омск, 01 декабря 2017 года*. – Омск: Омский государственный университет им. Ф.М. Достоевского, 2017. – С. 79-81. – EDN YOKOVI.
[14] Филимонов В.А., Чернявская В.С. Формализация одушевлённости на примере понятия «любовь» // Онтология проектирования. – 2022. – Т. 12, №1(43). - С. 11-24. – DOI: 10.18287/2223-9537-2022-12-1-11-24.
[15] Берс А.А. Коммуникация, понимание, мышление – онтологические представления (стенограмма приглашённого доклада) // *Рефлексивный театр ситуационного центра-2011 / Материалы 5-ой Всероссийской конференции с международным участием РТСЦ-2011*. Омск : Омский государственный институт сервиса, 2012, С. 6-20.
[16] Берс А.А. Философия информатики // *Рефлексивный театр ситуационного центра-2012 / Материалы 6-ой Всероссийской конференции с международным участием РТСЦ-2012.* Омск : Омский государственный институт сервиса, 2013, С. 9-23.
[17] Талеб Н. Антихрупкость. Как извлечь выгоду из хаоса.– М.: КоЛибри, Азбука-Аттикус, 2020.– 768 с/
[18] Филимонов В.А. Проект краткого обзорного курса «Избранные вопросы инженерии». Онтология проектирования. 2022. Т.12, №4(46). С.494-504. DOI:10.18287/2223-9537-2022-12-5-494-504.
[19] Бурмистрова Н.А., Филимонов В.А., Шамис В.А. Включённое обучение методам искусственного интеллекта: два эксперимента со студентами // *Робототехника и искусственный интеллект: материалы XIV Всероссийской научно-технической конференции с международным участием (г. Железногорск, 26 ноября 2022 г.). –* Красноярск: ЛИТЕРА-принт, 2022, С.198 -203. EDN JHZYSW
[20] Машевская С.М. Школа как игра: вариант использования идей театра Я. А. Коменского в современном образовании // *в кн.: Ян Амос Коменский и Дмитрий Чижевский: Диалог идей.*–Siedlice, 2015, с. 217-229.
[21] Filimonov V.A. Money and robots: the two shadows of humanity that will destroy it // Mathematical Structures and Modeling. 2021. № 4 (60). С. 109-114.
[22] Филимонов В.А. Искусственный интеллект и финансовая система: анализ взаимодействия // Робототехника и искусственный интеллект: материалы XIII Всеросс. науч.-техн. конф. с междун. участием (г. Железногорск, 27 ноября 2021 г.) / Красноярск: ЛИТЕРА-принт, 2021. – С. 272-275. – EDN BJUVKS.

Уменьшение влияния проблемы холодного старта при формировании туристических рекомендаций

А.К. Хорошавин[1]

[1]*Новосибирский Государственный Университет, ул. Пирогова 1, Новосибирск, 630090, Россия.*
[2]*Novosibirsk State University, Pirogova str. 1, Novosibirsk, 630090, Russia*

a.khoroshavin@g.nsu.ru

Аннотация. *В статье описывается алгоритм составления туристических рекомендаций, который может сводить к минимуму влияние известной проблемы рекомендательных систем, такой как «холодный стар». Алгоритм осуществляет сбор данных об оценках и перемещениях пользователя, а затем использует собранную информацию, чтобы найти потенциальные места, которые могут быть интересны конкретному пользователю.*

Ключевые слова: холодный страт, история местоположений, рекомендательная система, GPS, ближайшие места, анализ местоположений пользователей, контент-ориентированная рекомендация.

1 Введение

Туризм является одной из наиболее быстрорастущих отраслей экономики, и количество людей, путешествующих по миру, продолжает расти. Тем не менее, туристические маршруты, описанные в буклетах и гидах, не всегда оптимизированы для индивидуальных пользователей [1]. В этом случае полезны оказываются различные сервисы и приложения, которые стараются учитывать предпочтения потребителей [2], используя рекомендательные системы.

Обычно такие системы отслеживают переходы пользователей между тем или иным контентом, однако такой подход может быть малоэффективен в случае новых пользователей, о предпочтениях которых ничего не известно. Проблема «холодного старта» является распространенной для многих рекомендательных систем [3], в том числе в области туризма. Не зная предпочтений пользователя, мы не сможем предложить из множества мест те, которые могут быть ему интересны. Из-за этого пользователю приходится просматривать десятки мест, чтобы найти то, что зацепит его взгляд, это значительно увеличивает временные затраты на организацию досуга.

На сегодняшний день уже существуют подходы, позволяющие снизить влияние проблемы холодного старта. Например, сервисы, такие как GuideMe, используют интеграцию с социальными сетями пользователей [4]. Обычно такая страница отображает некоторые особенности пользователя, что позволит снизить влияние проблемы холодного старта. Однако не всегда пользователи ведут свои страницы в социальных сетях и не дают нам никакой информации. В других системах используются методы с применением рассуждения на основе прецедентов и искусственный интеллект [5, 6]. Сначала система предлагает пользователю места исходя из общих соображений, затем на основе его ответов (выбрал рекомендацию/отказался от рекомендации) идет процесс машинного обучения, который позволит сформировать лучший тур для данного пользователя.

Помимо этого, возможно использование гибридных рекомендательных систем, когда объединяются алгоритмы контентной и коллаборативной фильтрации, а также дополнительно используется демографический подход. В этом случае учитывается социально-экономическая

информация, такая как занятость, уровень образованности, средний доход, уровень рождаемости и смертности. Получить такую статистику можно без участия пользователя, а значит ее можно использовать чтобы составить рекомендации мест в регионе, где находится сам пользователь.

В данной статье мы рассматриваем подход с использованием индивидуальной истории местоположений. Для данного подхода учитывается удаленность интересных мест от текущего местоположения пользователя, а также удаленность от индивидуальных маршрутов, которые пользователь совершал в течение определенного времени.

2. Индивидуальная история местоположений

Растущая доступность технологий определения местоположения GPS позволяет регистрировать индивидуальную историю местоположений в виде пространственно-временных данных [7]. Такие истории в некоторой степени позволяют нам получить сведения об интересах пользователей к конкретным местам на карте. Анализ цифрового следа конкретного пользователя на карте местности используется в качестве неявных оценок в районе\городе. Зная, где пользователь находился, мы можем предложить ему те или иные близко расположенные места из списка непосещенных. В результате мы формируем профиль предпочтений, зная лишь местоположение, решая проблему «холодного старта» для новых пользователей [8].

Данная модификация контент-ориентированной рекомендательной системы была разработана в рамках проекта «*WhereToGo.NSK*». Для этого была использована следующая структура в базе данных для хранения индивидуальной истории местоположений пользователей (Рисунок 1).

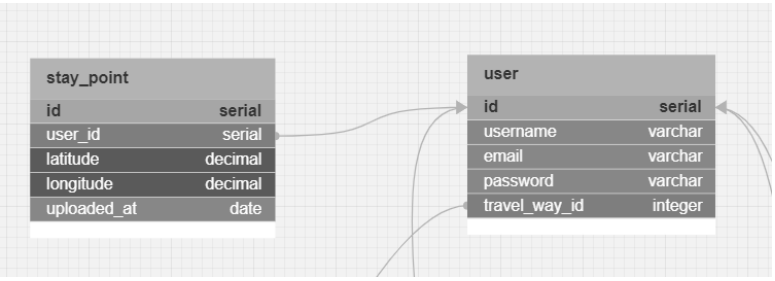


Рис.1. Сущность геоточек в индивидуальной истории перемещений пользователя.

Структура геоточек имеет следующие атрибуты:

- Широта
- Долгота
- Идентификатор пользователя
- Дата регистрации

Выбор таких атрибутов обусловлен несколькими причинами. Широта и долгота – стандартный формат данных, если речь идет о геолокации. Он также удобен при поиске мест в реляционных базах данных.

Дата регистрации нужна по двум причинам. Индивидуальные маршруты пользователя могут меняться с течением времени, поэтому нам важно, чтобы эти данные периодически обновлялись. Также было решено выделять каждому пользователю фиксированное число геоточек, чтобы избежать переполнения базы данных, а также избежать ситуации, когда история местоположений покрывает всю карту, что не позволит давать актуальные рекомендации и будет нагружать систему.

Таким образом, если число геоточек пользователя достигнет предельного количества, будет удалена самая старая геоточка, тем самым мы обеспечим обновление индивидуальных маршрутов пользователей, что позволит давать самые актуальные рекомендации на основе их перемещений.

Геоточки пользователя будут регистрироваться периодически, и может возникнуть ситуация, когда несколько раз будет записана точка с одними и теми же координатами. Чтобы избежать переполнения базы данных, было принято решение проверять схожесть значений широты и долготы и, если точки совпадают, обновлять время регистрации геоточки.

3. Рекомендации на основе геолокации

Для работы алгоритма рекомендаций на основе перемещений необходимо определить, как мы будем собирать информацию для индивидуальной истории местоположений. В ходе разработки было решено реализовать на стороне клиента алгоритм, который определяет, какие точки на карте необходимо вносить в базу данных.

Были определены два критерия:

1. Сколько времени пользователь находится в данной точке местности;
2. Насколько далеко пользователь переместился от геоточки-кандидата для истории перемещений;

Было решено использовать временной промежуток, равный 15 минутам. Раз в 15 минут алгоритм отправляет запрос на получение информации о местоположении пользователя. Затем проверяется второй критерий.

Если пользователь не отошел от точки, полученной на предыдущей итерации дальше, чем 200 метров, то можно сделать вывод о том, что точка с предыдущего шага является точкой остановки пользователя. Поэтому мы рассматриваем ее как геоточка индивидуальной истории местоположений и отправляем ее данные на сервер. Спустя несколько итераций мы получим набор непересекающихся областей карты, которые будет использовать алгоритм рекомендаций.

Идея алгоритма состоит в том, чтобы проходить по каждой из геоточек и находить фиксированное число ближайших к ним мест в базе данных.

4. Поиск ближайших геоточек в реляционной базе данных

Чтобы предложить пользователю места, используя его историю местоположений, нужно иметь возможность находить ближайшие геоточки в базе данных нашего сервиса. Делать это на стороне клиентской части с помощью Google Maps API не рационально, так как это будет сильно нагружать устройство пользователя.

Допустим, нам необходимо найти 5 ближайших мест к точке, заданной своими координатами. Однако база данных может быть очень большой, что может стать причиной долгого поиска подходящих мест кандидатов. Если мы будем проходить по всей базе данных и сравнивать у каждого места широту и долготу, то в худшем случае нам придется обойти всю базу данных, чтобы найти подходящие места.

Одним из возможных решений данной проблемы является рассмотрение мест, близких в обоих направлениях к целевой широте и долготе. При этом, предполагается рассматривать эти два направления по очереди, выполняя так называемое «разделение» по широте, а затем по долготе. Такой подход позволяет значительно уменьшить число обращений к диску базы данных по сравнению с простым обходом [9].

На вход алгоритму подается широта/долгота целевой точки, а также минимальный и максимальный радиус поиска. Следуя принципу «разделения», мы последовательно формируем «квадраты» на карте, в которых ищем потенциальные места координаты.

Сначала выполняется отбор по широте (горизонтальные стороны квадрата), затем для полученного множества мест выполняется подзапрос, который отбирает нужные места по долготе (вертикальные стороны квадрата). Если в текущем квадрате мы не смогли найти достаточное количество нужных мест, то мы выполняем расширение области поиска.

Итерации продолжаются до тех пор, пока не будет найдено нужное количество мест, либо пока у нас не закончится допустимая область поиска (Рисунок 2).

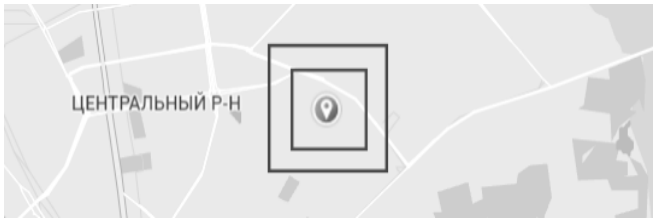


Рис.2. Формирование «квадратов» вокруг точки на карте для поиска ближайших мест.

Алгоритм 1 ниже описывает работу основной процедуры поиска мест в реляционной базе данных.

Алгоритм 1: *Процедура поиска ближайших геоточек.*

Вход: *Данные геоточки, описанные ее широтой и долготой. Критерии остановки работы алгоритма (минимальное число найденных мест* ***k****).*

Выход: *Набор геоточек, которые расположены вблизи данной точки.*

1: ***Пока*** *число найденных мест меньше* ***k****:*

2: *Формируем ограничивающую область по широте* ***lat****;*

3: *Отбираем места из базы данных, которые* ***входят в полученный диапазон широты****;*

4: *Формируем ограничивающую область по долготе* ***lon****;*

5: *Отбираем места из базы данных, которые* ***входят в полученный диапазон долготы****;*

6: ***Если*** *число найденных мест меньше* ***k****:*

7: *Повторяем шаги 2-5, расширяя область поиска;*

8: ***Если*** *при достижении максимального расширения области, число найденных мест меньше* ***k****:*

9: *Выводим список мест, которые смогли найти, либо пустой список.*

10: ***Иначе****:*

11: *Выводим список найденных мест.*

12: ***Завершение алгоритма****.*

Тем не менее, алгоритм зависим от таких параметров как минимальная и максимальная области поиска. Чем больше значение этих параметров, тем больше времени требуется алгоритму на поиск кандидатов в базе данных мест. Поэтому в рамках проекта было решено использовать минимальное расстояние, равное 1 км, а максимальное – 2 км. Также это способствует сведению к минимуму число повторений мест-кандидатов, оказавшихся на пересечении областей поиска вокруг геоточек.

Однако, возникает вопрос: что делать если нет даже истории местоположений? Такая ситуация может возникнуть, если пользователь совсем недавно появился в системе и у него нет истории перемещений, или если он не захотел предоставлять доступ к GPS своего мобильного устройства.

Для первого случая было решено добавить возможность поиска ближайших мест к текущей позиции пользователя. Этот метод работает при включенном GPS.

Для второго случая была введена система «посещенных» мест, которые пользователь может вносить с помощью интерфейса клиента. В этом случае нам не нужно от пользователя доступа к GPS, достаточно иметь набор мест, посещенных пользователем. Эти места будут рассматриваться аналогично геоточкам описанной ранее индивидуальной истории перемещений.

5. Сбор и обработка данных местоположений

Данный модуль был реализован на стороне клиента и представляет собой сервис системы Android, который работает в фоновом режиме. В ходе разработки пришлось учесть некоторые ограничения:

1. Современные операционные системы Android имеют встроенные алгоритмы оптимизации расхода ресурсов системы, если памяти устройства не хватает на выполнение тех или иных процессов, система может принудительно остановить одну из фоновых задач, которой может быть наш сервис анализа местоположений.
2. Для сервиса сбора геоданных FusedLocationProviderClient можно настроить частоту обращений к GPS устройства. Дополнительно происходит обращение к сети Интернет, чтобы отправить геоточку на сервер. Эти операции могут значительно снижать уровень заряда батареи при слишком частых вызовах.

В связи с перечисленными ограничениями было решено применить технологию ForegroundService. Данный сервис позволяет повысить приоритет процесса, что позволит избежать его отключения системой. Особенностью данного сервиса является уведомление на панели уведомлений. Таким образом, пользователь будет в курсе, что делает приложение, а также сможет отключить сбор данных в любой момент. Это делает наше приложение открытым для пользователя и не создает негативного представления о том, что мы пытаемся втайне от пользователя брать его данные (Рисунок 3).

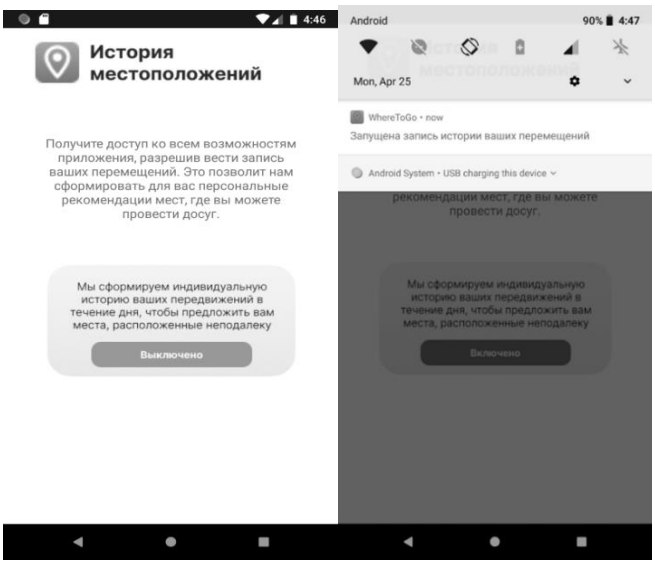


Рис.3. Окно с возможностью включения/отключения истории местоположений.

Для частоты обращения к данным был выбран интервал, равный 15 минутам, это позволит экономнее расходовать ресурсы устройства, а самое главное – заряд батареи.

В качестве примера, сформируем индивидуальную историю перемещений пользователя следующим образом. (Рисунок 4)

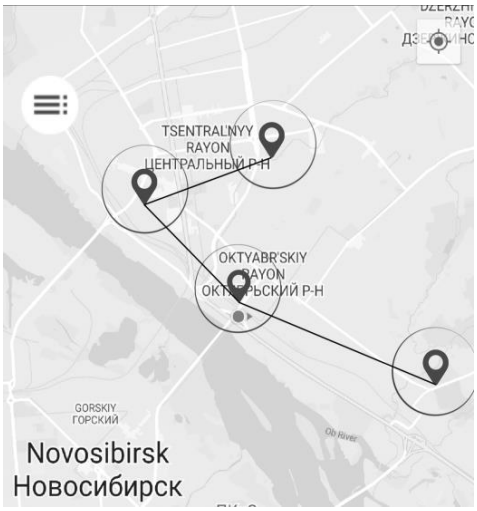


Рис.4. Схема индивидуальной истории местоположений.

Как только пользователь отправляет запрос на получение рекомендаций, сервер ищет в базе данных набор геоточек и использует их в качестве входных данных для алгоритма рекомендаций [10].

В результате происходит отбор таких мест, которые входят в области поиска геоточек индивидуальной истории местоположений (Рисунок 5).

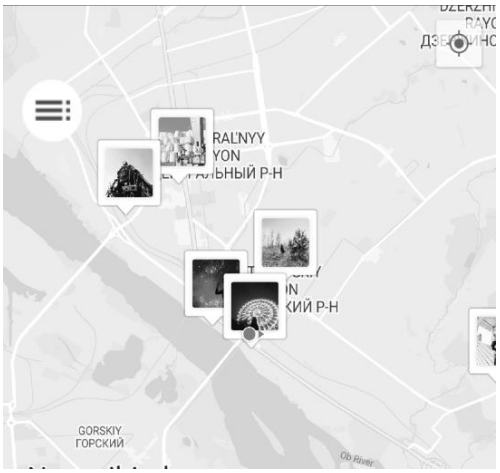


Рис.5. Карта с рекомендациями на основе истории местоположений.

Таким образом, мы получаем персональные рекомендации в соответствии с индивидуальной историей местоположений пользователя. Аналогичным образом алгоритм работает в случае, когда нет истории местоположений: рассматривается только одна точка – текущая геолокация пользователя.

За счет приведенного метода оценки поведения пользователей мы получили возможность предоставлять качественные рекомендации в случае, когда мы ничего не знаем о новом пользователе.

6. Заключение

Разработанный алгоритм был реализован в рамках проекта «Wheretogo.NSK». Интеллектуальный гид Wheretogo.NSK, в котором на сегодняшний день собрано около 200 мест Новосибирской области, облегчит его пользователям процесс поиска мест для посещения и отдыха. Все локации для отдыха и посещения собирались местными экспертами в области туризма. В дальнейшем интеллектуальный гид может быть масштабирован и на другие города России, такие как Томск и Москва.

Литература

[1] Коцюба И.Ю., Назаренко А.Е. Разработка рекомендательной системы для планирования туристических маршрутов в оптимизационной постановке // Моделирование, оптимизация и информационные технологии. 2020;8(2).

[2] Yakhyaeva G., Skokova V. Subjective Expert Evaluations in the Model-Theoretic Representation of Object Domain Knowledge. // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2021, 12948 LNAI, стр. 152–165

[3] Р. З. Омаров, А. В. Востротина, А. Д. Ли [и др.] Проблема «холодного старта» // Молодой ученый. — 2019. — № 26 (264). — С. 85-88.

[4] Artem Umanets, Artur Ferreira, Nuno Leite, GuideMe – A Tourist Guide with a Recommender System and Social Interaction // Procedia Technology, Volume 17, 2014, Pages 407–414

[5] Yakhyaeva, G., Yasinskaya, O. Application of case-based methodology for early diagnosis of computer attacks. Journal of Computing and Information Technology, 2014, 22(3), стр. 145–150

[6] Palchunov, D.E., Yakhyaeva G.E. Integration of Fuzzy Model Theory and FCA for Big Data Mining \\ SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings, 2019, стр. 961–966

[7] Yu Zheng, Lizhu Zhang, Zhengxin Ma, Xing Xie and Wei-Ying Ma, Recommending Friends and Locations Based on Individual Location History // ACM Transactions on the Web, Vol.5, No. 1, 2011, pp. 1–44.

[8] Хорошавин А.К. Использование индивидуальной истории местоположений для контент-ориентированной рекомендации мест. Знания-Онтологии-Теории (ЗОНТ-2021). 2021. С. 329.

[9] Rick James, MySQL Latitude // Longitude Indexing [Электронный ресурс] – Режим доступа: http://mysql.rjweb.org/doc.php/latlng

[10] Хорошавин А.К. Анализ поведения пользователей с использованием данных истории перемещений при составлении персональных рекомендаций мест // Материалы 60-ой Международной научной студенческой конференции МНСК – 2022: Информационные технологии. Программная инженерия и инженерия знаний. Новосибирск, НГУ 2022 – С.167.

Разработка интеллектуального помощника для автоматизированного порождения документов кафедры

Чернявцева Светлана Игоревна

Новосибирский государственный университет, Пирогова 1, г. Новосибирск, Россия.

s.chernyavtseva@g.nsu.ru

Аннотация. *Работа посвящена разработке формальных математических методов моделирования бизнес-процессов и применению этих методов для создания интеллектуальных помощников. Для формализации документооборота на кафедре университета применяется теория частичных моделей. Рассмотрена задача создания интеллектуального помощника для автоматизированного порождения документов кафедры. Для решения этой задачи была разработана онтологическая модель предметной области и построены шаблоны документов. Благодаря автоматизированному заполнению документов повышается эффективность и качество изготовленных документов, а также уменьшается время на работу с ними. Возможность гибкого изменения разработанной онтологической модели позволяет оптимизировать работу с документами согласно меняющимся регламентам.*

Ключевые слова: интеллектуальный помощник, автоматизация документооборота, частичная модель, гомоморфизмы частичных моделей, онтология предметной области, онтологическая модель, четырехуровневая модель представления знаний

1 Введение

В настоящее время актуальность автоматизации бизнес-процессов трудно переоценить [1]. Так, автоматизация создания и обработки документов на кафедре в значительной степени упрощает работу секретаря, способствует уменьшению числа ошибок, позволяет ему более эффективно выполнять задачи, в отличие от чисто ручного труда, который не всегда гарантирует положительный результат. Ручное создание документов может повлечь за собой неправильное их заполнение, поскольку разные люди принимают участие в изготовлении документов, вследствие чего возникают ошибки. Например, могут быть неверно указаны сведения о должностях, степенях руководителей, месте прохождения практики, неправильно сформулировано название выпускной квалификационной работы. Некорректно заполненные документы будут отклонены секретарем государственной аттестационной комиссии, вследствие чего придется повторно вносить изменения. Поэтому секретарю кафедры приходится вручную проверять большой объем документов, чтобы избежать ошибок. Таким образом, возникает необходимость решать задачу порождения заведомо правильных документов, включающая соответствие регламентам, правильность написания имен сотрудников, должностей.

В работе моделируется роль секретаря кафедры. Целью этого моделирования является разработка интеллектуального помощника секретаря кафедры. Существующие программные решения (системы электронного документооборота, автоматизированное рабочее место секретаря кафедры, интеллектуальные помощники для работы с текстом) имеют ряд недостатков в рамках задач, решения которых представлены в данной статье:

1. Отсутствует возможность гарантированного порождения заведомо правильных документов, в составлении которых принимают участие люди.

2. Отсутствуют средства нахождения и разрешения противоречий как в имеющихся данных, так и между документами.
3. Отсутствует возможность автоматической подстройки бизнес-процессов под изменения в регламентах.
В рамках разрабатываемого подхода решаются отмеченные выше задачи путем создания интеллектуального помощника секретаря кафедры, функционал которого позволяет:

- порождать предзаполненные шаблоны, которые должны быть заполнены;
- рассылать участникам бизнес-процесса предзаполненные документы;
- проводить проверку документов на наличие опечаток, противоречий.

Для решения проблемы порождения заведомо правильных документов нужно разделить данные, которые должны быть внесены в шаблон, на два вида:

- то, что однозначно определяется информацией, содержащейся в базе знаний (ФИО студента, группа, тема работы, должность научного руководителя и рецензента и пр.);
- то, что содержит мнение и оценку научного руководителя, рук. практики, рецензента.

Работа осуществляется с тремя видами документов: шаблоны, предзаполненные и заполненные документы. Шаблоны и предзаполненные документы описываются частичными моделями, потому что в них содержится не вся информация, а лишь частичная. В то время как документ содержит полную информацию. При изготовлении предзаполненного документа в шаблон вносится вся та информация, которая является детерминированной. Интеллектуальный помощник будет порождать для каждого студента пакет предзаполненных документов, которые далее окончательно заполняют научные руководители, руководители практики, рецензенты, студенты. Следует отметить, что именно в этой части не возникает проблемы с правильностью или несоответствием регламентам, поскольку оценки человек пишет исходя из своего мнения, что по определению является верным. Помимо этого, необходимо осуществлять проверку уже полученных документов на наличие изменений формальных требований в процессе их доработки, чтобы участники процесса не внесли данные, которые являются неверными. Таким образом, решается проблема порождения заведомо формально правильных документов. Для реализации этого подхода необходимо решать задачи:
1) создание и изменение шаблонов документов;
2) наполнение базы данных, актуализация, проверка на соответствие реальности.

2 Разработка интеллектуального помощника

2.1 Онтологическая модель

Для разработки модели предметной области в работе используются онтологии [2-4]. Применение онтологий дает большие возможности и преимущества при моделировании различных предметных областей [5—7]. Онтологией называют общее описание основных понятий предметной области и то, как они связаны между собой. Для того, чтобы корректно описать предметную область используется онтологическая модель, которая содержит информацию об объектах, событиях, процессах данной области знаний. Вся информация определена в терминах, заданных онтологией [8]. Используется четырехуровневая модель представления знаний при создании онтологической модели [9]:

1) онтология предметной области;
2) общие знания;
3) прецеденты;
4) вероятностные знания.

В рамках задач, описываемых в работе, используются первые три уровня.
Определение. Формальной онтологией предметной области SD называется пара O=<A, σ>, где σ — множество ключевых понятий предметной области, и A — множество аналитических предложений, описывающих смысл данных ключевых понятий [10]. Онтология включает такие понятия как: руководитель, учащийся, практика, профессор, ассистент и другие (рис. 1 и 2). Общие знания — это утверждения, которые должны выполняться для всех прецедентов. При помощи первых двух порождается третий — уровень прецедентов, то есть заполненные документы и автоматически предзаполненные шаблоны. Для описания модели предметной области используется язык OWL и редактор онтологий Protégé.

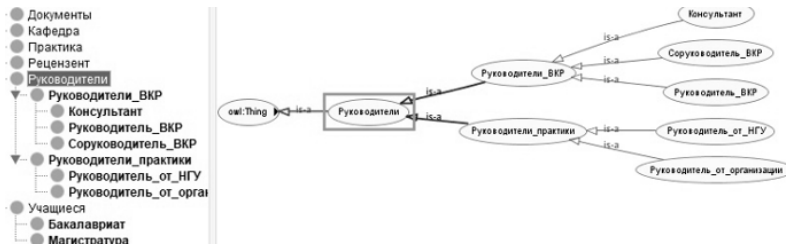


Рис. 1. Пример класса «Руководители»

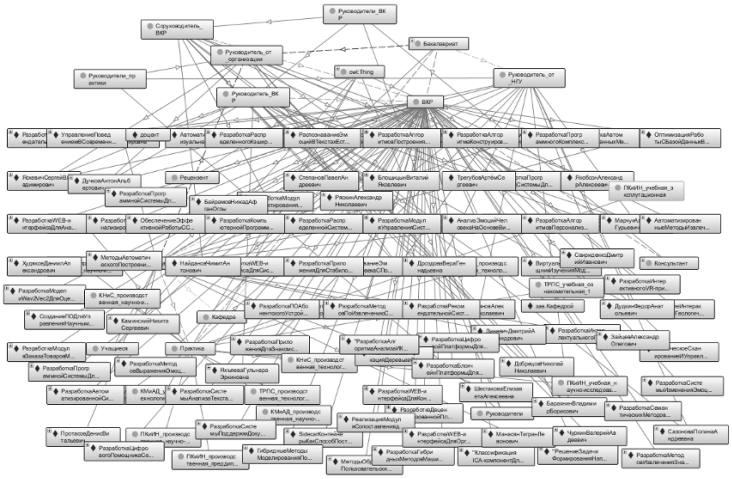


Рис. 2. Фрагмент онтологии предметной области

2.2 Конвертация данных из Excel таблиц в OWL

Для извлечения информации из таблиц об участниках бизнес-процесса были написаны правила, (рис. 3), которые позволяют добавлять новые знания в онтологическую модель.

	Sheet Name	Start Column	End Column	Start Row	End Row	Rule
✓	4 курс б.	C	C	2	+	Individual: @C*(["(\S+ \S+ \S+),"]) Types: Руководитель_ВКР Facts: согласовывает @F*, ФИО @C*(["(\S+ \S+ \S+),"])
✓	4 курс б.	F	F	2	+	Individual: @F* Types: ВКР Facts: Тема @F*
✓	4 курс б.	A	A	2	+	Individual: @A* Types: Бакалавриат Facts: ФИО @A*, защищает @F*
✓	4 курс б.	D	D	2	+	Individual: @D*(["(\S+ \S+ \S+),"]) Types: Соруководитель_ВКР Facts: ФИО @D*(["(\S+ \S+ \S+),"])
✓	4 курс б.	E	E	2	+	Individual: @E*(["(\S+ \S+ \S+),"]) Types: Консультант Facts: консультирует @F*, ФИО @E*(["(\S+ \S+ \S+),"])

Рис. 3. Правила для извлечения данных из Excel и добавление их в разрабатываемую модель

3 Функциональная часть

На основании определенного описания бизнес-процессов кафедры, в том числе сроков, когда рассылаются документы для прохождения практики / защиты ВКР, необходимо выполнить задачи: a) создавать шаблоны документов; b) заполнять шаблоны документов; c) заполнить и проверить источники информации; d) организовать рассылку писем.
Точкой входа программы является модуль для работы с интерфейсом. Далее происходит взаимодействие с онтологической моделью, а после с заполнение документов. В настоящее время программная система имеет три основных модуля (взаимодействие с онтологической моделью, заполнение документов и взаимодействие пользователя с интерфейсом).

3.1 Модуль для взаимодействия с онтологической моделью

Взаимодействие с базой знаний осуществляется с помощью библиотеки Jena SPARQL API [11]. Для работы с запросами используется язык SPARQL [12]. В зависимости от профиля обучения студента выбирается конкретный шаблон, так как для каждого направления указаны

определенные критерии оценивания компетенций. Запрос, позволяющий извлекать необходимую информацию для заполнения документов, представлен на рис. 5.

```
SELECT * WHERE {
?студент a my:$degree .
?студент my:защищает ?вкр .
?студент my:ФИО ?фио_студента .
?студент my:группа ?группа_студента .
?студент my:Место_прохождения_практики ?место_практики .
?студент my:Приказ_на_прохождение_практики ?приказ_практика .
?студент my:Профиль_обучения ?профиль .
FILTER regex(?профиль, "$profile")


?студент my:на_НГУ_практике_у ?руководитель_от_НГУ .
?руководитель_от_НГУ my:ФИО ?фио_НГУ_руководителя .
?руководитель_от_НГУ my:Должность_в_НГУ ?должность_НГУ_руководителя .

?студент my:на_орг_практике_у ?руководитель_от_организации .
?руководитель_от_организации my:ФИО ?фио_орг_руководителя .
?руководитель_от_организации my:Должность_в_организации ?должность_орг_руководителя

?руководитель my:согласовывает ?вкр .
?руководитель my:ФИО ?фио_руководителя .
?руководитель my:Должность ?должность_руководителя .

?вкр my:Тема ?тема_вкр .

?рецензент my:рецензирует ?вкр .
?рецензент my:ФИО ?фио_рецензента .
```

Рис. 5. SPARQL запрос для извлечения знаний из построенной онтологии

Для того, чтобы между ключевыми словами и данными в модели была связь, указывается наименование свойства в OWL (рис. 6). Знания преобразуются в HashMap, которая будет передана в модуль для заполнения документов.

```
DOC_FIELD_TO_SOLUTION = new HashMap<>();
DOC_FIELD_TO_SOLUTION.put("имяСтудентаР", fullName( solutionVarName: "фио_студента", Case.Genitive));
DOC_FIELD_TO_SOLUTION.put("группаСтудента", simple( solutionVarName: "группа_студента"));
DOC_FIELD_TO_SOLUTION.put("темаВКР", simple( solutionVarName: "тема_вкр"));
DOC_FIELD_TO_SOLUTION.put("местоПрактики", simple( solutionVarName: "место_практики"));
DOC_FIELD_TO_SOLUTION.put("полноеНаименованиеМестаПрактики", simple( solutionVarName: "место_практики_полное_наименование"));
DOC_FIELD_TO_SOLUTION.put("имяРуководителяОтНГУ", simple( solutionVarName: "фио_НГУ_руководителя"));
DOC_FIELD_TO_SOLUTION.put("должностьВНГУ", simple( solutionVarName: "должность_НГУ_руководителя"));
DOC_FIELD_TO_SOLUTION.put("имяРуководителяОтОрганизации", simple( solutionVarName: "фио_орг_руководителя"));
DOC_FIELD_TO_SOLUTION.put("должностьВОрганизации", simple( solutionVarName: "должность_орг_руководителя"));
DOC_FIELD_TO_SOLUTION.put("имяРуководителяВКР", simple( solutionVarName: "фио_руководителя"));
DOC_FIELD_TO_SOLUTION.put("должностьРуководителяВКР", simple( solutionVarName: "должность_руководителя_вкр"));
DOC_FIELD_TO_SOLUTION.put("должностьРуководителяКратковКР", simple( solutionVarName: "должность_руководителя_вкр_кратко"));
DOC_FIELD_TO_SOLUTION.put("обучСтудОбрПадеж", gender( solutionVarName: "фио_студента"));
```

Рис. 6. Фрагмент кода подстановки данных из модели на месте ключевых слов

3.2 Модуль для заполнения документов

Для автоматического заполнения документов секретарь должен подставить на месте пропусков ключевые слова в соответствие с таблицей 1.

Таблица 1. Ключевое слово и его значение.

Ключевое слово	***За какую информацию отвечает***
имяСтудентаИ	ФИО в именительном падеже
группаСтудента	Номер группы студента
темаВКР	Тема задания ВКР
местоПрактики	Место прохождения практики
имяРуководителяОтНГУ	ФИО руководителя практики от НГУ
имяРуководителяВКР	ФИО руководителя ВКР
имяРуководителяОтНГУ	ФИО руководителя практики от НГУ
фиоРецензента	ФИО рецензента
должностьВНГУ	Должн. рук. практики от НГУ
имяРуководителяОтОрганизации	ФИО руководителя от проф. орг.
должностьВОрганизации	Должн. рук. практики от проф. орг.
профильОбучения	Профиль обучения студента
приказПрактика	Акт о направлении на практику
должностьРецензента	Должность рецензента
фиоСоруководителяВКР	ФИО соруководителя ВКР
УчСтепРукВКР	Ученая степень руководителя ВКР
должностьСоруководителяВКР	Должность соруководителя ВКР

Для разработки программной системы выбран язык Java, поскольку он содержит библиотеку, предназначенную для работы с документами формата — Apache POI [13]. На вход модулю для заполнения документов подаются: HashMap, путь до шаблона и путь для конечного документа. Правильное заполнение документов включает в себя не только извлечение актуальной и верной информации из онтологической модели, но и подстановку правильной формы слова. Для склонения фамилии, имени и отчества была выбрана библиотека petrovich4j [14].

3.3 Модуль для рассылки писем

Функциональная часть интеллектуального помощника секретаря кафедры, отвечающая за рассылку писем, не только включает в себя отправку предзаполненных документов, но и имеет возможность уведомить секретаря кафедры и студентов о необходимости предоставить документы на прохождение практики/защиты ВКР. Имеется 3 вида сообщений с разной степенью важности: уведомление; напоминание; предупреждение.
Модуль для рассылки писем на данный момент находится в стадии разработки.

3.4 Модуль для взаимодействия с программным интерфейсом

Интерфейс построен с использованием языка Java и библиотеки Swing. Секретарю кафедры доступны 3 опции: заполнение, просмотр и удаление. Порождение документов осуществляется нажатием на кнопку «Заполнить документы». Можно сгенерировать определенный документ или полностью пакет документов, шаблоны которых загружены в систему (рис. 8). При нажатии на кнопки «Посмотреть документы», «Удалить документы» нужно выбрать ступень образования (бакалавриат, магистратура), курс, вид документа.

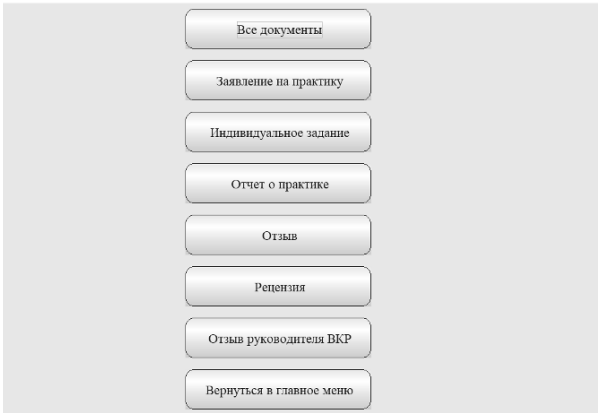


Рис. 8. Выбор документа для заполнения

4 Заключение

В работе предложен подход к созданию интеллектуальных помощников, направленных на автоматизацию документооборота на кафедре университета. Данный подход основан на использовании онтологий и теории частичных моделей. Он позволяет гибко модифицировать документооборот при изменении регламентов. Для создания и заполнения иерархических шаблонов используются частичные модели, где одна часть информации уже известна, другая — еще нет. Построена онтологическая модель кафедры. На основе онтологической модели реализуется автоматическое заполнение шаблонов документов. Разработана часть функционала интеллектуального помощника секретаря кафедры, позволяющая создавать шаблоны, заполнять документы и взаимодействовать с секретарем кафедры.

Литература

[1] Свистунов В. М., Лобачев В. В. Актуальные тренды автоматизации бизнес-процессов в отечественных компаниях // Управление персоналом и интеллектуальными ресурсами в России. 2022. Том 11, № 2. С. 72—76. DOI: 10.12737/2305-7807- 2022-11-2-72-76.

[2] Palchunov D., Yakhyaeva G., Yasinskaya O. Software system for the diagnosis of the spine diseases using case-based reasoning // Proceedings of the International Conference on Biomedical Engineering and Computational Technologies (SIBIRCON / SibMedInfo). 2015. P. 205—210.

[3] Капустина А. И., Пальчунов Д. Е. Разработка онтологической модели тарифов и услуг сотовой связи, основанной на логически полных определениях понятий // Вестник НГУ. Серия: Информационные технологии. 2017. Том 15, № 2. С. 34—46.
[4] Олейник А. О. Разработка интеллектуального помощника секретаря кафедры // Материалы 60-й Международной научной студенческой конференции. Информационные технологии. Программная инженерия и инженерия знаний. 2022. С. 159.
[5] Flanagan K., Stevens R., Pocock M., Lee P., Wipat A. Ontology for genome comparison and genomic rearrangements // Conference Papers, Comparative and Functional Genomics. 2004. Vol. 5, No. 6—7. P. 537—544, DOI: 10.1002/cfg.436.
[6] Клещев А.С., Москаленко Ф.М., Черняховская М.Ю.Модель онтологии предметной области «Медицинская Программная инженерия. Том 14, № 8, 2023 399 диагностика». Ч.1: Неформальное описание и определение базовых терминов//НТИ. Серия 2.2 005. №12. С.1—7.
[7] Найданов Ч. А. Разработка ядра онтологической модели, настраиваемой под предметную область // Вестник НГУ. Серия: Информационные технологии. 2019. Том 17, № 1. С. 72—81. DOI: 10.25205/1818-7900-2019-17-1-72-81.
[8] Пальчунов Д. Е. Моделирование мышления и формализация рефлексии. I: Теоретико-модельная формализация онтологии и рефлексии // Философия науки. 2006. № 4 (31). С. 62—99.
[9] Naydanov Ch., Palchunov D., Sazonova P. Development of automated methods for the prevention of risks of critical conditions, based on the analysis of the knowledge extracted from the medical histories // Сибирский научный медицинский журнал. 2016. Том 36, № 1. С. 105—113.
[10] Пальчунов Д. Е. Решение задачи поиска информации на основе онтологий // Бизнес-информатика. 2008. № 1. С. 3—13.
[11] Jena SPARQL API. URL: https://github.com/ SmartDataAnalytics/jena-sparql-api.
[12] SPARQL. URL: https://www.w3.org/TR/rdf-sparql-query/.
[13] Apache POI. URL: https://poi.apache.org.
[14] Petrovich. URL: https://github.com/ damirazo/Petrovich?ysclid=lgubb4xenj46782821.

Разработка блокчейн платформы для осуществления сделок с динамическим дисконтированием

Черпаков Кирилл Максимович

Новосибирский Государственный Университет,
ул. Пирогова, д. 2, г. Новосибирск, 630090, Россия.

cherpakov-kirill-42@mail.ru

Аннотация. *Данная работа посвящена разработке веб-приложения для осуществления сделок с динамическим дисконтированием с использованием технологии блокчейн. В работе рассматриваются методы применения смарт-контрактов блокчейн-сети Ethereum при заключении сделок с динамическим дисконтом.*

Ключевые слова: разработка веб-приложения, технология блокчейн, смарт-контракты, динамическое дисконтирование.

1 Введение

В современном мире развивается тренд на цифровизацию документооборота. Цифровизации подвергаются документы в бухгалтерии, заявления сотрудников, различные записи на приёмы к специалистам, в том числе и документооборот при заключении сделок между контрагентами.

Документы при заключении сделок можно вносить в централизованный или децентрализованный реестр. Использование децентрализованных систем баз данных с каждым годом становится всё более привычным и обыденным для крупных компаний. Одним из видов децентрализованных систем является технология блокчейн.

Технология блокчейн — децентрализованная система баз данных, реализующая хранение данных, записанных в специальные блоки, и предоставляющая возможность запуска смарт-контрактов в пределах блокчейн-системы.

Сравнительно недавно в мире заключения сделок появился финансовый инструмент — динамическое дисконтирование, позволяющий получать выгоду при заключении сделок как заказчику, так и поставщику.

Динамическое дисконтирование — финансовый инструмент, используемый в управлении оборотным капиталом компании, который заключается в согласовании стоимости услуги или товара в зависимости от сроков оплаты счетов между поставщиком и покупателем.

Инициатор сделки формирует правила расчёта величины дисконта в соответствие со сроком, на который будет осуществлена ранняя оплата счёта. Второй агент сделки соглашается на данные условия сделки или же отказывается, предлагая свои условия. Заключается договор ранней оплаты счетов, устанавливаются условия поставки или предоставления услуг взамен на скидку от поставщика.

Оба контрагента имеют выгоду. Заказчик получает скидку на заказ: чем раньше он оплатит счёт, тем большая будет выгода. Поставщик в свою очередь имеет выгоду за счет сокращения затрат на привлечение финансирования.

Для осуществления записи данных об условиях и правилах заключаемых сделок имеет смысл использовать современную технологию блокчейн.

1.1 Актуальность работы

Актуальность темы доклада обусловлена необходимостью проведения безопасных и быстрых в подготовке сделок между двумя контрагентами: соглашений с динамическим дисконтированием.

Компаниям-поставщикам необходимо заключать сделки быстрее, чтобы привлекать дополнительные финансы, а для этого нужно упростить механизм заключения соглашений и привнести дополнительную выгоду для заказчиков с помощью динамического дисконтирования.

В наше время динамическое дисконтирование — это наиболее выгодный способ пополнения оборотного капитала для крупных компаний. Привлечение капитала особенно актуально для производственных предприятий. Чем больше будет капитала, там больше продукции сможет произвести предприятие.

Таким образом, разработка программной блокчейн-системы для проведения сделок с динамическим дисконтированием, безусловно, является актуальной задачей.

1.2 Цель работы

Целью данной научно-исследовательской работы является разработка программного обеспечения для осуществления процессов динамического дисконтирования, используя современную технологию блокчейн.

1.3 Постановка задач

Для достижения цели были установлены следующие задачи:

1) проанализировать принципы работы технологии блокчейн, финансового инструмента динамическое дисконтирование, смарт-контрактов платформы Ethereum;
2) сделать обзор аналогов разрабатываемой системы;
3) сформировать список технических требований к блокчейн-платформе для осуществления сделок с динамическим дисконтированием;
4) спроектировать архитектуру программного комплекса;
5) реализовать программный комплекс;
6) произвести тестирование разработанного проекта.

Объектом исследования является современная технология блокчейн.

1.4 Новизна работы

Новизна разработанного проекта состоит в том, что в ходе выполнения работы был предложен и реализован новый подход к составлению условий сделок с динамическим дисконтированием с использованием смарт-контрактов и хранении данных о сделках в децентрализованной блокчейн системе.

Условия сделки и информация о контрагентах хранятся в смарт-контрактах, которые имеют программные методы взаимодействия для изменения состояния смарт-контракта посредством транзакций. Смарт-контракт является программным классом с набором полей и методов для изменения состояния смарт-контракта. В разработанной платформе методы смарт-контрактов используются для проведения оплаты, изменения статуса сделки, проверки выполнения условий соглашения.

Смарт-контракт для осуществления сделки с динамическим дисконтом принимает входные параметры: сроки сделки, предмет сделки, процент динамического дисконта, информация о контрагентах. Для обновления состояния смарт-контракта имеются методы взаимодействия:

1) проверить статус сделки;
2) получить условия динамического дисконта;

3) обновить условия заключения сделки;

4) обновить стоимость товара или услуги.

1.5 Практическая ценность

Практической ценностью разработанной системы является возможность найти контрагента и заключить сделку онлайн на условиях динамического дисконта с использованием децентрализованной базы данных. Условия, установленные контрагентами, будут записаны в смарт-контракт и отправлены в распределённый реестр сразу после заключения сделки, что повышает уровень защищённости данных. Разрабатываемая платформа будет полезна организациям, предприятиям, производствам для привлечения оборотного капитала. Пользователи, заинтересованные в упрощении заключения сделок с динамическим дисконтированием, смогут пользоваться данной платформой.

2 Анализ предметной области

2.1 Динамическое дисконтирование

Динамическое дисконтирование — финансовый инструмент, используемый в управлении оборотным капиталом компании, принципом которого является согласование стоимости услуги или товара в зависимости от сроков оплаты счетов между поставщиком и покупателем [14].

В наше время динамическое дисконтирование — это наиболее выгодный способ пополнения оборотного капитала для крупных компаний.

Инициатор сделки формирует правила расчёта величины дисконта в соответствие со сроком, на который будет осуществлена ранняя оплата счёта. Второй агент сделки соглашается на данные условия сделки или же отказывается, предлагая свои условия. Заключается договор ранней оплаты счетов, устанавливаются условия поставки или предоставления услуг взамен на скидку от поставщика.

Поставщики проводят сделки с динамическим дисконтированием, потому что это наиболее простой и выгодный способ привлечения внешнего финансирования.

Заказчикам выгодно оплачивать сделку ранее срока поставки товара, предоставления услуги, так как они получают скидку за раннюю оплату.

Подводя итоги, можно сделать вывод о том, что оба контрагента имеют выгоду. Заказчик получает скидку на заказ: чем раньше он оплатит счёт, тем большая будет выгода. Поставщик в свою очередь имеет выгоду за счет сокращения затрат на привлечение финансирования, так как сможет пополнить свой оборотный капитал раньше планируемого срока, а следовательно, сможет произвести большее количество продукции, развивать предприятие.

2.2 Технология блокчейн

В основе технологии блокчейн лежит выстроенная по определённым правилам цепочка блоков с информацией о транзакциях (операциях) внутри системы.

Цепочка блоков представляет собой связный список, где каждый блок имеет заголовок, включающий в себя хеш-суммы информации в блоке, предыдущего блока, транзакций; дополнительную служебную информацию.

Транзакции — это операции, которые могут быть осуществлены только полностью. Таким образом, транзакции, хранящиеся в блоках, могут представлять собой как денежные операции, так и операции по внесению новых данных в базу данных, построенную на децентрализованной системе с технологией блокчейн. Именно поэтому широкое применение технология блокчейн находит в бизнесе, где очень часто заключаются сделки.

Безопасность при хранении информации в структуре данных с блокчейном достигается с помощью хеш-сумм, которые организуют высокий уровень связности блоков.

Механизм консенсуса является способом прийти к всеобщему соглашению для принятия изменений в системе. Алгоритм консенсуса представляет собой набор математических правил и функций, с помощью которых система достигает соглашения между всеми участниками распределённой сети.

2.3 Смарт-контракт блокчейна Ethereum

Умный контракт (англ. Smart contracts) — электронный алгоритм, описывающий набор условий, выполнение которых влечет за собой некоторые события в реальном мире или цифровых системах. Для реализации умных контрактов требуется децентрализованная среда, полностью исключающая человеческий фактор, а для возможности использования в умном контракте передачи стоимости требуется цифровая валюта [1].

Смарт-контракт представляет собой фрагмент программного кода, хранящийся в блоках. Он приводится в действие транзакциями и имеет возможность читать данные или писать их в блокчейн. Мощный механизм поддержки смарт-контрактов делает блокчейн-платформу Ethereum универсальной с любой областью применения.

2.4 Сравнение аналогов разрабатываемой системы

Существует несколько платформ динамического дисконтирования:

1) SmartDiscount [16];
2) KEPT [17];
3) isource [18].

Перечисленные платформы предоставляют схожую функциональность: публикация предложения о сделке, настройка дисконта для заказчика, заключение сделки подписанием онлайн договора.

Платформы не используют технологию блокчейн для записи данных о сделке. Они разработаны по модели централизованного реестра, то есть вся информация размещается на серверах владельца платформы, а значит теоретически подвержена подмене данных и другим атакам, которые изучает «Информационная безопасность».

3 Требования к разрабатываемой платформе

После проведённого анализа предметной области было сформировано видение разрабатываемой платформы.

1) Веб-приложение должно быть разработано для использования в браузерах на компьютере, смартфоне, планшете.
2) Для реализации платформы должен быть использован блокчейн Ethereum со смарт-контрактами на языке Solidity.
3) Данные о пользователях и их сделках должны храниться на защищённом сервере баз данных.
4) Платформа для осуществления сделок с динамическим дисконтированием должна предоставлять функциональность:
   - зарегистрировать личный аккаунт в системе;
   - авторизоваться по логину и паролю;
   - заполнить анкету о пользователе;
   - создать новое предложение о сотрудничестве;
   - открыть поисковую форму;
   - настроить фильтры для поиска предложений о заключении сделки;
   - согласовать условия и размер скидки при заключении сделки с контрагентом;
   - заключить соглашение;
   - подтвердить заключение сделки;
   - подтвердить выполнение сделки;
   - просмотреть список заключённых партнёрских соглашений.

Разрабатываемая блокчейн-платформа должна поддерживать бизнес-процесс, изображённый на рисунке 1.

На диаграмме изображены операции и переходы между ними с проверками условий. Таким образом, описывается порядок операций от момента создания предложения о сотрудничестве до заключения сделки на условиях динамического дисконта.

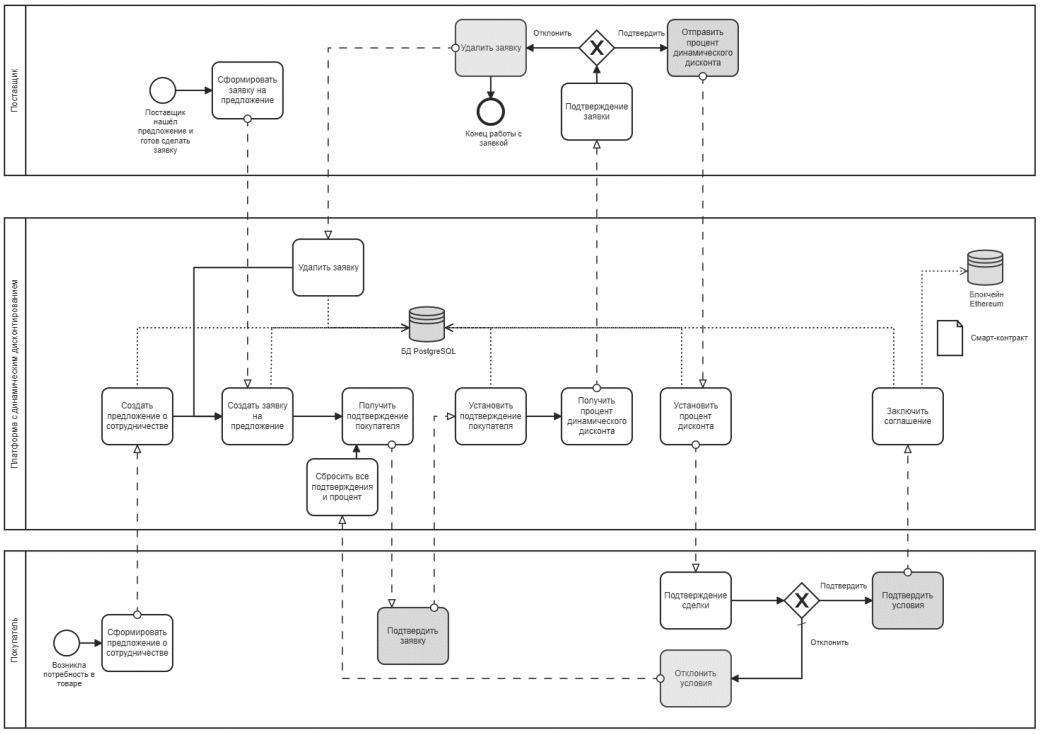


Рис.1. BPMN диаграмма

4 Архитектура разработанного программного комплекса

Интернет-платформа имеет архитектуру клиент-серверного приложения с веб-интерфейсом. На рисунке 2 изображена компонентная модель разработанной блокчейн-платформы.

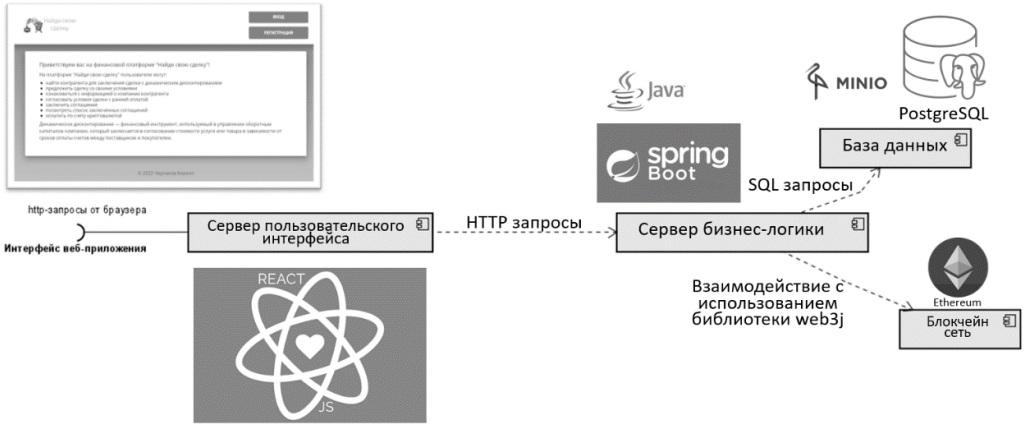


Рис.2. Компонентная модель системы

4.1 Сервер пользовательского интерфейса

Сервер пользовательского интерфейса представляет собой сервер на платформе Node.js. Node.js — это серверная среда выполнения JavaScript, выполняющая код JavaScript.

Сервер занимается обработкой запросов от браузеров пользователей, перенаправляет HTTP-запросы другому внутреннему серверу для обработки всех операций (back-end сервер) и возвращает ответ пользователю в виде веб-страницы.

Интерфейс разработан с использованием библиотеки React для языка программирования JavaScript. React — это интерфейсная платформа для создания пользовательского интерфейса [11].

4.2 Внутренний сервер бизнес-логики

Java–Spring приложение, которое взаимодействует с базой данных PostgreSQL для хранения данных о пользователях, учётных записях; с блокчейн-сетью с помощью библиотеки Web3j для простого взаимодействия с блокчейном и смарт-контрактами.

Java Spring Framework [7] — это универсальный фреймворк для разработки веб-приложения, который включает в себя полезные модули для взаимодействия с системами управления базами данных, для обработки HTTP-запросов, JSON сериализации и десериализации данных из тела запросов. Данный фреймворк позволяет передать управление зависимостями между объектами классов компьютеру, то есть сам Spring занимается внедрением зависимостей в объекты классов. Именно так реализован принцип инверсии контроля зависимостями.

Web3j — это легковесная, высокомодульная, реактивная Java-библиотека для работы со смарт-контрактами и интеграции с блокчейнами Ethereum [9]. Эта библиотека позволяет нам работать с блокчейнами Ethereum без дополнительных накладных расходов, связанных с написанием собственного кода интеграции для платформы. С помощью web3j можно по упрощённой схеме реализовать приложение для взаимодействия с блокчейн-сетью, ведь внутри библиотеки уже реализованы все необходимые методы для подключения и взаимодействия с блокчейном [10].

4.3 Сервер базы данных PostgreSQL

Сервер базы данных PostgreSQL, который используется в Java-приложении с помощью Spring Data JPA.

Spring Data — модуль фреймворка Spring для взаимодействия с сущностями базы данных. Данный механизм организовывает сущности в репозитории для выполнения запросов с точки зрения Java-программы. Таким образом, Spring Data является удобным интерфейсом для работы с базой данных с точки зрения объектно-ориентированного программирования. Все сущности автоматически транслируются в объекты Java для работы в программе.

В базе данных хранится информация о пользователях платформы, предложениях о сотрудничестве, заявках на заключение соглашений. Условия сделки, информация о контрагентах и размер скидки в процентах годовых хранится в смарт-контракте в блокчейн-сети Ethereum.

4.4 Блокчейн-сеть

Во время работы программной системы разворачивается локальная блокчейн-сеть Ethereum с помощью программы Ganache. Это поможет организовать собственную блокчейн-сеть, чтобы обучиться правильному взаимодействию с ней и не затрагивать глобальную сеть [12].

Ganache — программный инструмент для разворачивания и запуска приватного блокчейна. Таким образом, нам представляется возможность работать с блокчейном в локальном варианте, не используя глобальную сеть блокчейна. Программа реализовывает все необходимые операции по поддержанию работы блокчейн-сети. Данное программное решение необходимо для реализации и отладки такого большого проекта.

Условия сделки и информация о контрагентах хранятся в смарт-контрактах, которые имеют программные методы взаимодействия для изменения состояния смарт-контракта посредством транзакций. Смарт-контракт представляет собой класс с набором полей и методов для изменения состояния. На разработанной платформе методы смарт-контрактов используются для проведения оплаты, изменения статуса сделки, проверки выполнения условий соглашения.

Функциональность смарт-контракта изображена на рисунке 3.

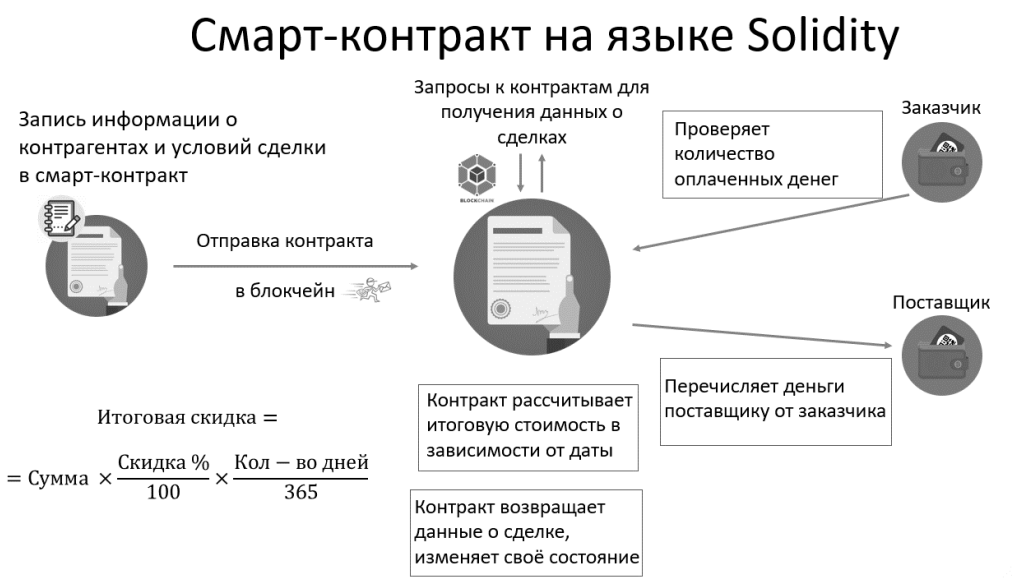


Рис.3. Функциональность смарт-контракта

Смарт-контракт для осуществления сделки с динамическим дисконтом принимает входные параметры дисконта (сроки сделки, предмет сделки, коэффициенты динамического дисконта, дополнительные условия). Для обновления состояния смарт-контракта имеются методы взаимодействия:

1) проверить статус сделки;
2) получить условия динамического дисконта;
3) обновить условия заключения сделки;
4) обновить стоимость товара или услуги.

Формула скидки за раннюю оплату:

$$\text{Итоговая скидка} = \text{Сумма} \times \frac{\text{Скидка \%}}{100} \times \frac{\text{Кол} - \text{во дней}}{365}$$

Формула динамического дисконтирования:

$$SD = S \times \left(1 - \frac{P}{100} \times \frac{d}{365}\right)$$

SD — итоговая сумма сделки;

S — стоимость товара или услуги;

D — количество дней до поставки;

P — ставка дисконтирования.

5 Результаты разработки

5.1 Разработка Java-Spring приложения

В ходе разработки проекта было реализовано программное обеспечение на языке программирования Java [6] с использованием фреймворка для разработки веб-приложения Spring [7].

Программное обеспечение сервера бизнес-логики полностью соответствует техническим требованиям, описанным во 2 главе. Сервер взаимодействует с базой данных PostgreSQL, хранилищем изображений Minio S3, блокчейн-сетью Ethereum.

Разработанная блокчейн-платформа использует смарт-контракты для фиксирования условий сделок, информации о контрагентах, сроках поставки, размере скидки в процентах годовых (рисунок 4).

При заключении сделки, смарт-контракт отправляется в сеть блокчейн для хранения данных и исполнения программного кода.

После заключения сделки контрагенты видят информацию о соглашении, которая была загружена из смарт-контракта.

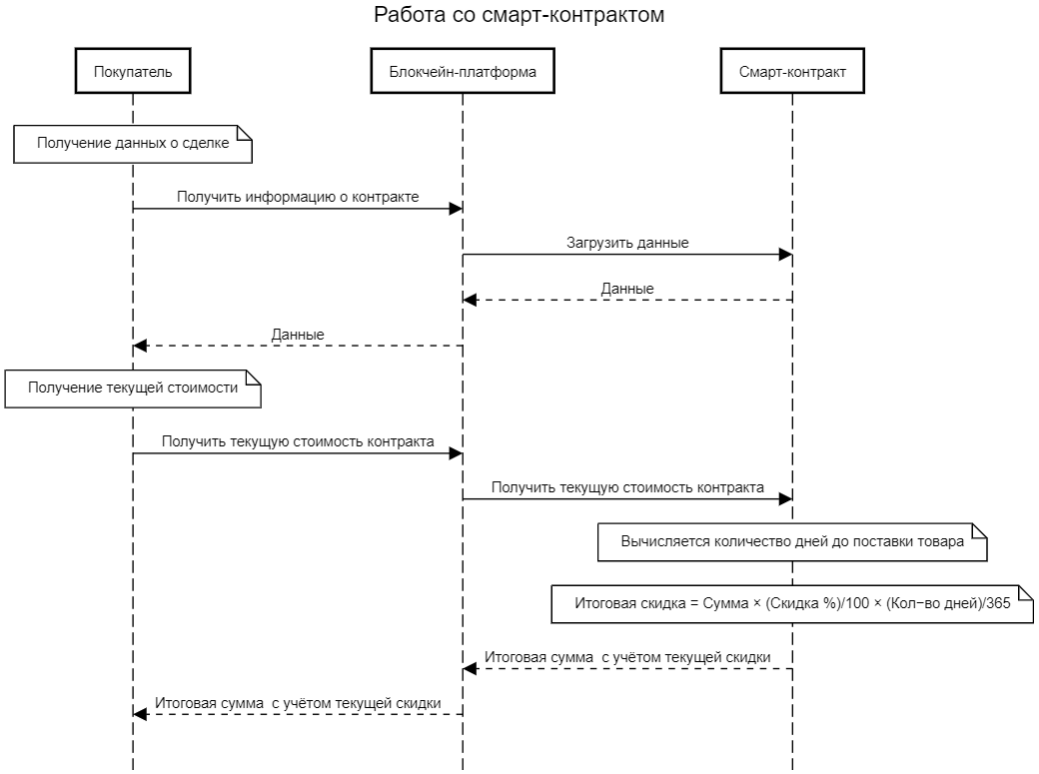


Рис.4. Взаимодействие со смарт-контрактом

Смарт-контракт производит транзакцию перевода денежных средств с кошелька заказчика на кошелёк поставщика, подсчитывая количество оплачиваемых денег в зависимости от количества дней до назначенной поставки товара.

Сервер бизнес-логики хранит информацию о пользователях, предложениях о сотрудничестве, отправляет данные о заключённых сделках в блокчейн-сеть, взаимодействует со смарт-контрактами заключённых сделок.

Сервер бизнес-логики взаимодействует с сервером пользовательского интерфейса по протоколу HTTP, выдавая информацию в формате JSON.

Каждый запрос проверяется на доступ к ресурсам платформы посредством авторизации каждого запроса специальным JSON Web Token [15]. По нему сервер распознаёт пользователя, запросившего данные от системы. Защитные токены выдаются в ответе на авторизацию в системе по логину и паролю.

5.2 Разработка пользовательского сайта

В ходе разработки проекта был реализован сайт для пользователей с графическим интерфейсом. Программное обеспечение разработано на языке JavaScript с использованием фреймворка React.

Сайт доступен из сети Интернет с компьютера, телефона, планшета.

Все запросы от пользовательского сайта поступают серверу бизнес-логики по протоколу HTTP. Пользовательский браузер получает информацию от внутреннего сервера в формате

JSON, после чего отображает данные в графическом виде на экране пользователя (рисунки 5 и 6).

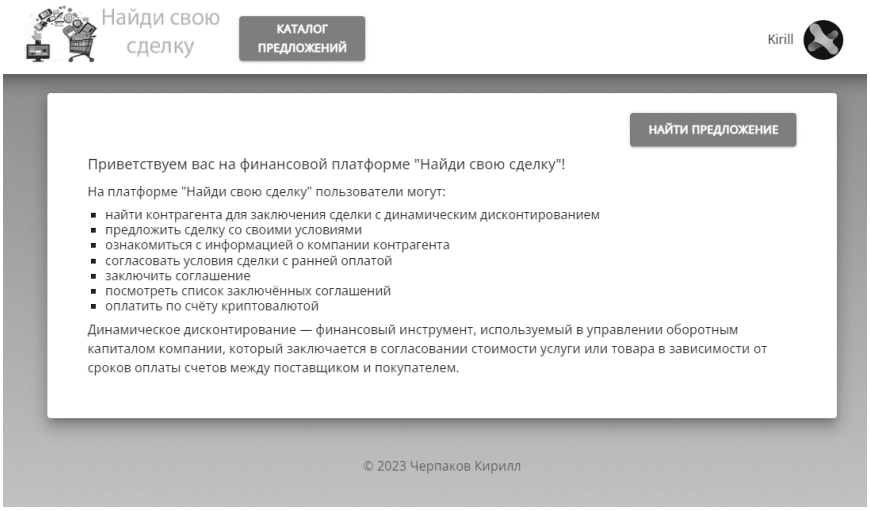


Рис.5. Главная страница сайта

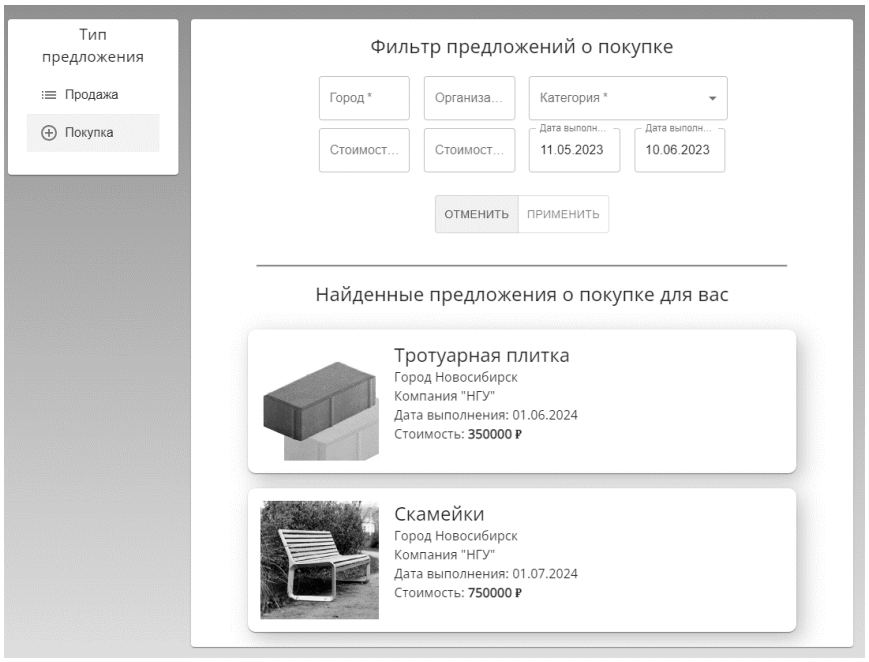


Рис.4. Страница поиска предложений о сотрудничестве

6 Заключение

По итогам работы была разработана блокчейн платформа для осуществления сделок с динамическим дисконтированием. В результате получен рабочий программный комплекс, который можно запустить и работать с ним, используя компьютер с доступом в Интернет.

В дальнейшей работе планируется расширение функциональности программного комплекса, совершенствование пользовательского интерфейса. Планируется разработать оплату сделки из личного кабинета пользователя. Настроить взимание пени платформой за проведённые транзакции и заключение сделки. Разработать отдельный микросервис, отвечающий за рассылки уведомлений о предложениях о сотрудничестве на электронную почту пользователя. Разработать возможность переписываться пользователям в пределах платформы.

Литература

[1] Команов, П. А. Исследование безопасности смарт-контрактов Ethereum / П. А. Команов, Х. Ю. Ревазов, Д. А. Тавасиев // Международный научно-исследовательский журнал. – 2021. – № 1-1(103). – С. 80-83. – DOI 10.23670/IRJ.2021.103.1.011.

[2] Супибекова, А. К. Применение smart-contract в платформе Ethereum / А. К. Супибекова, Г. А. Эсеналиева // Вестник Кыргызского государственного университета строительства, транспорта и архитектуры им. Н.Исанова. – 2017. – № 2(56). – С. 73-77. – EDN YMBAPA.

[3] Круглик, Р. И. Создание смарт-контакта на блокчейне Ethereum / Р. И. Круглик // Постулат. – 2019. – № 9(47). – С. 32.

[4] Приказчиков, Д. Е. Механизмы консенсуса proof of work и proof of stake на примере платформы Ethereum / Д. Е. Приказчиков, С. В. Малахов // Инновации. Наука. Образование. – 2021. – № 34. – С. 1093-1098. – EDN LGSKDZ.

[5] Ходаков, А. Д. Особенности разработки смарт-контракта на базе блокчейн платформы Ethereum / А. Д. Ходаков, С. А. Зайкова // Современные инструментальные системы, информационные технологии и инновации : Сборник научных трудов XIII Международной научно-практической конференции, Курск, 15–16 марта 2018 года / Ответственный редактор А.А. Горохов. – Курск: Закрытое акционерное общество "Университетская книга", 2018. – С. 205-207.

[6] Документация по использованию языка программирования Java. [Электронный ресурс] // URL: https://docs.oracle.com/en/java/ (дата обращения: 15.11.2022).

[7] Документация по использованию фрейморвка Spring. [Электронный ресурс] // URL: https://spring.io (дата обращения: 15.11.2022).

[8] Документация языка Solidity [Электронный ресурс] // URL: https://docs.soliditylang.org/en/latest/ (дата обращения 19.01.2023).

[9] Документация библиотеки Web3j [Электронный ресурс] // URL: https://github.com/web3j/web3j (дата обращения 20.01.2023).

[10] Клиенты Ethereum, использующие Web3j [Электронный ресурс] // URL: https://www.baeldung.com/web3j (дата обращения 24.01.2023).

[11] Документация фреймворка React [Электронный ресурс] // URL: https://react.dev/reference/react (дата обращения 30.01.2023).

[12] Локальная блокчейн-сеть Ethereum в программе Ganache [Электронный ресурс] // URL: https://trufflesuite.com/ganache/ (дата обращения 30.01.2023).

[13] Документация по использованию Docker. [Электронный ресурс] // URL: https://docs.docker.com/reference (дата обращения: 15.04.2023).

[14] Что такое динамический дисконт [Электронный ресурс] // URL: https://getfinance.ru/soveti/faktoring-ili-dinamicheskoe-diskontirovanie-chto-vybrat/ (дата обращения 20.04.2023).

[15] JSON Web Token [Электронный ресурс] // URL: https://www.baeldung.com/java-json-web-tokens-jjwt (дата обращения 03.02.2023).

[16] Платформа динамического дисконтирования «SmartDiscount» [Электронный ресурс] // URL: https://smartdiscount.ru/ (дата обращения 02.05.2023).

[17] Платформа динамического дисконтирования «KEPT» [Электронный ресурс] // URL: https://dd.kept.ru/ (дата обращения 02.05.2023).

[18] Платформа динамического дисконтирования «isource» [Электронный ресурс] // URL: https://www.isource.ru/ (дата обращения 02.05.2023).

Прунинг свёрточных нейронных сетей через проецирование свёрток

Чудаков Д.С.[1], Бериков В.Б.[1]

[1]*Институт математики им. С.Л. Соболева, пр. Академика Коптюга, д. 4, г. Новосибирск, 630090, Россия.*

DimChudo98@yandex.ru, berikov@math.nsc.ru

Аннотация. *Прунинг — эффективный и широко распространённый метод оптимизации уже обученной нейронной сети. Прунинг — это удаление весов из уже обученной НС с целью уменьшения её вычислительной сложности. Структурированные методы прунинга позволяют добиться прироста в производительности и при этом не накладывают ограничений на вычислительное устройства, как например это требуют методы квантования или неструктурированного прунинга. Одной из главных проблем структурированного прунинга является сложность выбора наименее значимых каналов. В современных свёрточных сетях из-за сложной архитектуры фильтр в начале НС может непосредственно влиять на фильтр в конце НС, что усложняет вычисление значимости фильтра. В данной статье предлагается иной подход, позволяющий получить уменьшенную НС в один проход. Предлагается обучаемым образом проецировать свёртки в пространство меньшей размерности, избегая отдельных этапов с поиском незначимых фильтров. В исходную НС перед и после каждого слоя свёртки добавляются свёртки с размером ядра равным 1, в процессе обучения НС эти свёртки выучат корректно проецировать вектора признаков в пространство меньшей размерности, после чего их можно будет объединить с исходным слоем свёртки.*

Ключевые слова: Машинное обучение, нейронные сети, ускорение нейронных сетей, прунинг.

1 Введение

Структурированный прунинг – распространенный метод оптимизации нейронных сетей (НС), во многом это связано с тем что данный метод не накладывает никаких ограничений на вычислительное устройство. Например, методы квантования требуют от вычислительного устройства эффективной поддержки целочисленных вычислений малой разрядности, а методы неструктурированного прунинга требуют эффективной поддержки умножения разреженных матриц. Однако структурированный прунинг очень чувствителен к выбору наименее значимых фильтров. Наивные методы основывающиеся на абсолютном значении весов или использующие значения дисперсий выходных карт признаков по умолчанию никак не учитывают связанность слоёв между друг другом. Также наивные методы не используют данные на которых обучалась НС и целевое значение выхода НС. Более современные методы [1], аккумулируют значения градиентов на всех фильтрах НС, позволяя более точно вычислять значимость фильтров. Однако эти подходы требуют постепенного уменьшения размера НС, что значительно замедляет процесс ускорения НС. В данной работе предлагается переформулировать задачу прунинга, как поиск подсети меньшей размерности. Для того чтобы на практике это было возможно, было принято несколько предположений:

1. Нейронная сеть состоит только из стандартных свёрточных слоёв и не использует, например, независимые по глубине свёртки (Depth-wise convolutional). DWS свёртки невозможно спроецировать в DWS свертки меньшей размерности непрерывным образом.

2. Необходимое число фильтров для удаления из каждого слоя заранее известно. Это избавляет нас от необходимости учитывать ещё одну переменную и упрощает сравнение с другими методами прунинга.

2 Постановка задачи

Задача прунинга формулируется как модификация задачи обучения нейронной сети (1):

$$min_W E(D, W) + \lambda||W||_0 \quad (1)$$

Где W – веса НС, D – обучающий набор данных, E – функция потерь, λ – гипер параметр отвечающий за степень сжатия НС.

Норма l_0 – представляет собой число ненулевых элементов. К сожалению не существует эффективного способа минимизировать эту норму, т.к. задача не является выпуклой и является NP-трудной, что требуется переборного поиска.

Часто задачу переформулируют с предположением о независимости фильтров относительно друг друга (2):

$$I_m = (E(D, W) - E(D, W|w_m = 0))^2 \quad (2)$$

Где I_m – это оценка значимости m-ого фильтра. Даже в такой формулировке вычисление значимости для каждого фильтра будет очень трудоёмким, поэтому для его оценки используют разложение функции потерь в ряд Тейлора по весам [1].

3 Метод

Рассмотрим операцию проецирования оператора свёртки в пространство меньшей размерности. Возьмём за основу свёртку с числом входных и выходных каналов равных N, и размером ядра свёртки равным k. Наша цель получить свёртку с числом входных и выходных каналов N / r, где r степень сжатия НС. Для этого до целевой свёртки расположим свёртку с размером ядра 1, числом входных каналов N / r и числом выходных каналов N. После целевой свёртки расположим свёртку с размером ядра 1, числом входных каналов N и выходных N / r. Таким образом эта композиция из трёх операторов будет преобразовывать карту признаков с N / r каналами в N / r каналов (3, 4).

$$input_N \rightarrow TargetConv_{kxk} \rightarrow input_N \quad (3)$$

$$input_{N/r} \rightarrow Conv_{1x1} \rightarrow TargetConv_{kxk} \rightarrow Conv_{1x1} \rightarrow output_{N/r} \quad (4)$$

Этот блок из трёх свёрток можно заменить одной эквивалентной свёрткой с размером ядра k, и числом входных и выходных каналов N / r. Для этого нам нужно будет применить наши дополнительные свёртки к ядру целевой свёртки. Окружив каждую свёртку в нейронной сети такими проецирующими свёртками мы фактически уменьшим число каналов для каждого слоя в r раз.

Следующим этапом необходимо оптимизировать эти дополнительные свёртки так, чтобы минимизировать функцию потерь НС на целевом обучающем наборе данных. На данном этапе мы получаем наиболее подходящие проекции для каждого фильтра. В отличие от методов ищущих проекцию для каждой свёртки напрямую через метод главных компонент, данный метод не использует никаких дополнительных предположений о свойствах весов НС.

Большинство методов прунинга в том или ином виде используют допущения о свойствах нейронной сети такие как:

1. Независимость слоёв внутри нейронной сети.
2. Прямая зависимость абсолютного значения веса и его значимости.
3. Прямая зависимость дисперсии выхода слоя и его значимости.

Предложенный метод не использует никаких допущений о свойствах нейронной сети и ищет целевую уменьшенную НС непосредственно в процессе градиентного спуска.

Далее веса нейронной сети обучаются вместе с весами проецирующих свёрток, для того чтобы добиться максимальной точности НС. На финальном этапе мы объединяем веса целевых свёрток с проецирующими и получаем итоговую ускоренную НС.

Стоит отметить что в теории методы подобные [1] выбирающие наименее значимые веса могут добиться сравнимой точности согласно гипотезе лотерейного билета [2]. Согласно этой гипотезе с самого начала обучения НС неявно выбираются лотерейные билеты – наиболее точная нейронная подсеть и существуют некоторые наименее значимые веса, которые можно удалить сразу и после дообучения получить максимальную точность. Но в случае с методом использующим разложение Тейлора, необходимо за раз выбрасывать как можно меньше фильтров, т.к. разложение верно лишь в небольшой окрестности текущих весов. А такой итеративный подход значительно увеличивает время работы алгоритма.

4 Результаты

Для эксперимента мы возьмём НС ResNet18, это самая легковесная НС семейства ResNet. Благодаря её размерам, мы имеем возможность быстро проверять гипотезы и подбирать гиперпараметры. Также необходимость использовать нейронную сеть семейства ResNet обуславливается тем, что это одна из немногих современных и популярных архитектур, которые не используют независимые по глубине свёртки (Depth-wise convolutional layer).

Таблица 1. Сравнительные точности нейронной сети ResNet18 [3] на задачу ImageNet [4].

Метод.	***Точность после прунинга***	***Точность после дообучения***
Importance Estimation for Neural Network Pruning [1] Степень сжатия 25%	6%	59%
Наш метод. Степень сжатия 25%	55%	60%
Importance Estimation for Neural Network Pruning [1] Степень сжатия 50%	1.9%	52.9%
Наш метод. Степень сжатия 50%	50.1%	55.8%

Изначальная точность ResNet-18 на задаче классификации ImageNet составляет 69 процентов. В данном эксперименте мы нацелились проверить точности подходов в сжатом эксперименте, где использовалась небольшая обучающая подвыборка (10 процентов от изначального объёма). Небольшая выборка позволяет значительно сократить время на полную работу алгоритма. Использованная нами стратегия обучения:

Число эпох = 50

Число предварительных эпох = 10

Шаг обучения = 0.001

Алгоритм оптимизации = RAdam [5]

Расписание обучения = Косинусный отжиг

5 Заключение

Предоставленный нами метод позволяет значительно упростить процедуру прунинга нейронных сетей. Нами была переформулирована задача прунинга так, чтобы её можно было решать в один проход (end-2-end), таким образом мы смогли избавить от большинства предположений которые необходимы для реализации методов прунинга основанных на выборе наименее значимых каналов. Нам удалось добиться увеличения точности нейронной сети при той же степени сжатия. Методы основанные на разложении Тейлора работают хуже при выборе большого количество отбрасываемых каналов, мы продемонстрировали что разрыв в точностях увеличивается вместе со степенью сжатия сети.

Литература

[1] Molchanov P, Mallya A, Tyree S, Frosio I, Kautz J. Importance estimation for neural network pruning. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2019 (pp. 11264-11272).

[2] Frankle J, Carbin M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. arXiv preprint arXiv:1803.03635. 2018 Mar 9.

[3] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770-778).

[4] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC. Imagenet large scale visual recognition challenge. International journal of computer vision. 2015 Dec;115:211-52.

Разработка и верификация программ обработки графов в предикатном программировании

В.И. Шелехов [1,2], И.В.Каблуков [1], Д.Р. Янбулатов [2]

[1] *Институт систем информатики им. А.П.Ершова СО РАН, пр. Лаврентьева 6, г.Новосибирск, 630090, Россия.*
[2] *Новосибирский государственный университет*

vshel@iis.nsk.su, vanyok05@mail.ru, d.yanbulatov@g.nsu.ru

Аннотация. *Традиционно средства для работы с графами реализуются в виде библиотек в объектно-ориентированной технологии. Требования эффективности вынуждают существенно усложнять интерфейс пользователя. В нашем подходе графы представлены множествами – математическими структурами на языке функционального (предикатного) программирования. Это дает возможность проведения дедуктивной верификации программ на графах. Описывается расширение языка предикатного программирования P средствами работы с графами. Метод разработки программ и их дедуктивной верификации представлен на простой программе определения степеней вершин графа. Эффективность предикатных программ достигается применением оптимизирующих трансформаций.*

Ключевые слова: алгоритмы на графах, предикатное программирование, GraphBLAS, дедуктивная верификация, система верификации Why3

1 Введение

Эффективная и корректная реализация алгоритмов на графах является сложнейшей задачей, в частности, для высокоэффективных кластеров GPU. В последние семь лет для повышения эффективности алгоритмов на графах используется метод линейной алгебры для разреженных матриц на базе стандартного интерфейса GraphBLAS [23], базирующегося на математической формализации [10] и предоставляющего исчерпывающий набор операций для эффективной реализации алгоритмов на графах. На базе стандарта GraphBLAS [23] разработано более семи разных библиотек для работы с графами. Это SuiteSparse [20], GBTL [19], GraphBLAST [21] и другие [4].

Интерфейс GraphBLAS [23] весьма сложен, в частности потому, что ориентирован на высокую степень оптимизации графовых алгоритмов при реализации библиотеки на его базе. Однако во многих приложениях не требуется предельной оптимизации. Учитывая это, библиотека LAGraph [4] предоставляет существенно более простой интерфейс. Тем не менее, во всех существующих подходах реализация графовых алгоритмов является сложной. О дедуктивной верификации даже речи не идет. Конечно, в мире достаточно много исследований по дедуктивной верификации графовых алгоритмов, но не в практической реализации.

Объектно-ориентированная технология, применяемая в библиотеках графовых алгоритмов, призвана скрывать внутри библиотеки детали сложной реализации, предоставляя пользователю простой интерфейс. В действительности ровно наоборот. Потребности оптимизации алгоритмов вынуждают существенно усложнять интерфейс.

В настоящей работе предлагается развитие технологии предикатного (функционального) программирования [1,6,7, 9,11,14,17] для разработки графовых программ на языке предикатного программирования P [1] с возможностью применения набора

оптимизирующих трансформаций к предикатной программе для получения эффективной программы на императивном языке типа Си. Язык P ориентирован также на использование в качестве языка публикации алгоритмов на графах вместо псевдокода. Другое направление исследований – дедуктивная верификация предикатных программ обработки графов.

В языке P для представления графов используется множества. Аналогичный подход при построении типов данных используется в системах моделирования и верификации Event-B [15] и TLA+ [16]. Однако там не стоит задача получения эффективного кода. Эффективная трансляция программ, оперирующих множествами, проблематична. В предикатном программировании для этого используются оптимизирующие трансформации.

Эффективность предикатных программ достигается применением оптимизирующих трансформаций [2]. Результатом трансформаций является эффективная императивная программа в стиле языка Си.

Во втором разделе данной работы описываются основные конструкции и расширения языка предикатного программирования P [1]. Введены операции линейной алгебры для матриц и векторов на полукольцах для эффективной реализации на графических процессорах. Эти конструкции определены на математическом уровне GraphBLAS [10], как в языке PyGB [3]. В третьем разделе определяются основные структуры данных для представления графов в языке P. Предикатная программа определения степеней вершин представлена в четвертом разделе. В пятом разделе описывается ее дедуктивная верификация.

2 Предикатное программирование

Предикатное программирование [1,6,7,9,11,14,17] – это чистое функциональное программирование с нотацией в стиле языка Си. Наряду с функциями используются предикаты (операторы). Расширения предикатного программирования P [1] по отношению к функциональному программированию, в частности гиперфункции, в принципе могут быть представлены конструкциями классического функционального программирования.

2.1 Предикатная программа

Полная предикатная программа на языке P [1] состоит из набора предикатных программ вида:

<program name>**(**<argument declarations>**:** <result declarations>**)**
pre <precondition>
post <postcondition>
measure <expression>
{ <statement> }

Предусловие и постусловие являются формулами на языке исчисления предикатов. Они обязательны при дедуктивной верификации [6,7,9,11,14,17]. Мера задается только для рекурсивных программ и используется для доказательства их завершения.

Ниже представлены основные конструкции языка P [1]: оператор присваивания, блок (оператор суперпозиции), параллельный оператор, условный оператор, вызов программы и описание переменных, используемое для аргументов, результатов и локальных переменных.

<variable> = <expression>
{<statement1>; <statement2>}
<statement1> || <statement2>
if (<boolean expression>**)** <statement1> **else** <statement2>
<program name> (<argument list>**:** <result list>)
<type> <space> <list of variable names>

Операторы, входящие в параллельный оператор, независимы по результатам. Циклы и указатели запрещены в языке P.

Допускается использование функциональной формы. Вызов предиката B(x**:** y), где y – список переменных, эквивалентен оператору присваивания |y| = B(x) или просто y = B(x), где B(x) интерпретируется как вызов функции. В частности, оператор суперпозиции вида B(x**:** z); C(x, z**:** y) можно записать так: y = C(x, B(x)). Оператор присваивания вида y = E в конце программы, где y – список результатов программы, может быть заменен выражением E.

Эффективность предикатных программ достигается применением оптимизирующих трансформаций [2]. Они определяют отличную от классической оптимизацию среднего уровня

с преобразованием предикатной программы в эффективную императивную программу на языке подобном Си.

2.2 Гиперфункции

Гиперфункция – это программа с несколькими ветвями результатов [1, разд. 3.1, 3.3]. Гиперфункция B(x: y: z) имеет две ветви результатов y и z. Выполнение гиперфункции заканчивается вычислением результатов одной из ветвей; результаты других ветвей не вычисляются. Ветвь может не иметь результатов. Вызов гиперфункции B(x: y: z) обычно используется в следующем операторе switch:

switch B(x) { **case** y: <statement1> **case** z: <statement2>}

Используется также другая форма оператора switch:

switch B(x: y: z) { **case** <statement1> **case** <statement2>}

Ограничитель **switch** обычно опускается. Оператор switch в некотором роде является гибридом условного оператора и оператора суперпозиции. Использование гиперфункций делает программу быстрее и короче [8].

После последнего исполняемого оператора программы-гиперфункции указывается, какой ветвью гиперфункции завершается программа. Оператор перехода #1 фиксирует завершение по первой ветви, а #2 – по второй.

Если одна из альтернатив **case** в операторе switch становится пустой, оператор switch вырождается. Взамен может появиться оператор вызова гиперфункции вида, например, B(x: y: z #3). При завершении второй ветвью вызова вырабатывается значение z; далее программа, содержащая данный вызов, завершает исполнение своей третьей ветвью. Если вызов B(x: y: z #3) завершается первой ветвью, то вырабатывается значение y, и далее программа продолжается со следующего оператора после данного вызова гиперфункции.

Возможна ситуация, когда заведомо известно, что одна из ветвей вызова гиперфункции будет недостижимой. В этом случае в позиции недостижимой ветви вставляется оператор ##. Например, B(x: y: ##).

2.3 Типы

Типы **bool**, **nat**, **int**, **real** и **char** являются *примитивными*. Значение типа **struct**(T_1 f_1, ..., T_n f_n) - это *структура* из n значений, имеющих *поля* f_1, f_2,..., f_n и типы T_1, T_2,..., T_n, соответственно. Тип *кортеж* (T_1, T_2,..., T_n) определяет упорядоченный набор значений типов T_1, T_2,..., T_n.

Значение типа **array**(T_e, T_i) представляет собой *массив* с элементами типа T_e и индексами конечного типа T_i. Конструкция **for** (T_i j) <expression> определяет новый массив. Каждому элементу массива с индексом j, принадлежащему T_i, присваивается значение <expression>, зависящее от j. Для массива A конструкция A **with** (i: x) определяет новый массив, заменяя элемент массива A[i] значением выражения x. Конструкция A **with** (i: x, j: y) определяется как (A **with** (i: x)) **with** (j: y).

Тип **array**(T_e, T_i, T_j) определяет *матрицу* с индексами конечных типов T_i и T_j. Для матрицы A конструкции A[i,] и A[, j] определяют массивы (векторы), которые являются i-й строкой и j-м столбцом матрицы A, соответственно. Операция A# определяет транспонирование матрицы A.

Пусть $A^{m\times n}$ (с m строками и n столбцами) и $B^{n\times k}$ - матрицы, c^n и d^m - векторы. Для *полукольца* с произвольными операторами сложения $\oplus$ и умножения $\otimes$ [10], вводятся операции умножения матрица на матрицу $A \oplus.\otimes B$, матрица на вектор $A \oplus.\otimes c$ и вектор на матрицу $d \oplus.\otimes A$ для компактной и эффективной реализации графовых алгоритмов на GPU.

Значением *алгебраического типа* **union**(K_1, …, K_n) является один из конструкторов K_1, …, K_n. Конструктор определяется именем конструктора и набором полей, подобно структуре; набор полей может быть пустым. Типичными объектами алгебраических типов являются списки и деревья. *Список* определяет последовательность элементов некоторого типа T. Описание типа списка выглядит следующим образом:

```
type list (type T) = union( nil,  cons(T car, list(T) cdr) );
```

Тип list имеет два конструктора: nil, обозначающий пустой список, и cons, определяющий список, первый элемент которого представлен полем car, а остальная часть списка определяется полем cdr. Обработка некоторого объекта s типа list реализуется оператором switch с конструкторами списков в качестве альтернатив:

```
switch s { case  nil: <statement1> case  cons(x, v): <statement2>}
```

Здесь локальные переменные для элемента x и списка v используются в <statement2>. Ниже пример функции, вычисляющей длину списка.

```
len(list(int) s): nat
{    switch s {
          case nil: 0
          case cons(a, u): len(u)+1
}};
```

Любой элемент типа **set**(T) - это *множество*, которое является конечным подмножеством типа T. Тип T называется *базовым типом* для типа **set**. *Пустое множество* представляется как {}. Конструкция {x} - это множество с единственным элементом x. Результатом операции add(x, s) является множество s∪{x}. Операция s+u задает объединение множеств s и u. Операция s^u задает пересечение множеств s и u. Операция s\u определяет вычитание множества u из множества s. Конструкция |s| обозначает число элементов множества s. Гиперфункция pick(s : : x, s1) недетерминировано выбирает произвольный элемент x, если он существует. Множество s1 – это остальное множество без элемента x, то есть s = s1∪{x}. Первая ветвь – это результат выбора гиперфункции, когда множество s пусто. Заметим, что в случае непустого множества s гиперфункция pick быстрее, чем последовательное применение двух действий: проверка на пустоту и выбор произвольного элемента из s. Гиперфункция pick определяется следующей спецификацией:

```
pick(set(T) s : : T x, set(T) s1) pre 1: s=∅ post 2: s=s1∪{x} & x∈s & x∉s1
```

Предусловие по первой ветви: s=∅. Предусловие по второй ветви: s≠∅. Оно не указывается, поскольку является дополнением к предусловию первой ветви. Постусловие по первой ветви отсутствует, так как там нет результатов.

Выражение типа **set** может быть использовано в качестве типа переменных в предикатной программе. В частности, множество может быть типом индексов массива и коллекции вершин графа.

3 Графы в предикатном программировании

Граф определяется множеством *вершин* и множеством ребер. *Ребро* – это пара вершин графа. Вершины x и y ребра (x, y) называются *смежными* (соседними). Ребро соединяет x и y и *инцидентно* вершинам x и y. Вершины x и y также называются *концевыми* ребра (x, y). *Степень вершины* – это количество инцидентных ей рёбер.

В языке P [1] любая вершина имеет тип **node**. Множество вершин графа имеет следующий предопределенный тип:

```
type nodeset = set(node);
```

Здесь **node** – базисный тип для множества вершин. Ребра графа имеют тип:

```
struct(source: node, target: node) .
```

Для ребра e его поля e.source и e.target являются концевыми, то есть e = (e.source, e.target). Будем также писать e.1 и e.2 и оперировать с ребром e как с кортежем типа (**node, node**). Тип множества ребер является предопределенным типом:

```
type edgeset = set(struct(source: node, target: node));
```

Вершины и ребра графа могут быть снабжены *атрибутами* (или метками в терминах теории графов), являющимися объектами произвольных типов. Атрибуты – реальное содержимое графа, тогда как вершины и ребра определяют лишь скелет графа. Атрибуты определяются в программе как массивы, индексами которых являются все вершины или все

ребра. Ниже приведен пример описания графа с набором вершин nodes и набором ребер edges, а также атрибутами: name – строковый атрибут вершин, size – целым атрибутом ребер.

```
nodeset nodes;
edgeset edges;
array(string, nodes) name;
array(int, edges) size;
```

По умолчанию тип **node** совпадает с типом **nat**. Если для конкретного приложения требуется другой базовый тип для набора вершин, отличный от **nat**, пользователю разрешается изменить тип вершины. Например, пользователь может ввести следующее описание:

```
type node = (int, 1..10);
```

Тип **set**(**node**), обозначаемый как nodeset, определяет наиболее общую и адекватную структуру для набора вершин. Любой подграф является частью структуры исходного графа. Извлечение подграфа не требует копирования из исходного графа. Решение использовать в качестве типа множества вершин диапазон вида m..n было бы хуже, так как извлечение подграфа и дальнейшая работа с ним как с независимым графом станет невозможным без копирования. Эффективная обработка на GPU графов, полученных в результате декомпозиции (без перемещения) исходного графа, вершины которого изначально расположены в диапазоне m..n, является новой задачей.

Для произвольного графа, определенного множеством вершин nodes и множеством ребер edges, необходимо дополнительное условие, определяемое предикатом is_graph: концевые вершины каждого ребра принадлежат множеству nodes. Кроме того, множества nodes и edges должны быть конечными.

```
formula is_graph(nodeset nodes, edgeset edges) =
        finite(nodes) & finite(edges) &
        ∀ node a, b. (a, b) ∈ edges => a ∈ nodes & b ∈ nodes;
```

Эффективность программ обработки графов достигается путем применения оптимизирующих трансформаций операций с множествами, учитывающих особенности конкретного приложения.

4 Программа определения степеней вершин

Технология предикатного программирования иллюстрируется на простой программе вычисления степеней вершин. Допустим, имеется неориентированный граф G = (V, E) с множеством вершин V и множеством ребер E. *Неориентированный граф* определяется следующей спецификацией:

```
formula unordered (nodeset nodes, edgeset edges) =
    is_graph(nodes, edges) & ∀ node a, b. (a, b)∈edges => a ≠ b & (b, a)∉edges;
```

Ребро (a, b) совпадает с ребром (b, a). Однако в множестве ребер edges указывается только одно. В частности, не допускаются ребра вида (a, a).

Программа Degree по множеству ребер E строит массив d. Элемент d[j] есть степень вершины j. Предикат degree определяет степень вершины v.

```
formula degree(node v, nodeset V, edgeset E : nat) = | { V x | (x, v) ∈ E ∨ (v, x) ∈ E } |;

formula postDeg (nodeset V, edgeset E, array(nat, V) d) = ∀ V v. d[v] = degree(v, V, E);

Degree(nodeset V, edgeset E : array(nat, V) d)
    pre   unordered(V, E)
    post  postDeg (V, E, d)
{ NodesDegree(V, E, E, for (V k) 0) }
```

Главная программа Degree вызывает рекурсивную программу NodesDegree. Ее третий параметр eds – это оставшееся необработанное подмножество ребер. Таким образом, мы можем выразить E как E = eds0 ∪ eds, где eds0 – уже обработанное подмножество ребер. Массив deg, четвертый параметр, получен в результате обработки подмножества eds0. Вначале deg инициализируется нулевым массивом.

```
NodesDegree(nodeset V, edgeset E, eds, array(nat, V) deg: array(nat, V) d)
    pre  unordered(V, E) & eds ⊆ E & ∀ v ∈ V. deg[v]= degree(v, V, E\eds)
    post  postDeg (V, E, d)
  { switch pick(eds){ case : deg
                      case (node, node) x, edgeset eds1:
                        (node i, node j) = x;
                        NodesDegree( V, E, eds1, deg with (i: deg[i]+1, j: deg[j]+1) )
  }}
```

Программа NodesDegree недетерминировано выбирает следующее ребро из множества eds с помощью гиперфункции pick. При $eds = \varnothing$ срабатывает первая ветвь гиперфункции, и результатом работы NodesDegree становится исходный массив deg. В случае $eds \neq \varnothing$ гиперфункция pick выбирает ребро (i, j) из множества eds. Остается набор ребер eds1 = eds \ {(i, j)}. Рекурсивный вызов NodesDegree применяется к модифицированному массиву deg для вершин i и j. В этой модификации элемент deg[i] заменяется на deg[i]+1, а deg[j] заменяется на deg[j]+1.

В дальнейшем ограничитель **switch** будет отсутствовать. Вместо переменной x в **case** части вставляются компоненты i и j из определения компонент кортежа (**node** i, **node** j) = x, которое далее становится ненужным. Наконец, параметры V и E сделаем глобальными. Данные изменения дают следующую программу.

```
nodeset V;
edgeset E;
Degree( : array(nat, V) d)
    pre   unordered(V, E)
    post  postDeg (V, E, d)
{ NodesDegree( E, for (V k) 0 ) }

NodesDegree(edgeset eds, array(nat, V) deg: array(nat, V) d)
    pre  unordered(V, E) & ∀ v ∈ V. deg[v] = degree(v, V, E \ eds)
    post  postDeg (V, E, d)
  { pick(eds){  case : deg
                case (node i, node j), edgeset eds1:
                        NodesDegree(eds1, deg with (i: deg[i]+1, j: deg[j]+1) )
  }}
```

Результатом оптимизирующей трансформация является эффективная императивная программа, представленная ниже. Множество вершин V кодируется диапазоном 0..Vm-1, где Vm – число вершин. Множество ребер задается массивами Es и Et. Для k=0..En-1 кортеж (Es[k], Et[k]) является ребром из множества E.

```
array(nat, nat) *Degree(nat Vm, En, array(nat, nat) Es, Et)
{ array(nat, nat) d;
  for (nat j=0; j<Vm; j++) d[j] = 0;
  for (nat k =0; k < En; k++) { d[Es[k]]++; d[Et[k]]++ };
  return &d
}
```

5 Верификация программы определения степеней вершин

Система верификации Why3 [13] - это платформа для дедуктивной верификации программ на функциональном языке WhyML. Функция на языке WhyML снабжается спецификацией в виде предусловия и постусловия. Инструмент Why3 генерирует по программе и ее спецификации формулы корректности на основе классической логики Хоара [25]. Для доказательства формул корректности инструмент Why3 может использовать большое число внешних систем автоматического доказательства, как автоматических, в том числе SMT-решателей, так и интерактивных, в которых пользователь строит доказательство, применяя команды инструмента.

Программа Degree вместе с ее спецификацией была преобразована на язык WhyML. Верификация программы проводилась в системе Why3 версии 1.5.1 с автоматическими SMT-решателями Alt-Ergo 2.4.2, CVC3 2.4.1, CVC4 1.8, CVC5 1.0.2, Vampire 4.7, Z3 4.11.2.

Сгенерированные формулы корректности были доказаны применением набора перечисленных инструментов.

Программа Degree достаточно проста в сравнении с другими известными алгоритмами на графах. Тем не менее, обнаружилось, что она верифицируется впервые в мире. Нигде ни в одной системе дедуктивной верификации не обнаружено других попыток верифицировать программу Degree.

В спецификации такой простой программы Degree обнаружено четыре ошибки. В формуле is_graph отсутствовало условие конечности множества вершин и ребер. Вторая ошибка: пропущено условие $eds \subseteq E$ в программе NodesDegree. Третья ошибка: в неориентированном графе отсутствовало условие недопустимости дуг вида (a, a). На практике, дуги такого вида иногда допускаются с условием, что их надо считать дважды. Четвертая ошибка: при наличии дуги (a, b) недопустимо присутствие дуги (b, a).

Верификация в системе Why3 оказалась сложной и трудоемкой. Доказательство в Why3 хорошо поддерживается стандартными библиотеками теорий. Однако в используемой библиотеке FSet для конечных множеств не оказалось лемм для доказательства свойств конструкций вида $| \{ x \in V \mid R(x, y) \} |$ для некоторого произвольного предиката R. Более того, обнаружилось, что подобных лемм нет и в других известных системах верификации.

Доказательство постусловия для программы NodesDegree было сведено к следующей лемме:

lemma $\forall R, S. \forall V x,y.$ **not** $R(x, y) \& S(x, y) \Rightarrow$

$| \{ V x \mid R(x, y) \vee S(x, y) \} | = | \{ V x \mid R(x, y) \} | + | \{ V x \mid S(x, y) \} |$

Система Why3 базируется на логике первого порядка. Поэтому произвольные предикаты R и S представимы там лишь как параметры клонируемого модуля.

6 Заключение

Представлен новый подход к разработке эффективных и корректных алгоритмов обработки графов в рамках предикатного программирования. Вершины и ребра графа представляются как значения типа set в языке P [1]. Предикатная программа обработки графов значительно проще, чем аналогичная программа на других языках программирования, и поэтому больше подходит для проведения дедуктивной верификации.

Метод предикатного программирования для разработки и верификации программ на графах апробирован также на следующих алгоритмах: построения списков смежности и матрицы смежности, нахождения списков соседей, упорядоченных по убыванию степеней вершин, трех алгоритмов построения максимальных клик, построения изоморфного подграфа, алгоритма Косараджу нахождения сильно связных компонент графа.

То, что на простом алгоритме определения степеней обнаружено четыре ошибки, является показательным свидетельством, что дедуктивная верификация графовых программ необходима, особенно для критических приложений. При верификации всплывают неожиданные особенности, ускользающие при традиционном математическом анализе. Например, обнаружено, что пустое множество вершин является максимальной кликой.

Начата работа по построению библиотек теорий с набором лемм для разнообразных алгоритмов на графах. В дальнейшем планируется перенести в систему Why3 достаточно полные библиотеки [22, 24] для систем автоматического доказательства PVS [5] и Isabelle/HOL [18]. Многочисленные коллекции теорий графов существуют и для других систем доказательства.

Литература

[1] Карнаухов Н.С., Першин Д.Ю., Шелехов В.И. Язык предикатного программирования P. Версия 0.14. ИСИ СО РАН, Новосибирск, 2018. 45с. URL: https://persons.iis.nsk.su/files/persons/pages/plang14.pdf

[2] Каблуков И.В., Шелехов В.И. Реализация оптимизирующих трансформаций в системе предикатного программирования. *Системная информатика*, № 11. Новосибирск, 2017. С. 21-48. Электрон. журн. 2018. http://persons.iis.nsk.su/files/persons/pages/opttransform4.pdf

[3] J. Chamberlin, M. Zalewksi, S. McMillan, A. Lumsdaine, “PyGB: GraphBLAS DSL in Python with dynamic compilation into efficient C++,” in Graph Algorithms Building Blocks (GABB) Workshop at IEEE Intl. Parallel and Distributed Processing Symposium, 2018, pp. 310-319.

[4] T. Mattsonz, T.A. Davis, M. Kumar, A. Buluc, S.McMillanx, J. Moreira, C. Yang, “LAGraph: A Community Effort to Collect Graph Algorithms Built on Top of the GraphBLAS,” in 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2019, pp. 276-284.

[5] S. Owre, J. M. Rushby, N. Shankar, **“**PVS: specification and verification system,” in CADE-11: Automated Deduction. LNCS, v. 607. 1992, pp. 748-752.

[6] Тумуров Э.Г., Шелехов В.И. Трансформация, спецификация и верификация программы вычисления числа элементов множества, представленного в виде битовой шкалы. — Системная информатика, № 16. — Новосибирск, 2020. — С. 103-136. URL: https://www.system-informatics.ru/files/article/tumurovshelehov.pdf

[7] Шелехов В.И. Верификация предикатной программы бинарного поиска объекта произвольного типа // Системная информатика, № 15. Новосибирск, 2019. С. 45-64. URL: http://persons.iis.nsk.su/files/persons/pages/fsearch2.pdf

[8] Шелехов В.И. Разработка и верификация алгоритмов пирамидальной сортировки в технологии предикатного программирования. Препринт 164, ИСИ СО РАН, Новосибирск, 2012. — 30с.

[9] Шелехов В.И. Верификация программы преобразования строки в целое число // Системная информатика, № 17. — Новосибирск, 2020. — С. 43-90. https://persons.iis.nsk.su/files/persons/pages/kstr2.pdf

[10] J. Kepner, etc. “Mathematical foundations of the GraphBLAS,” in 2016 IEEE High Performance Extreme Computing Conference (HPEC). 2016, pp. 1–9.

[11] Shelekhov V. Applying Program Transformations for Deductive Verification of the List Reverse Program. Software Engineering, vol. 12, no. 3, 2021 pp. 127-139. (In Russian). https://persons.iis.nsk.su/files/persons/pages/listspi.pdf

[12] R. Chen, J. Levy, Formal proofs of two algorithms for strongly connected components in graphs. 2016, 19p. URL: https://hal.inria.fr/hal-01422216

[13] Why 3. Where Programs Meet Provers. URL: http://why3.lri.fr

[14] Шелехов В.И. Верификация предикатной программы пирамидальной сортировки с применением обратных трансформаций // Системная информатика, № 16. Новосибирск, 2020. С. 75-102. URL: https://persons.iis.nsk.su/files/persons/pages/sort9.pdf

[15] J.-R. Abrial. Modeling in Event-B: System and Software Engineering. Cambridge University Press, 2010. 586p.

[16] L. Lamport. Specifying systems: The TLA+ language and tools for hardware and software engineers. Addison-Wesley, 2021. 382p.

[17] Шелехов В.И. Дедуктивная верификация программы конкатенации строк с применением обратной трансформации // Знания-Онтологии-Теории (ЗОНТ-19). Новосибирск, ИМ СО РАН, 2019. С. 369-378. URL: http://persons.iis.nsk.su/files/persons/pages/logcflc1.pdf

[18] T. Nipkow, L.C. Paulson, M.Wenzel, “Isabelle/HOL — A proof assistant for higher-order logic,” Springer, 2002, 211p.

[19] GraphBLAS Template Library (GBTL), https://github.com/cmu-sei/ gbtl.

[20] T. A. Davis, “Algorithm 9xx: SuiteSparse:GraphBLAS: graph algorithms in the language of sparse linear algebra,” ACM Trans. Math. Softw. 2018, 24p.

[21] C. Yang, A. Buluç, J.D. Owens, “GraphBLAST: a highPerformance linear algebra-based graph framework on the GPU,” in CoRR, 2019. Vol. abs/1908.01407. 1908.01407.

[22] L. Noschinski, “A graph library for Isabelle”, Math.Comput.Sci. 9. 2015, pp. 23–39.

[23] A. Buluc, T. Mattson, S. McMillan, J. Moreira, C. Yang, “Design of the GraphBLAS API for C,” in 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017, pp. 643-652.

[24] R.W. Butler, J.A. Sjogren, “A PVS graph theory library,” Tech. rep., NASA Langley. 1998.

[25] Hoare C. A. R. An axiomatic basis for computer programming // Communications of the ACM. 1969. Vol. 12 (10). P. 576–585.

Знания-Онтологии-Теории (ЗОНТ-2023)

Применение генеративно-состязательных нейронных сетей для порождения предложений, связанных отношением логической выводимости

Шишкин А.А., Трегубов А.С.

Факультет Информационных Технологий, Новосибирский государственный университет, ул. Пирогова, д. 1, г. Новосибирск, 630090, Россия.

shaa6661@gmail.com, tematre@gmail.com

Аннотация: Данная работа посвящена вопросу генерации предложений на естественном языке, связанных отношениями логического следствия и противоречия. Цель работы - создать генеративно-состязательную нейронную сеть, способную порождать тексты, удовлетворяющие этому критерию. Предложенный метод предлагается использовать совместно с существующими методами определения отношений логической выводимости, поскольку он позволяет увеличить точность таких методов, и создавать обучающие выборки данных, специфичные для конкретной предметной области, что само по себе является сложной и кропотливой задачей. Использование предложенного метода позволит автоматизировать данный процесс, чем обусловлена актуальность и практическая ценность исследования. При этом предложенный подход не требует построения сложных наборов данных. Представленная модель генеративной-состязательной нейронной сети с использованием архитектуры Transformer показала достойный результаты при оценке сгенерированных текстов. В дальнейшем планируется развитие данной работы за счет модификации архитектуры и использования других базовых языковых моделей.

Ключевые слова: машинное обучение, нейронные сети, генеративно-состязательные сети, глубокое обучение, обработка текстов, анализ противоречий, анализ текстов, компьютерная лингвистика, анализ данных, искусственный интеллект.

1. Введение

Задача определения наличия отношения логической выводимости между двумя текстами на данный момент активно исследуется и развивается. Все дело в большом количестве возможных сценариев ее применения. Среди них приложения, позволяющие автоматизировать анализ корректности ответов на вопросы или выявление противоречий в документах.

На данный момент для решения задачи используются нейронные сети, обучающиеся на фиксированном наборе данных. Такой подход требует большого количества размеченных обучающих данных для первичной настройки нейронной сети, и, при этом, для корректной настройки сети данные должны принадлежать специфичной предметной области, в которой предполагается ее работа.

Для получения достаточной точности предсказаний без наблюдателя в модулях обработки непрерывных данных используются генеративно-состязательные нейронные сети. Это достаточно новая архитектура, которая уже показала высокие результаты в обработке изображений. Однако, для решения задачи порождения текстов, требуется предварительная адаптация данной архитектуры.

Целью данной работы является разработка генеративно-состязательной нейронной сети для построения логически связных предложений на естественном языке. Для достижения цели были поставлены следующие задачи:

- Разработать архитектуру нейронной сети;
- Разработать генератор предложений на естественном языке;
- Разработать дискриминатор;
- Провести тестирование генеративно-состязательной сети.

Далее в тексте каждая задача будет рассмотрена отдельно.

2. Обзор литературы

Существуют различные подходы к построению нейронных сетей, выполняющих те или иные задачи.

Одним из наиболее каноничных способов является нейронная сеть, обучаемая на фиксированном и заранее определенном массиве данных. Такой подход хорошо зарекомендовал себя и решил большое количество задач. Тем не менее, нейронная сеть, построенная таким образом, сильно привязана к предметной области данных, на которых она настраивается, поэтому для каждой новой предметной области необходимо собирать новые массивы данных. При этом, из-за сложности формулирования требований к набору данных и его объему, часто сбор обучающего набора данных сам становится отдельной задачей для исследователей [1, 2].

Генеративно-состязательная сеть – модель машинного обучения, способная имитировать заданное распределение данных [3]. Применимо к изображениям, генеративно-состязательные сети позволяют генерировать изображения, статистически неотличимые от настоящих [4]. Данная модель состоит из двух нейронных сетей: сеть-генератор и сеть-дискриминатор.

Слово «состязательная» в названии следует из того, что данная модель подразумевает работу компонент «друг против друга»: генератор обучается генерировать данные таким образом, чтобы предсказание дискриминатора оказалось ошибочным, а дискриминатор обучается давать безошибочные предсказания относительно данных генератора.

Таким образом, в ходе работы модели сеть-генератор обучается обманывать сеть-дискриминатор и, соответственно, учится создавать все более реалистичные изображения: поддельные изображения, неотличимые от настоящих в той мере, на какую способна сеть-дискриминатор (рис. 1). Сеть-дискриминатор, в свою очередь, постоянно адаптируется к увеличивающейся способности сети-генератора и устанавливает все более высокую планку реализма для генерируемых изображений. По окончании обучения генератор способен превратить любую точку из своего входного пространства в правдоподобное изображение.

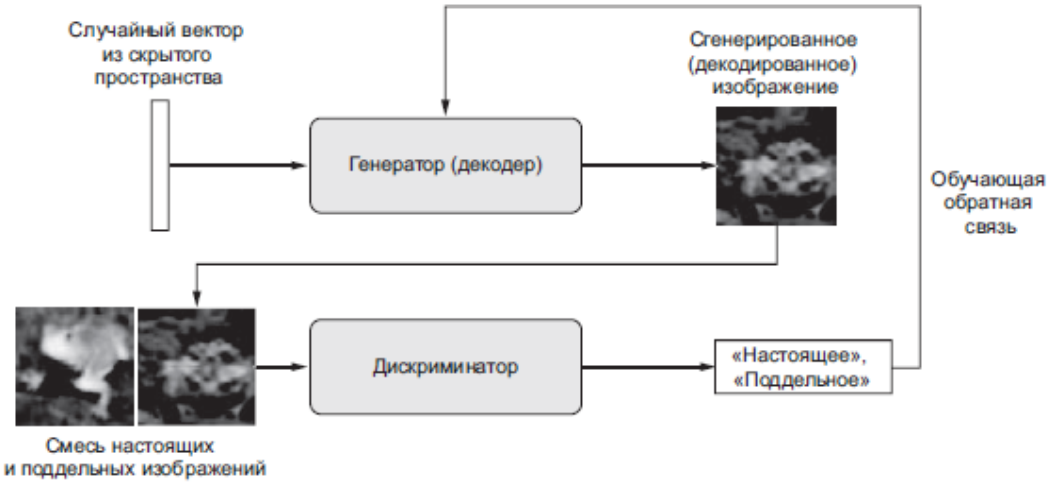


Рис. 1. Схема генеративно-состязательной сети, работающей с изображениями.

Для решения задачи генерации предложений активно используется архитектура Transformer [5]. Она заключается в разделении нейронной сети на часть кодирования предложения и декодирования. При этом, обе части, используя механизм Attention, могут на каждой итерации для каждого слова независимо определять, какая информация от остальных слов им наиболее полезна. Предлагается использовать данную архитектуру для получения векторных представлений слов.

Наиболее популярным и общепринятым на сегодня массивом данных, используемым в области определения логических связей между предложениями на естественном языке, является Stanford Natural Language Inference (SNLI) Corpus [2]. Он состоит из предложений на естественном языке, связанных 3 видами отношения логической выводимости: логическое следствие, противоречие и неопределенность. Существуют и альтернативные наборы данных, используемые для решения задачи определения логических связей между предложениями, такие как MNLI [2]. Предлагается использовать данный массив для первичной инициализации весов.

3. Предлагаемое решение

В рамках работы, с учетом всех требований, предлагается построить 2 нейронных сети: сеть-генератор и сеть-дискриминатор. При этом использование генеративно-состязательной нейронной сети позволит получить генератор логически связных предложений, по качеству не уступающих примерам из собранных наборов данных. Далее, с использованием генератора, будет строиться динамический набор обучающих данных, на котором будет настраиваться сеть-дискриминатор.

Модель генеративно-состязательной сети представлена на рис. 2: для обработки логических следствий и логических противоречий предлагается использовать независимые модули как внутри генератора, так и внутри дискриминатора.

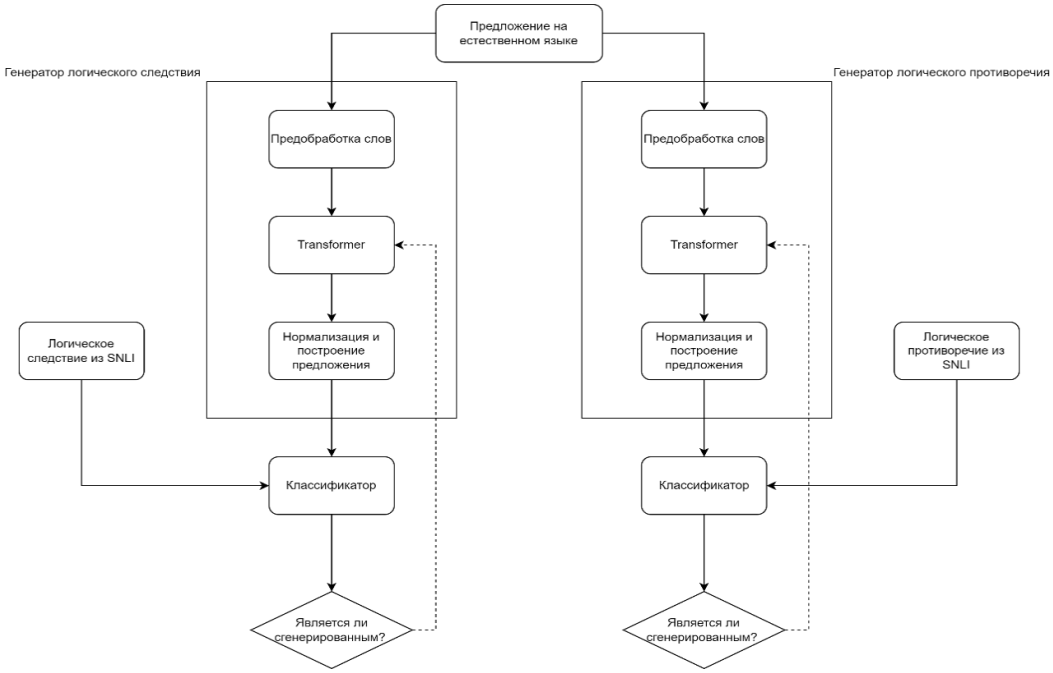


Рис. 2. Схема построенной генеративно-состязательной сети.

Поскольку нейронная сеть занимается обобщением данных, а в случаях логических следствий и противоречий между предложениями данные имеют схожую структуру, а отличаются только разными правилами обобщения, генератор был разбит на двух отдельных независимых модуля, занимающихся построением предложения-противоречия и предложения-следствия. При этом каждый модуль состоял из трех частей: предобработка слов, Transformer и построение предложения.

Часть предобработки слов отвечает за предобработку предложений и их перевод в числовой вектор. Для этого предложение токенизируется, каждый токен заменяется на свой уникальный индекс, затем каждый индекс переводится в уникальный вектор из многомерного векторного пространства - эмбеддинг слова. После этого набор индексов преобразуется таким образом, чтобы учитывался порядок каждого слова в предложении, используя специальную добавочную матрицу [5]

$$p_{(t,\,2i)} = \sin\left(\frac{t}{10000^{\frac{2i}{d}}}\right) (1)$$

$$p_{(t,\,2i+1)} = \cos\left(\frac{t}{10000^{\frac{2i}{d}}}\right) (2)$$

, где t - позиция слова в предложении, d - число позиций внутри эмбеддинга каждого слова, i - позиция внутри эмбеддинга слова.

Следующая часть реализует архитектуру Transformer [5], и отвечает за перевод эмбеддинга предложения в набор векторов, соответствующих вероятностям каждого индекса слова в каждой позиции сгенерированного предложения (рис. 3).

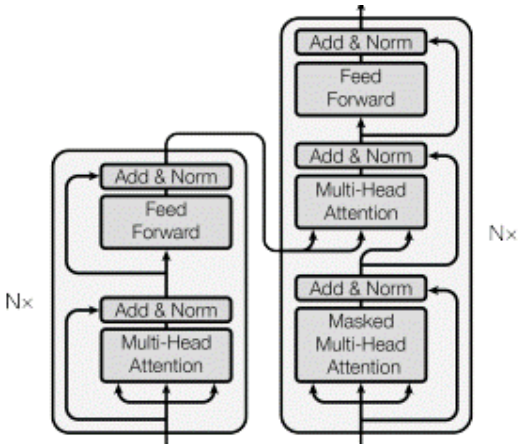


Рис. 3. Схема архитектуры Transformer.

Часть, отвечающая за построение предложение, по результату работы предыдущей части определяет индекс, с наибольшей вероятностью встречающееся в каждой позиции выходного предложения. После этого, из последовательности убираются служебные индексы, и по остальным индексам составляется результирующее предложение.

Далее следует модуль, реализующий логику дискриминатора. В рамках него построена модель, способная классифицировать предложения как настоящие, то есть взятые из набора данных, или поддельные, то есть построенные генератором.

Архитектура дискриминатора представлена на рис. 4: модуль имеет блок векторизации предложений, блок, реализующий архитектуру Transformer, и блок классификации. В качестве аргументов модуль приминает предложение, взятое из набора данных или из генератора. Возвращаемым значением модуля является булево значение, свидетельствующее о принадлежности предложения к классу сгенерированных или настоящих.

Для настройки генераторов логических следствий и противоречий предлагается использовать независимые дискриминаторы, построенные по общей архитектуре, но использующие только следствия или противоречия, соответственно.

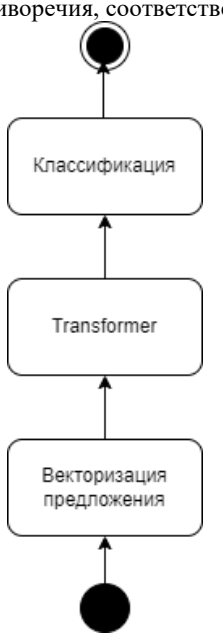


Рис. 4. Схема архитектуры дискриминатора.

На основании построенной сети был сформирован алгоритм для ее обучения. Алгоритм обучения включает в себя переменное число эпох, каждая из которых состоит из следующих шагов:

1. Подготовка необходимых данных. Шаг заключается в формировании выборок логических следствий и противоречий из массива данных SNLI, а также выборок со

следствиями и противоречиями, построенными генератором для тех же исходных предложений.
2. Обучение дискриминатора на данных из обоих выборок, взятых из SNLI.
3. Обучение дискриминатора на данных из обоих выборок, построенных генератором.
4. Обучение генератора логического следствия на соответствующей выборке из SNLI с использованием обратной связи дискриминатора.
5. Обучение генератора логического противоречия на соответствующей выборке из SNLI с использованием обратной связи дискриминатора.

4. Результаты

На первом этапе было выполнено предобучение генератора логических следствий и противоречий на данных из SNLI (рис. 5). В результате функция потерь обоих частей генератора на валидационной выборке уменьшалась, то есть построенные нейронные сети способны обобщать рассматриваемую предметную область, и подходят для решения задачи.

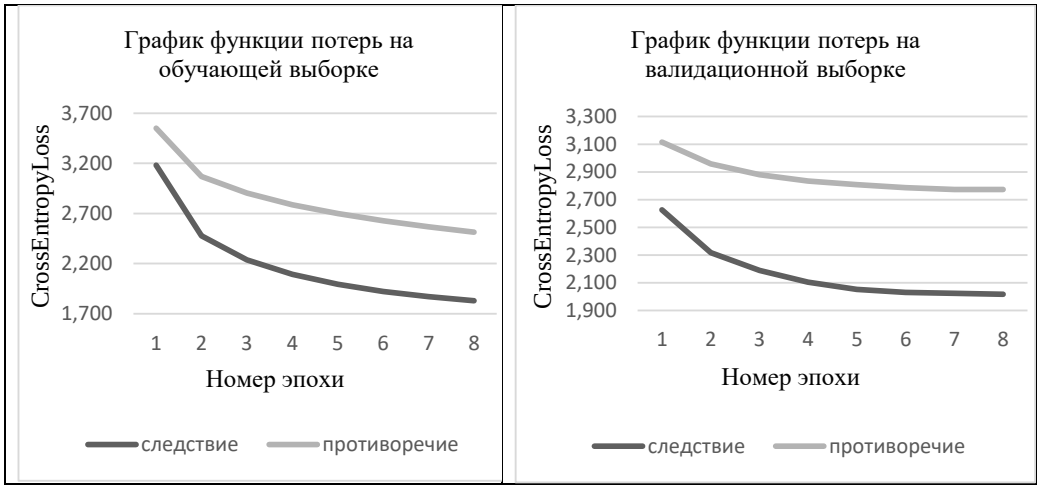


Рис. 5. Графики функции потерь CrossEntropyLoss при обучении и валидации генератора

После предварительной настройки генератора на данных SNLI, было проведено обучение построенной нейронной сети по предложенному алгоритму. В результате обучения реализованных модулей в составе генеративно-состязательной нейронной сети, функция потерь генератора незначительно снизилась. Данное наблюдение совпадает с теоретически ожидаемым результатом.

В результате совместного обучения, значения функции потерь дискриминатора также уменьшались (рис. 6), что свидетельствует о том, что построенный дискриминатор подходит для использования в рамках задачи классификации предложений на настоящие и поддельные, и так же совпадает с ожидаемым результатом.

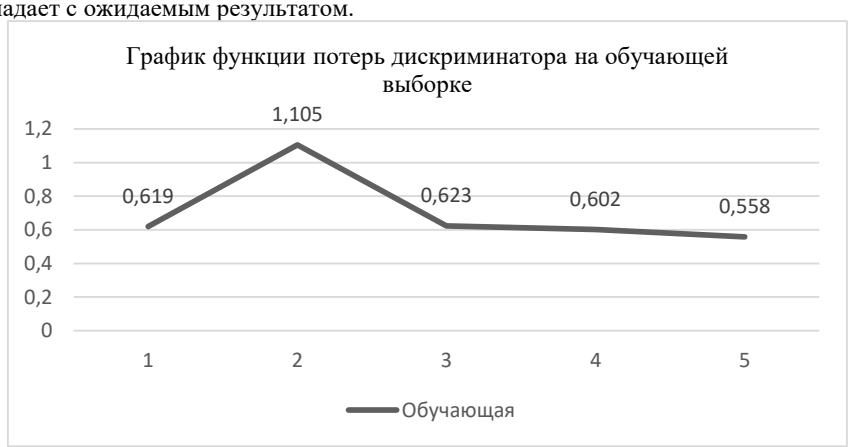


Рис. 6. График функции потерь CrossEntropyLoss при обучении дискриминатора в составе генеративно-состязательной сети

Кроме того, была выполнена проверка качества работы генератора путем сравнения сгенерированных следствий и противоречий с данными из тестового набора данных SNLI, который не использовался для обучения и валидации сети. В качестве исходных предложений были взяты “A little boy in a gray and white striped sweater and tan pants is playing on a piece of playground equipment.” и “Male in a blue jacket decides to lay in the grass.”. В результате сравнения был сделан вывод, что сгенерированные предложения находятся в требуемых отношениях логической выводимости с исходными, а также достаточно вариативны.

Таблица 1. Результаты работы построенного генератора с примерами SNLI.

Источник	Следствие	Противоречие
SNLI	A boy is on a playground.	The boy is sitting on the school bus on his way home.
Генератор	A little boy is playing on a playground.	A boy is playing in the sand.
SNLI	The guy wearing a blue jacket is laying on the green grass.	The guy in yellow is rolling on the grass.
Генератор	A man is laying in the grass.	A man is sitting on a couch.

5. Заключение

Данная работа посвящена вопросу генерации текста на естественном языке с помощью генеративно-состязательной нейронной сети. Задача порождения текстов с заданной семантикой является достаточно сложной задачей, представляющей высокую практическую ценность. Поскольку созданные данные, удовлетворяющие заданному распределению, могут в дальнейшем быть использованы как входные данные для различных задач.

Генеративно-состязательные нейронные сети являются достаточно новой архитектурой чаще всего применяемой для задачи генерации изображений. Но в рамках данной работы были предложены методы адаптации данной архитектуры для задачи генерации текстов, что представляет научную новизну данной работы. В частности, предлагается использование нейронных сетей архитектуры Transformer для получений векторного представления текста. А также SNLI для первичной инициализации весов, что позволило значительно снизить время и сходимость процесса обучения.

Целью данной работы является разработка генеративно-состязательной нейронной сети для построения логически связных предложений на естественном языке.

В рамках работы были выполнены следующие задачи для достижения поставленной цели:

- Разработать архитектуру нейронной сети;
- Разработать генератор предложений на естественном языке;
- Разработать дискриминатор;
- Провести тестирование генеративно-состязательной сети.

Представленная модель генеративной-состязательной нейронной сети с использованием архитектуры Transformer показала достойные результаты при оценке сгенерированных текстов. В дальнейшем планируется развитие данной работы за счет модификации архитектуры и использования других базовых языковых моделей.

Литература

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[2] P. Jain, B. Kulis, I. Dhillon, and K. Grauman. Online metric learning and fast similarity search. In Advances in Neural Information Processing Systems, volume 22, 2008a

[3] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2008b.

[4] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In Proc. of ICML, New York, NY, USA, 2009. 2

[5] H. Kanayama and T. Nasukawa. Fully Automatic Lexicon Expansion for Domain-Oriented Sentiment Analysis. EMNLP’06, 2006.

Прототипирование Базы Знаний для Определения Цитологических Изменений Эндоцервикса в Гинекологии

Б.О. Щеглов[1], Г.В. Рева[2], С.Н. Щеглова[3]

[1]*Медицинский центр Дальневосточного федерального университета, ул. Малый Аякс 10, о. Русский, Приморский край, г. Владивосток, 690920, Россия*
[2]*Школа медицины Дальневосточного федерального университета, ул. Малый Аякс 10, о. Русский, Приморский край, г. Владивосток, 690920, Россия*

[3]*Северо-Восточный государственный университет, ул. Портовая 13, Магаданская обл., г. Магадан, 685007, Россия*

b.shcheglov@mail.ru, reva.gv@dvfu.ru, schegloff@mail.ru

Аннотация. *Рак шейки матки - это одно из самых распространенных онкологических заболеваний среди женщин во всем мире, и частота заболеваемости им продолжает расти каждый год. Однако, до сих пор основным методом диагностики РШМ на протяжении более полувека остается Пап-тест, который имеет субъективный характер и зависит от опыта и состояния психоневрологической системы врача. В связи с этим, в данной работе представлена концепция для разработки предметной области лабораторных исследований по гинекологической цитологии в области цифровой микроскопии на основе системы Bethesda классификации клеточных изменений от 2014 года. Это новый подход, который позволяет проводить объективную диагностику и классифицировать клеточные изменения в зоне трансформации эндоцервикса в гинекологии. В результате проведенного обзора была создана база знаний для классификатора, состоящая из 11 гипотез и 7 атрибутов, характеризующих их. Также был разработан прототип базы знаний для классификатора гипотез на платформе IACPaaS ИАПУ г. Владивостока. Использование данного технологического решения позволит значительно повысить точность путем объективизации диагностики РШМ и улучшить качество проведения диагностических мероприятий у женщин.*

Ключевые слова: база знаний, РШМ, гинекология, система Bethesda

1 Введение

На сегодняшний день цифровизация является основным трендом в развитии современного здравоохранения. Она предоставляет возможность медицинским специалистам применять базы знаний для более точной диагностики и верификации различных заболеваний, в том числе и заболеваний репродуктивного тракта женщин.

Рак шейки матки (РШМ) - это одно из самых распространенных онкологических заболеваний в женской популяции. По данным Всемирной организации здравоохранения, ежегодно в мире выявляется около 570 000 новых случаев РШМ, при этом около 90% из них происходят в развивающихся странах [Giuliani et al 2020; Пивазян и др., 2022; Кравец и др., 2020]. Большинство смертей от РШМ также приходится на эти страны.

В России количество новых случаев заболевания также остается высоким и имеет тенденцию к прогрессированию. Согласно данным Федеральной службы по надзору в сфере защиты прав потребителей и благополучия человека, за последние годы в среднем было выявлено более 14 000 случаев заболеваемости РШМ, что составило около 2,7% всех новых диагнозов онкологических заболеваний у женщин в стране [Eun et al 2020; Smith et al 2018; Rezende et al 2021].

Основными причинами развития РШМ являются вирус папилломы человека (ВПЧ), который передается половым путем, а также поздний период деторождения у женщин, в связи с их возрастающей вовлеченностью в социально-экономические отношения в обществе [Chrysostomou et al 2018]. Вакцинация против ВПЧ является одной из главных мер по профилактике развития РШМ. Также очень важную роль в предотвращении заболевания играют регулярные скрининговые исследования, которые позволяют выявлять изменения в состоянии шейки матки на ранних стадиях. Благодаря принципам обработки больших массивов данных могут быть использованы базы знаний для анализа результатов скрининга и определения наличия РШМ, что позволит более эффективно бороться с этим заболеванием.

В гинекологии проведение Пап-теста является одним из важнейших методов диагностики РШМ [Staats et al 2019]. Но в последнее время на смену микроскопии при анализе клеточных изменений зоны трансформации приходят компьютерные технологии и использование баз знаний для классификации полученных результатов.

Существует достаточно большое количество проблем с диагностированием в гинекологической цитологии. Одна из главных проблем заключается в том, что существует множество различных шкал и критериев, которые могут быть использованы для диагностики клеточных изменений. Это может привести к несогласованности между лабораториями и различными врачами в вопросах их интерпретации и, соответственно, расходящейся оценке риска развития РШМ.

В настоящее время основной системой классификации остается Bethesda, которая была обновлена в 2014 году [Banerjee et al 2022; Swailes et al 2019]. Эта система классификации используется во многих странах, и она обеспечивает общий медико-диагностический язык для описания клеточных изменений и трансформаций.

Одной из наиболее проблемных областей в цитологической диагностике по данной системе является определение и клиническая интерпретация ASC-US (атипичные клетки неясной природы). ASC-US - это результат, когда клетки, полученные в ходе исследования шейки матки, выглядят необычно, но недостаточно, чтобы считаться по-настоящему аномальными или злокачественными и быть связаны напрямую с клиникой [Eun et al 2020; Rezende et al 2021].

Кроме того, существует множество факторов, которые могут влиять на результаты диагностики, такие как маточный цикл пациента во время проведения скрининга, его уровень гормонов в крови, наличия персистирующих инфекций и многих других. Также сложность диагностики может быть связана с недостаточной квалификацией диагноста или использованием устаревших методик применения традиционной микроскопии. Чтобы уменьшить субъективность и повысить точность диагностики, важно использовать различные методы и инструменты для детерминации объективных параметров, а также проводить перепроверку результатов с применением информационных технологий.

Для решения задачи субъективного подхода к оценке и интерпретации лабораторно-диагностических данных цитологии зоны эндоцервикса в гинекологии могут быть применены базы знаний как альтернативный метод перепроверки результатов анализа и как система поддержки принятия врачебных решений (СППВР) по типу «Третье мнение».

Это даст возможность обеспечить формализацию и объективизацию различных атрибутов клеток для ранней диагностики РШМ и других заболеваний по типу «Умного справочника». Это связано с тем, что базы знаний на основе онтологии позволяют компьютерной системе выполнять логические выводы, сопоставляя информацию из базы знаний по атрибутам и признакам, введенных экспертом, и предоставлять гипотезы на основе определенных правил формальной логики.

Для применения базы знаний в клеточной гинекологии необходимо данные о клеточном составе препарата собирать при помощи различных методов, таких как цифровая микроскопия, объективизация и вычленение признаков с изображения клеток (размеры клеток, ядра и ядерно-цитоплазматическое отношение (ЯЦО)).

После проведения микроскопического исследования или установки окуляра над клеткой, когда врач ее изучает, информация о цитологической структуре препарата может быть использована для создания логических правил, которые позволяют системе прогнозировать риск развития РШМ на основе данных об изменениях в зонах эндоцервикса. Вводя данные, полученные в ходе цифровой обработки изображения, можно подтвердить или опровергнуть гипотезу о классе рассматриваемого клеточного изменения.

В данной работе будет рассмотрено применение базы знаний для классификации клеточных изменений в гинекологии на базе сервиса института автоматики и процессов управления (ИАПУ) г. Владивостока клеточных изменений зоны трансформации в гинекологии.

Цель исследования. Разработать концепцию и прототип базы знаний для объективной классификации клеточных изменений зоны трансформации эндоцервикса для клинической лабораторной диагностики по системе Bethesda от 2014г.

Задачи исследования:

- Изучить общемировой опыт медицинской диагностики клеточных изменений по Bethesda 2014 г.
- Выбрать классы клеточных изменений, ассоциированных с определенными значениями признаков.
- Реализовать логику принятия решений гипотез о принадлежности клеток к какому-либо классу (по определенным признакам и их атрибутам).
- Реализовать интерпретируемую часть знаний для каждой каждой из гипотез о классе.

2 Материалы и методы

Исходя из цели исследования был проведен краткий обзор литературы для создания гипотез и характеризующих их атрибутов в отечественных и международных базах данных Elibrary, Cyberleninka, PubMed, Cochrane, Scopus и Web of Science. Критерием отбора были актуальность литературы (не старше 5 лет), а также описание и применение в исследованиях признаков, которые могут объективно и в формализованной форме описывать различные типы клеток по системе Bethesda от 2014г.

В качестве ключевых слов для поиска использовались следующие: система классификации Bethesda, тест Папаниколау (Pap-test), РШМ (cervical cancer), объективные параметры (objective parameters), такие как размер клетки (cell size), размер ядра (nuclei size) и ядерно-цитоплазматическое отношение (nuclear-cytoplasmic ratio). Эти поисковые термины были выбраны потому, что они считаются важными индикаторами для выявления и классификации различных типов клеток, связанных с раком шейки матки [Giuliani et al 2020; Пивазян и др., 2022; Кравец и др., 2020; Rezende et al 2021; Alrajjal et al 2021; Tiwana et al 2022].

Обзор литературы выявил несколько существующих методов классификации, используемых при анализе клеток РШМ, включая систему Всемирной организации здравоохранения (ВОЗ), шкалы дисплазий CIN и классификации по Папаниколау, которые сообщены между собой и системой классификацией Bethesda и классифицируют клеточные аномалии по конкретным диагностическим категориям на основе их морфологических характеристик.

Часть из этих исследований были сосредоточены на разработке автоматизированных алгоритмов анализа изображений и вопросах их сегментации для классификации клеточных изменений зоны трансформации на основе объективных параметров [Smith et al 2018; Rezende et al 2021; Alrajjal et al 2021; Tiwana et al 2022].

База знаний и детерминированные условия для реализации гипотез к ней были реализованы на основе сервиса Института автоматики и процессов управления. Сервис классификации также находится на веб-ресурсе данного учреждения.

3 Результаты

В ходе данного исследования была реализована база знаний для классификации клеточных изменений в гинекологии и определено 11 основных гипотез от 4-7 атрибутов, характеризующие результат цитологического исследования, полученного в ходе Пап-теста (Таб. 1). В ходе литературного обзора были определены следующие гипотезы, связанные с результатом исследования по системе классификации клеточных изменений в зоне трансформации по Bethesda от 2014г. [Rezende et al 2021; Banerjee et al 2022; Alrajjal et al 2021; Bispo Pereira et al 2023]:

- NILM - патологические внутриэпителиальные изменения в клетках отсутствуют. При этом данному результату анализа соответствуют следующие 6 видов

результатов диагностики клеточных изменений и соответствующих им гипотез: поверхностный плоский эпителий, промежуточный плоский эпителий, плоский метаплазированный эпителий, плоский парабазальный эпителий, эндоцервикальные железистые клетки и эндометриальные железистые клетки.

- ASC-US - атипичные клетки плоского эпителия неопределенного значения, которые делятся на категории поражений низкой (ASC-L) и высокой (ASC-H) степеней внутриэпителиального поражения клеток.
- LSIL - плоскоклеточное интраэпителиальное поражение низкой степени, с низким и средним риском развития РШМ.
- HSIL - плоскоклеточное интраэпителиальное поражение высокой степени, с средним и высоким риском развития РШМ.
- AIS - аденокарцинома на месте (in situ), характеризующая предраковые изменения канала шейки матки.

Таблица 1. Характеристика используемых атрибутов в разработанной базе знаний

Атрибут	***Тип***	***Краткое описание***
Размер клетки	Числовой, диапазон значений	Диапазон размеров клетки в мкм
Размер ядра	Числовой, диапазон значений	Диапазон размеров ядра в мкм
ЯЦО	Числовой, диапазон значений	Диапазон ЯЦО в у.е. или %
Характеристика клетки	Строковый	Качественное описание признаков клетки
Характеристика ядра	Строковый	Качественное описание признаков ядра
Фаза менструального цикла	Строковый	Качественное описание фазы менструального цикла, во время которого было проведено исследование
Комплекс наблюдений	Числовой	Количество атрибутов, необходимых для подтверждения гипотезы
Общая характеристика результата исследования по Bethesda	Строковый	Характеристика, описывающая неспецифические параметры, связанные с тем или иным результатом исследования

При обобщении литературных данных было обнаружено, что размер клеток, для которых характерен результат диагностики по NILM, может изменяться от 10-1800 мкм (Рис. 1) [Tiwana et al 2022; Bispo Pereira et al 2023; Senthil Kumar et al 2022; Kaiser et al 2023]. При наличии клеточных изменений, для которых характерна в зависимости от происхождения клеток изменений колеблется приблизительно от 1200 до 2000 микрометров, при этом большинство значений попадает в диапазон от 1450 до 1550 микрометров. При описании качественных характеристик клеток чаще всего наблюдалась круглая или овальная форма.

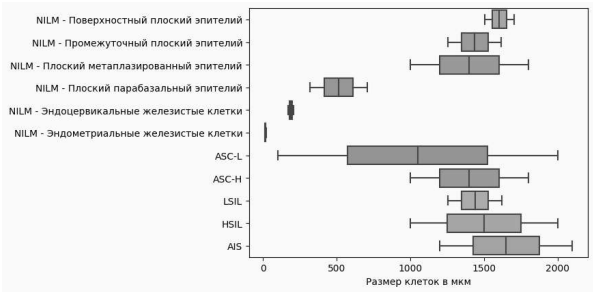


Рис. 1. Диаграмма размаха для размеров клеток в зависимости от их происхождения и результата диагностики по Bethesda от 2014г.

При анализе результатов диаграмм размаха ЯЦО также как и с размерами клеток прослеживается довольной широкий разброс значений (Рис. 2) [Rezende et al 2021; Alrajjal et al 2021; Senthil Kumar et al 2022]. Было отмечено, что интенсивность клеточных изменений и риск малигнизации по типу РШМ, коррелируют с их ЯЦО.

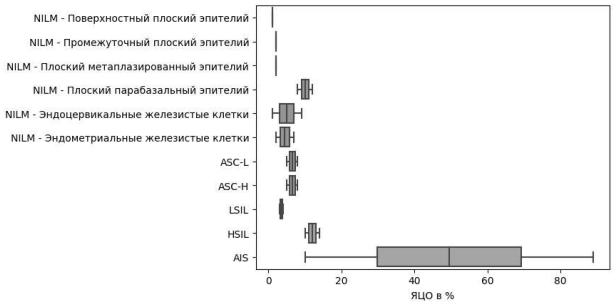


Рис. 2. Диаграмма размаха для ЯЦО в зависимости от их происхождения и результата диагностики по Bethesda от 2014г.

После обобщения и внесения значений атрибутов и гипотез в IACPaaS-платформу ИАПУ г. Владивостока, на основе формализованных данных базы знаний был реализован прототип классификатора клеточных изменений зоны трансформации зоны эндоцервикса по шкале классификации Bethesda от 2014 г. (Рис. 3) [Сайт IACPaaS, 2020].

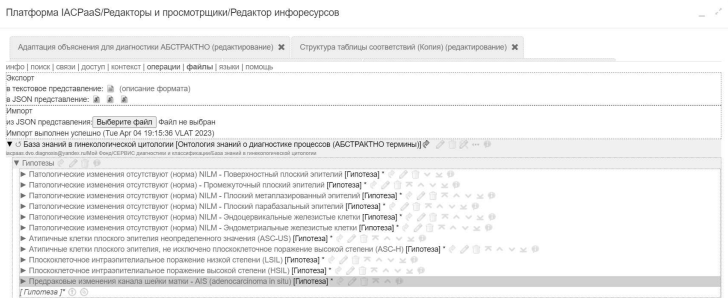


Рис. 3. Общий вид знаний для классификатора на платформе IACPaaS ИАПУ г. Владивостока

Алгоритм работы на платформе следующий: пользователь-эксперт, авторизовавшись в системе, переходит в классификатор, где на основе предопределенных правил вводит данные в базу знаний (Рис. 4) [Сайт IACPaaS, 2020]. Если рассматривать данную ситуацию с точки зрения практикующего врача или ординатора, то остановившись окуляром над клеткой, степень изменения которой вызывает у него вопрос, он вносит имеющиеся данные в систему.

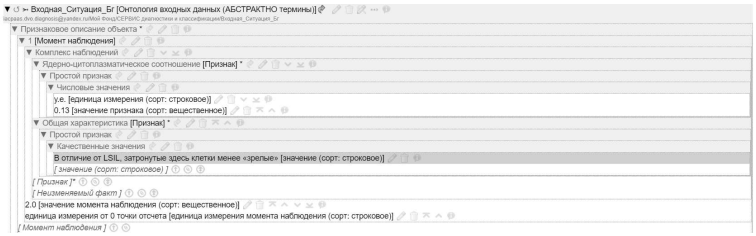


Рис. 4. Пример ввода входных данных, характеризующих предраковые изменения клеток

Далее система на основе установленных в ней правил и определенных атрибутов экспертом проводит установление 3 типов гипотез: подтвержденных, возможных и опровергнутых, в зависимости от того, как введенные пользователем-экспертом данные согласуются с прописанными в системе параметрами (Рис. 5). На выходе он может посмотреть статус диагностики и определить, на основании каких атрибутов была выбрана или отвергнута та или иная гипотеза.

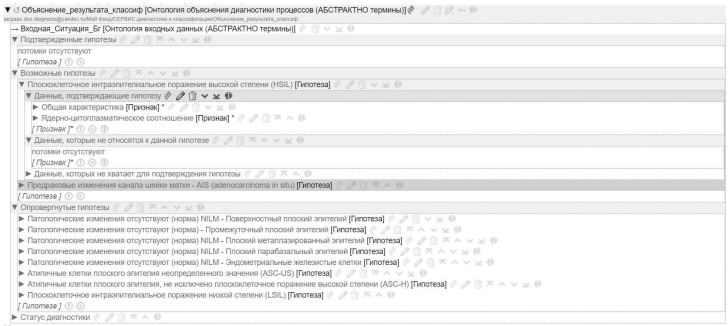


Рис. 5. Выходные данные для введенных данных

4 Обсуждение

На основании результатов обзора литературы и системного анализа области была разработана онтология базы знаний для классификатора, использующего современные алгоритмы и методы анализа объективных атрибутов. Данная система может стать прообразом СППВР, позволяющей на основе определенных методами машинного обучения параметров изображения проводить установление гипотезы о результате диагностики и соответствующем типе клеточных изменений в гинекологии.

На данный момент проводится тестирование разработанного классификатора на большом наборе изображений, полученных в ходе оцифровки стёкол с клетками, полученных в ходе Пап-теста в Центре лабораторной диагностики Медицинского центра Дальневосточного федерального университета г. Владивостока, для оценки его точности и эффективности в сравнении с полученными результатами атрибутов и описаниями традиционной микроскопии и анализатора графических данных [Щеглов и др., 2023].

В целом, результаты данного исследования дают достаточно ценную информацию о количественных характеристиках клеточных изменений, представленных на диаграммах диапазонов из результатов исследований других специалистов. Эти результаты имеют важное значение для объективной диагностики и раннего выявления клеточных изменений, ассоциированных с повышенным риском РШМ. Однако до сих пор остаются “белые пятна”, связанные с вопросами интерпретации результатов по типу ASC-US, в частности, ASC-H и ASC-L.

5 Заключение

Создание базы знаний и применение на ее основе классификатора для совершенствования цитологической диагностики в гинекологии дает достаточно большие перспективы для повышения точности и эффективности диагностики РШМ путем

объективизации медико-лабораторных данных. Используя результаты современных исследований и передовой международный опыт, системы по СППВР могут помочь в выявлении аномалий клеточных изменений как можно раньше и предоставлять пациентам лучшие варианты их лечения и последующего наблюдения в условиях отделений лабораторной диагностики и гинекологии в больницах и при скрининге в поликлиниках. В свою очередь, это может привести к улучшению результатов ранней диагностики, повышению качества оказываемой медицинской помощи и профилактике РШМ в центрах лабораторной диагностики при больницах.

6 Благодарности

Авторы благодарят за финансовую поддержку программы Приоритет-2030 Дальневосточного федерального университета. Также благодарят Е.А. Шалфееву (Институт автоматики и процессов управления, Владивосток) за консультацию по использованию онтологической среды и фреймворка IACPaaS.

Литература

[1] Giuliani, E., As-Sanie, S., & Marsh, E. E. (2020). Epidemiology and management of uterine fibroids. *International journal of gynecology and obstetrics: the official organ of the International Federation of Gynaecology and Obstetrics*, 149(1), 3–9. https://doi.org/10.1002/ijgo.13102.

[2] Профилактика и скрининг РШМ: обзор международных клинических рекомендаций / Л. Г. Пивазян, А. Л. Унанян, Ю. Д. Давыдова [и др.] // *Проблемы репродукции*. – 2022. – Т. 28. – № 5. – С. 90-99. – DOI 10.17116/repro20222805190. – EDN SCTXVS

[3] Клинические рекомендации по диагностике и лечению рака шейки матки / А. Кравец, В. В. Кузнецов, К. Ю. Морхов [и др.]. – Москва : *Общероссийский союз общественых объединений Ассоциации онкологов России*, 2020. – 51 с.

[4] Eun, T. J., & Perkins, R. B. (2020). Screening for Cervical Cancer. *The Medical clinics of North America*, 104(6), 1063–1078. https://doi.org/10.1016/j.mcna.2020.08.006

[5] Smith, E. R., George, S. H., Kobetz, E., &et al. (2018). New biological research and understanding of Papanicolaou's test. *Diagnostic cytopathology*, 46(6), 507–515. https://doi.org/10.1002/dc.23941

[6] Rezende, M. T., Bianchi, A. G. C., & Carneiro, C. M. (2021). Cervical cancer: Automation of Pap test screening. *Diagnostic cytopathology*, 49(4), 559–574. https://doi.org/10.1002/dc.24708

[7] Chrysostomou, A. C., Stylianou, D. C., Constantinidou, A., et al. (2018). Cervical Cancer Screening Programs in Europe: The Transition Towards HPV Vaccination and Population-Based HPV Testing. *Viruses*, 10(12), 729. https://doi.org/10.3390/v10120729

[8] Staats, P. N., Booth, C. N., Rosenthal, D. L., et al. (2019). Self-collected Papanicolaou tests in the United States market: more questions than answers. *Journal of the American Society of Cytopathology*, 8(6), 342–351. https://doi.org/10.1016/j.jasc.2019.05.003

[9] Banerjee, D., Mittal, S., Mandal, R., & Basu, P. (2022). Screening technologies for cervical cancer: Overview. *CytoJournal*, 19, 23. https://doi.org/10.25259/CMAS_03_04_2021

[10] Swailes, A. L., Hossler, C. E., & Kesterson, J. P. (2019). Pathway to the Papanicolaou smear: The development of cervical cytology in twentieth-century America and implications in the present day. *Gynecologic oncology*, 154(1), 3–7. https://doi.org/10.1016/j.ygyno.2019.04.004

[11] Alrajjal, A., Pansare, V., Choudhury, M. S. R., et al. (2021). Squamous intraepithelial lesions (SIL: LSIL, HSIL, ASCUS, ASC-H, LSIL-H) of Uterine Cervix and Bethesda System. *CytoJournal*, 18, 16. https://doi.org/10.25259/Cytojournal_24_2021

[12] Tiwana, K. K., Kaur, M., Goyal, S., & Bhandhari, L. (2022). The role of micronucleus scoring in cervical papanicolaou smears: A 1-year study. *Annals of African medicine*, 21(4), 355–360. https://doi.org/10.4103/aam.aam_87_21

[13] Bispo Pereira, E. H., Camilo-Júnior, D. J., Correa Garcia Pires D'ávilla, S., et al. (2023). Cervical cytology results among pregnant and non-pregnant women in Brazil. European journal of obstetrics, gynecology, and reproductive biology, 282, 161–167. https://doi.org/10.1016/j.ejogrb.2023.01.027

[14] Senthil Kumar, D., & Ravikumar, G. (2022). Quality metrics in cervical cytopathology: A single institutional study. Cytopathology : official journal of the British Society for Clinical Cytology, 33(2), 230–235. https://doi.org/10.1111/cyt.13082

[15] Kaiser, L. A., Kupec, T., Najjari, L., Stickeler, E., & Wittenborn, J. (2023). Predictors of CIN2+ in Patients with PAP III-P (ASC-H): A Cross-Sectional Study. Diagnostics (Basel, Switzerland), 13(6), 1066. https://doi.org/10.3390/diagnostics13061066

[16] Сайт IACPaaS ДВО РАН ИАПУ г. Владивостока // IACPaaS URL: https://iacpaas.dvo.ru/ (дата обращения: 14.06.2023).

[17] Свидетельство о государственной регистрации базы данных № 2023620481 Российская Федерация. Определение типов клеток по морфологическим характеристикам для цитологической диагностики рака шейки матки : № 2023620190 : заявл. 01.02.2023 : опубл. 06.02.2023 / Б. О. Щеглов, Г. В. Рева, С. Н. Щеглова.

Increasing the efficiency of teaching programming through the use of innovative technologies in programming paradigms in higher education

Sanobar Khakimova[1]
[1]*Urgench Branch of Tashkent University of Information Technologies named after Muhammad al Khwarizmi*

sanobarhakimova30081971@gmail.com

Abstract: This article provides materials on various educational technologies used to facilitate teaching and learning of a programming methods and paradigms course in universities. Programming paradigms are an important topic in computer science and information systems development, and the demand for them in the computer industry is increasing day by day. It is known that there is a lack of availability to develop systematic and algorithmic thinking and appropriate methodological tools in the teaching of the programming paradigms course using cognitive technologies in universities. In the article, we focus on the problem-based and semiotic approach, which is considered one of the main methodological approaches to teaching programming paradigms in higher education. A problem-based approach involves building a programming course based on a system of specially designed tasks or situations whose content is interesting to students and expands their programming experience. The semiotic approach is aimed at the purposeful development of students' symbolic actions. At the end of the article, the didactic possibilities of the ideas and principles of the semiotic approach in increasing the effectiveness of teaching programming are concluded.

The objective of this study is to pinpoint obstacles encountered when employing technology within the educational framework, with a focus on university-level courses on programming paradigms. Further, we intend to suggest efficient strategies for future incorporation of technology within this context. We posit that the integration of the suggested innovative technologies and efficacious solutions into programming paradigm coursework will enhance both the instructional process and learning outcomes in the subject matter.

Keywords: educational technology, programming methods and paradigms course, problem-based approach, semiotic approach, teaching and learning.

1 Introduction

In the present era, initiatives such as the "One Million Programmers" project are being undertaken in our republic with the aim of popularizing information and communication technologies among younger demographics. Concurrently, these initiatives also strive to enhance the digital literacy skills across all strata of society. By introducing online and virtual instructional technologies, these projects are actively contributing to the training of highly proficient specialists in the field. This sets the task of training higher education personnel with high intellectual potential, capable of competing on equal terms with the world's youth.

Numerous domestic and international researchers have conducted extensive studies on the methodologies of teaching the course of programming paradigms. Specific methodological approaches to teaching programming paradigms have been discussed by several of these scholars.

Gorodnya L. V., for instance, argued that the programming paradigm acts as the foundational conceptual framework that delineates task definition and problem-solving within the context of programming [5]. A paradigm essentially serves as a linguistic mechanism to describe facts, events, and processes that, while not co-existing, are intuitively grouped under a common concept. It provides a guideline for programmers on how to approach a programming task. For instance, in object-oriented programming, the paradigm instructs the programmer to perceive the program as a set of intercommunicating objects. Conversely, in functional programming, the paradigm portrays the program as a sequence of function evaluations [5].

The term "programming paradigm" has been assigned multiple interpretations in academia. Daniel G. Bobrow, for instance, conceives it as "a programming style that signifies the programmer's intention"[6]. In her book "Comparative Programming Languages", Linda Wieser Friedman postulates that "a programming paradigm represents an approach to resolving programming issues"[7]. Pamela Zave extends this further by articulating that "a programming paradigm delineates a perspective on conceptualizing computer systems"[8].

Thus, a programming paradigm outlines a broad methodology for crafting programs. Robins A., Rowntree J., and Rowntree N. argue that the elementary programming course offered at universities has been the fulcrum of extensive debate among educators in computer science[1].

Echoing this sentiment, Astrachan O., Bruce K., Koffman E., Kölling M., and Reges S. assert that programming is inherently complex, which inevitably renders the task of instructing programming an even more daunting challenge[2].. Basically, paradigms play an important role in software development planning when creating an application. Taking these into account, programming methods and paradigms can be divided into practical, cognitive-oriented, semiotic, and problem-based approaches, depending on their place in the teaching process. This article emphasizes the semiotic and problem-based approach and makes its principles concrete in the process of teaching the science of programming methods and paradigms. The outlined principles circumvent students' contextual comprehension of sign systems, the emotional value field, and the sequential acquisition of sign systems, which simultaneously amplifies abstraction capabilities. These principles also fail to encapsulate the unity representation of tangible and virtual objects and the discrete interpretation of information. Fundamentally, this manuscript delineates the methodological stratagems employed in the teaching of programming paradigms, along with the salient features associated with the implementation of a semiotic approach.

Pedagogical University N.A. Kurganova's "Formalization and Modeling" course is based on the consideration of the semiotic approach and the development of sign-symbol activity of students as the main approach[3]. The characteristics of programming paradigms are very suitable for teaching based on the semiotic approach.

The principles embodied within the semiotic approach elucidate avenues of interaction between students and educators, and may serve as a foundational framework for architecting instructional methodologies for programming at university level. The core of the semiotic approach in programming pedagogy is rooted in the cultivation of specific symbolic actions within students, which are requisite for future professionals in the field of development. The primary stages of the semiotic approach are detailed in the following sections.

1.Replacement (repeat the function of the replaced item).

Implementation of the task: in the process of determining input and output data, the types of variables and their properties are concluded.

For example: let's define data types about car color, weight, model, number of doors, price. Int, double, string type attributes are considered here.

2.Coding (replacing one symbol system with another).

Execution of the task in a practical setting: Identification of the algorithmic constructs (commands, functions, and procedures) that are essential to the problem-solving process.

To access a private attribute, use the public "get" and "set" methods. Table 1

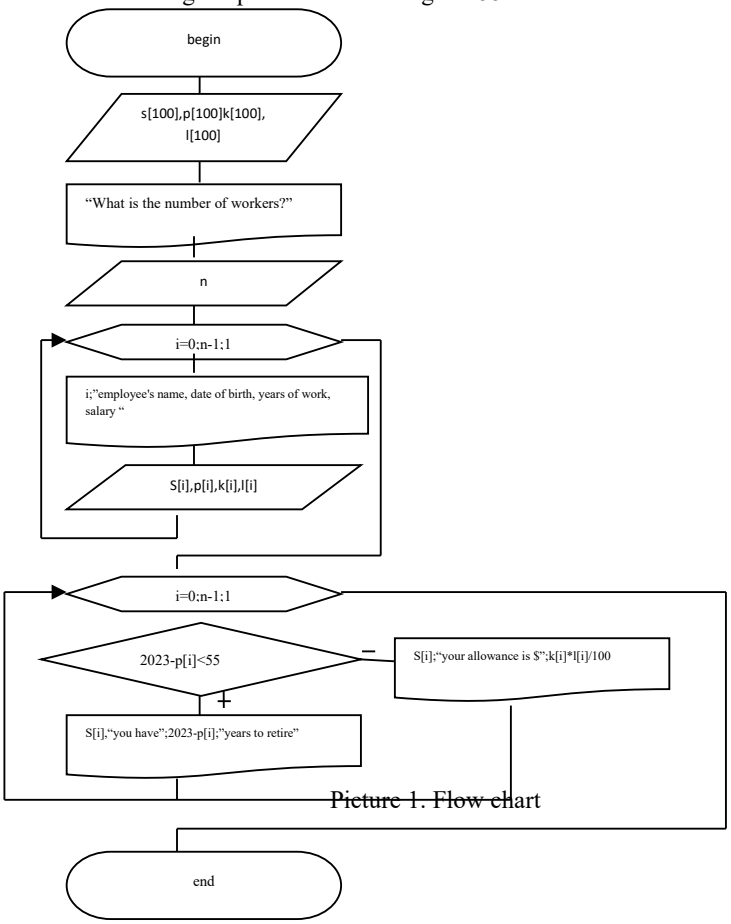
Example	Solution

To access a private attribute, use the public "get" and "set" methods:	#include <iostream>
	#include<string>
	using namespace std;
	class worker {
	private: // Private attribute
	string name;
	public: // Setter
	void setName(string s) {
	name= s;
	} // Getter
	string getName() {
	return name;
	}
	};
	
	int main() {
	worker myObj;
	myObj.setName("Artur");
	cout << myObj.getName();
	return 0;
	}
	
	class MyClass { // create a class
	public: // Free access information
	int myNum; // property (int variable)
	string myString; // property (string variable)
	};

3.Schematization (designation of connections between events).

Practical implementation of the task: present the algorithm for solving the problem in a visual form (for example, in the form of a flow chart).

For example: based on the name of the employee, year of birth, amount of time and salary of the employee, the block diagram of the issue of calculating his pension after the age of 55 is described.



Picture 1. Flow chart

4.Modeling (turning an object into a model while preserving its important features).

The practical execution of tasks involves creating a program that emulates a particular process. This necessitates that students have a pre-existing familiarity with certain symbolic systems.

For example: Create a class called "Human". Select the attributes as follows: the number of years of service of the employee, salary, allowance. In the "Pensioner" method, if the employee is over 55 years old, provide information and calculate his pension based on the following formula =work experience*salary/100*3, if the employee's age is less than 55, then output information to the console about how many years are left until retirement.

Select the attributes as follows. Table 2

Example	Solution
Create a class called "Human". Select the attributes as follows: the number of years of service of the employee, salary, allowance	#include <iostream> #include <iostream> #include<string> using namespace std; class Person { public: string FIO[100]; int date[100]; int year[100]; double salary[100]; int setQ(int i,int p,string s, int k,double l) {date[i]=p; FIO[i]=s; year[i]=k; salary[i]=l;}; int pension(int i) { if(2023-date[i]>55) cout << FIO[i]<<" your allowance is $"<<year[i]*salary[i]/100<<endl;

	else cout<<FIO[i]<<" you have"<<55-(2023-date[i])<<" years to retire"<<endl; }
	};
	int main() { int p,k,n; double l; string s;Person ob1;
	cout<<" What is the number of employees in the company?";cin>>n;
	for(int i=0 ;i<n;i++)
	{ cout<<" Enter the employee's first and last name:"; cin >>s; cout<<" Enter date of birth:"; cin>>p;
	cout<<" how many years the employee worked:"; cin>>k; cout<<"Enter the employee's salary:"; cin>>l;
	ob1.setQ(i,p,s,k,l);}; for(int i=0;i<n;i++)
	ob1.pension(i);
	return 0;
	}

Such experience has a positive effect on the learning of features in programming paradigms. The main challenge for students is to choose the structures of the sequence of actions from thinking when formulating a problem. In this, the semiotic approach is used in full, which makes it possible to describe the translation process of the solution algorithm.

Helping students understand the programming process is one of the most important tasks when teaching programming, programming paradigms, and it helps you gain student confidence. Programming paradigms are heuristics used to solve an algorithmic problem. According to Orit Hazzan, Tami Lapidot, Noa Ragonis [4], the programming paradigm analyzes the problem through a specific lens and, based on this analysis, develops a solution to this problem by dividing the solution into specific building blocks and determining the relationships between them.

The problem-based approach is practice-oriented and presents programming as a problem-solving tool. A problem-based approach is best used in the final lessons for each module. It is possible to develop students' logical abilities only if they are actively involved in the process of acquiring new knowledge. Problem-based learning, a critical aspect of a student's creative activity, is regarded as a potent tool for fostering autonomous logical thinking. This pedagogical approach, as proposed by O.V. Obukhova[10], involves the formulation of a problematic situation founded upon students' current knowledge. Occasionally, students may draw conclusions that contravene known facts, indicating that their knowledge is incomplete and additional information is required for conflict resolution. Such an approach invariably piques students' interest in the subject matter, resulting in enhanced cognitive efficiency[9].

Problem-based learning strategies rely on the active cognitive participation of students. They are designed to foster the creation of problem situations, the refreshing of knowledge, and the identification and resolution of complex issues that necessitate discerning the phenomena and laws that underpin specific facts. The educator constructs a problem situation and steers the students towards its resolution. Consequently, students not only acquire new knowledge but also develop novel approaches to problem-solving. The teacher employs various specialized methodological techniques to craft the problem situation[10].

Depending on the nature of the interaction, teachers and students distinguish four levels of problem-based learning [10]:

1.Level of non-independent activity - students' perception of the teacher's explanation, mastering the model of mental action in a problem situation, performing independent work, oral repetition;

Non-independent activity in the study of programming paradigms is manifested in the study of ready-made texts of programs. For example, in the object-oriented paradigm, the general form of the class declaration operator can begin by considering constructions.

Students' perception of the teacher's explanation. Table 3

Example	Solution

Create a className Select the attributes as follows.	class ClassName {
	private: //scope within a class,
	//only member functions of the given class have access
	List of class members;
	protected: //can be used by methods of this and derived classes
	List of class members;
	public: //visible outside the class, can be accessed from outside
	List of class members;
	};
	
	For example ,
	class Student {
	private:
	char *name; //name is a private member of the class

	public:
	void show (void); //function "show" - public
	void setname (char *); //function "set name" - public
	};

2. The level of semi-independent activity is the application of previous knowledge in a new situation. As a result, it is possible to learn new material based on the teacher's instructions. Application templates can be used for this. For example, Create a class Car Select the attributes as follows.

New material based on the teacher's instructions. Table 4

Example	Solution

Create a class Car Select the attributes as follows.	class Car { // The class
	… : // Access specifier
	… ; // Attribute “brand”
	… ; // Attribute “speed”
	int year; // Attribute
	… (… x, … y, … z) { // Constructor with parameters
	brand = … ;
	speed = … ;
	year = … ;
	}
	};
	
	int main() {
	// Create Car objects and call the constructor with different values
	Car carObj1("BMW",…, 1999);
	… … ("…” , … , …);
	
	// Print values
	cout << carObj1.brand << " " << … << " " << … << "\n";
	cout << … << " " << … << " " << … << "\...";
	return 0;
	}

3.Level of independent activity - performance of tasks of the reproductive-research type, in which the student applies previous knowledge in a new situation, creates problem programs of moderate complexity, etc. For this level, the student should be given a task that he must solve without the help of the teacher. For example, the following example creates the player class. In the program, the class function show_player is defined outside the constructor. Then two arrays of player type are created and information about each is displayed.

Performance of tasks of the reproductive-research type.Table 5

Example	Solution

the class function show_player is defined outside the constructor.	#include <iostream>
	#include <string>
	using namespace std;
	class player
	{
	public:
	player();
	player (string name,int weight, int age);
	void show_player (void);
	private:
	string name;
	int weight;
	int age;
	};
	player::player()
	{
	name="";

	weight = 0;
	age = 0;
	};
	player::player(string name,int weight, int age)
	{
	player::name=name;
	player::weight = weight;
	player::age = age;
	};
	void player::show_player (void)
	{
	cout<<"Ism: " << name << endl;
	cout<<"Vazn: " << weight << endl;
	cout<<"Yosh: " << age << endl;
	}
	class array_player
	{ public:
	void show_array(player a[],int n)
	{ for(int i=0;i<n;i++)
	{a[i].show_player();cout<<endl;}}
	void input_array(player a[],int n)
	{string name;int weight,age;
	for(int i=0;i<n;i++)
	{cin>>name>>weight>>age;
	a[i]=player(name,weight,age);
	}
	}
	};
	
	int main()
	{array_player arr;
	Player happy[]={player("Olimov",58,24),
	player("Abdiyev",72,35)};
	arr.show_array(happy,2);
	player matt[2];
	arr.input_array(matt,2);
	arr.show_array(matt,2);
	return 0; }

4. The level of creative activity - performing independent work that requires creative imagination, logical analysis and assumptions, finding a new way to solve an educational problem, proving it independently; draw independent conclusions. At this level, the teacher should set a task, and in solving it, the student not only works independently, but also approaches it creatively. The task of the teacher can stimulate creativity using the element of the project method. For example, the name of factories with more than 50 employees should be output to the console.

Logical analysis and assumptions. Table 6

Example	Solution

the name of factories with more than 50 employees should be output to the console.	#include <iostream>
	#include <string>
	using namespace std;
	class Sewing{
	public :
	string NameFactory[20];
	int NumberOfWorkers[20];
	int print(int m)
	{ cout<<"Factories with more than 50 employees: "<<endl;
	for(int i=0;i<m;i++)
	{
	if(NumberOfWorkers[i]>50) cout<<NameFactory[i]<<"\n";
	} }};
	int main () {
	cout<<"Number of factories m=";

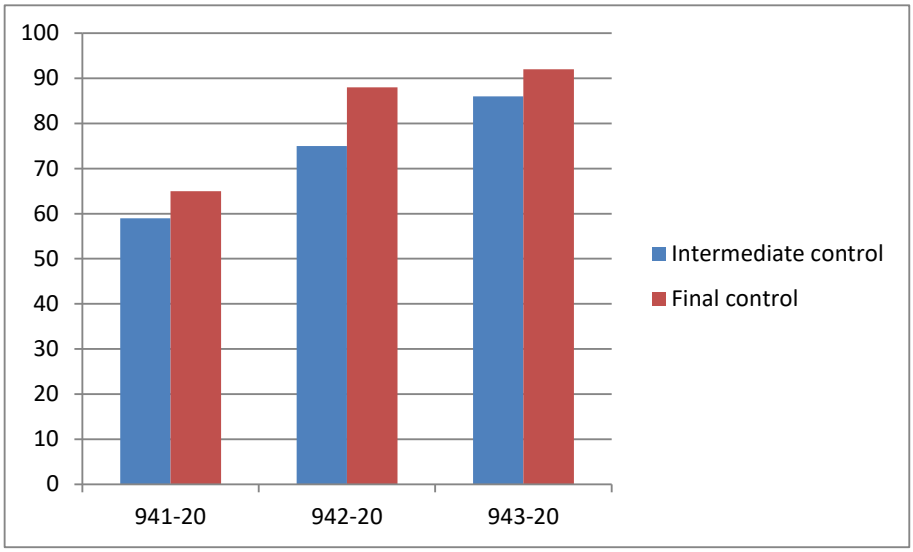
```
  int m;cin>>m;Sewing n;
  for(int i=0;i<m;i++){
        cout<<"Factory name:"<<endl;
   cin>>n.NameFactory[i];
     cout<<"The number of workers in the factory:"<<endl;
   cin>>n.NumberOfWorkers[i];

  }n.print(m);
  return 0;
}
```

A problem-based approach does not allow students to comprehensively consider all types of tasks that a future specialist may face in a training course in the field of programming. The problem-based approach has a very useful practical component, without which complete preparation for future professional activity is unthinkable. However, when talking about approaches to teaching programming paradigms in the comprehensive development of all skills necessary for a developer, one should not forget about the increasingly popular semiotic approach.

ANALYSIS OF THE RESULTS OF THE PEDAGOGICAL EXPERIMENT.

23 students from Group 941-20, 24 students from Group 942-20, and 13 students from Group 943-20 were selected to evaluate the effectiveness of the above methods in teaching the course of Programming methods and paradigms at the university. The "Programming methods and paradigms" course was conducted in the form of a simple lecture in Group 941-20, and in the rest of the groups, classes were conducted based on the problem-based and semiotic approach.



Picture 2 . Analysis of the results of the pedagogical experiment.

Based on the results of the mid-term and final exams, the following indicators were recorded in the groups.
The experience shows that teaching the course with regular lectures and classroom activities achieved 65 percent of the final control, while groups trained on the basis of problem and semiotic approaches achieved 88 and 92 percent. Furthermore, the students of Urganch branch of TUIT named after Muhammad al-Khwarizmi participated in ICPC 2022-23 International Algorithmic Programming Olympiad in Kazakhstan, and 2 teams actively participated in the semi-finals of the Olympiad. In addition, students are taking pride of place in Informatics and information technologies at the Republican Olympiads. We have come to the conclusion that the use of new innovative methods to achieve high efficiency in higher education in teaching programming paradigms course is highly effective.

CONCLUSIONS

Thus, programming paradigms, especially object-oriented programming, are the main paradigms in modern application development. Its feature is the representation of the subject area in the form of objects that combine data

and behavior. The teaching methodology of the course, the logical structure of teaching the subject, the activation of students in learning help to implement the modern concept of education in the field of information technology and develop advanced teaching methods. The base of modern approaches, created on the basis of the structure of science, not only increases the interest of university students in the subjects studied, but also allows the teachers themselves to keep achievements in the arsenal of academic subjects.An example topic in our semiotic and problem-based approach method is the "Object-Oriented Programming Paradigm" section of the "Algorithms and Programming Fundamentals" course and "Programming" on a logical basis, along with experimental verification, showed educational efficiency in teaching these information technologies and improving the systematic knowledge of university students.The selection and composition of the content of modern educational science and the topic "Object-oriented programming paradigm" (a collection of terms and concepts, facts, types of educational and research activities) provides interaction between the subject and methods of teaching, science methodology and the field of knowledge. Orientation of educational materials to the formation of an integrated system of professional and educational skills increases the quality of higher professional education.

REFERENCES

[1]. Robins A., Rountree J., Rountree N. (Eds.). Learning and teaching programming: A review and discussion. Robins A., Rountree J., Rountree N. *Journal of Computer Science Education 13(2)*, 2003.

[2]. Astrachan O., Bruce K., Koffman E., Kölling M., Reges S. (Eds.). Resolved: objects early has failed. In: *Proceedings of the thirty-sixth SIGCSE technical symposium on Computer science education*, St. Louis, Missouri, USA, 2005.

[3]Kurganova N.A. The development of the sign-symbolic activity of students in the process of teaching informatics based on the semiotic approach. Abstract of the dissertation ... candidate of pedagogical sciences. Omsk, 2006.

[4]Orit Hazzan ● Tami Lapidot ● Noa Ragonis “Guide to Teaching Computer Science”An Activity-Based Approach p-34

[5]Gorodnyaya L.V. Paradigm of programming : a course of lectures / ; Novosib. state un-t. – Novosibirsk: RIC NSU, 2015. 198–206 p. [6] Daniel G. Bobrow, “If Prolog is the answer, what is the question”, in FGCS,Tokyo, Japan, 1984, pp. 138-145.

[7] Linda Wieser Friedman, Comparative Programming Languages: Generalizing the
Programming Function. Prentice Hall, 1991.

[8] Pamela Zave, “A compositional approach to multiparadigm programming,” IEEE Softw., vol. 6, no. 5, pp. 15-25, Sept. 1989.

[9]http://www.znanie.org/jornal/n2_01/psih_podhod.html

[10] O.V. Obukhova Practice-oriented approach in teaching programming Computer science and information technology: Proceedings of the XV All-Russian conference with international participation of students, graduate students and young scientists "Science and Education" - Tomsk 2011.

Self-tuning hyper-parameters for unsupervised cross-lingual tokenization

Anton Kolonin[1,2][0000−0003−4180−2870]

[1]*Novosibirsk State University, Novosibirsk, Pirogova 1, 630090, Russia.*
[2]*Aigents, Novosibirsk, Pravdy 6-12, 630090, Russia.*

akolonin@gmail.com

Annotation. *We explore the possibility of meta-learning for the language-independent unsupervised tokenization problem for English, Russian, and Chinese. We implement the meta-learning approach for automatic determination of hyper-parameters of the unsupervised tokenization model proposed in earlier works, relying on various human-independent fitness functions such as normalized anti-entropy, compression factor and cross-split F1 score, as well as additive and multiplicative composite combinations of the three metrics, testing them against the conventional F1 tokenization score. We find a fairly good correlation between the latter and the additive combination of the former three metrics for English and Russian. In the case of the Chinese, we find a significant correlation between the F1 score and the compression factor. Our results suggest the possibility of robust unsupervised tokenization of low-resource and dead languages and allow us to think about human languages in terms of the evolution of efficient symbolic communication codes with different structural optimization schemes that have evolved in different human cultures.*

Keywords: cross-lingual, language learning, meta-learning, tokenization, unsupervised.

1 Introduction

Earlier studies on unsupervised tokenization learning relying on interpretable graph-based models relying on probabilistic metrics as well as metrics expressing uncertainty, such as ”freedom of transition” or ”transition freedom” have shown promising results, especially relying on the latter metrics applied for English, Russian and Chinese languages [1–3].

However, the latest cross-lingual study carried out by previous researchers for English, Russian, and Chinese [3] has found that different languages require different hyper-parameters to be used in the process of the model building and further application of it, in order to get highly accurate tokenization output. That means, in order to obtain completely unsupervised solution for a language acquisition task, starting from the tokenization level, some extra metric is required to drive the construction of the language specific models and definition of the parameters for the application of these models without of supervision from human and prior knowledge of the language itself.

To address the latter problem, in this work we explore the possibility of meta-learning for the language-independent unsupervised tokenization setup referring to the latest of prior studies [3]. We implement the meta-learning approach for automatic determination of hyper-parameters being applied to unannotated raw training text corpus, searching for human-independent fitness functions such as normalized anti-entropy, compression factor and so called ”cross-split *F1* score”, as well as additive and multiplicative combinations of the three metrics, testing them against the conventional *F1* tokenization score.

2 Experimental Approach

2.1 Metrics for Tuning of Hyper-Parameters

In order to find human-independent and culture-agnostic metric for optimal segmentation of symbolic sequences involved into massive human-to-human communications, such as human languages, we attack it from from few different perspectives. First, it is ”compression” view - perspective of information compressing nature of human language [4], which assumes that cultural evolution of linguistic structure of each of the human languages is targeting information compression goal. Second, it is ”information” view - perspective of information theory where the structure of a language could be considered as driven my objective of entropy minimization across all possible messages being transmitted in this language [5]. Third, it is ”concordance” view - social and contractual perspective of the language where is is thought as a medium of enabling efficient communication between different groups minimizing discrepancy of social contexts across these social groups [6, 7].

In fact, the three perspectives above might be seen at least complementary or even equivalent. For instance, a metric measuring optimal compression can be seen from perspective of minimized amount of entropy [8] and respective measure. Moreover, as it is seen later in our study, we find that ”compromising” capacity of a language model to provide consensus across different training corpora, which can be attributed to different social groups, is also highly correlating with the former ”compressing” and ”informative” metrics. Below we define the three measures.

The ”compression” tokenization metric which we call ”compression factor” *C%* corresponds to the ratio between numerator as ”compressed” size of training set given current tokenization model and denominator as uncompressed size of training set. The compressed size is evaluated as length of sequence of token indexes in tokenized text corpus entirely plus size of the ”dictionary” - sum of lengths of all tokens in it. The uncompressed size of training set is evaluated just as count of symbols in it.

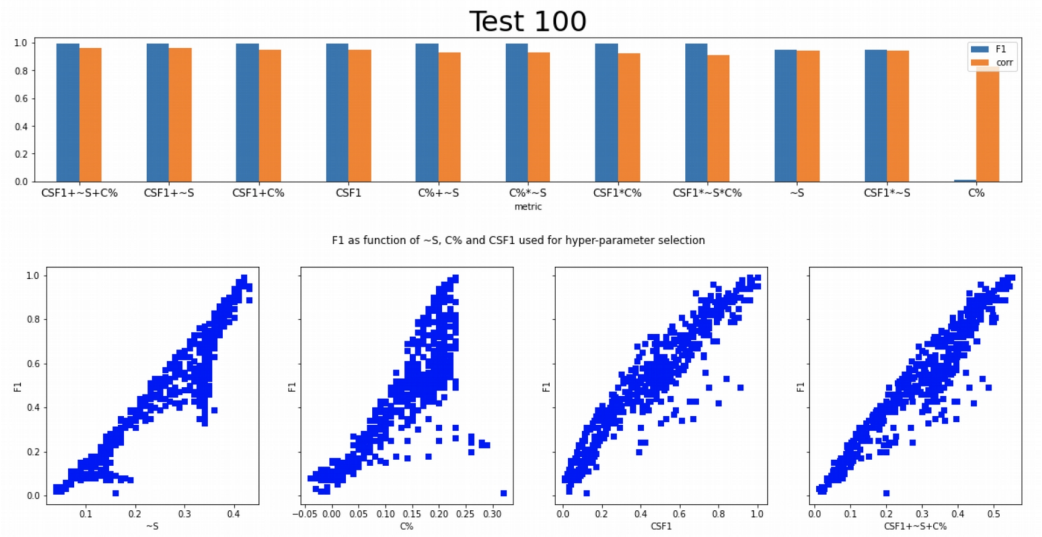


Fig.1. For English language and test set of 100 lines. Upper row: blue bars correspond to the *F1* score obtained based on set of hyper-parameters corresponding to the maximum target metric on horizontal axis, blue bars correspond to Pearson correlation between the *F1* score and respective metric.

For the ”information” tokenization metric we introduce so called ”normalized anti-entropy” $\tilde{S}$ defined as $\tilde{S} = 1 - H/(log2(N))$ where is the H is a Shannon entropy of entire training set tokenized with given tokenization model on base *2* and N is size of of the lexicon under such tokenization model.

For the ”concordance” metric we use what we call cross-split *F1* score, defined as follows. First, we split the training set corpus in two pieces of the same size, call them set A and set B. Next, we create the graph traversal models across N-grams according to the previously cited work [3] for each of the corpus, call them *M(A)* and *M(B)*. Then, we tokenize the test set with the both models, so that

T(M(A)) and *T(M(B))* are obtained. Finally, evaluate the cross-split *F1* score of tokenization as *CSF1* for *T(M(A))* against *T(M(A))* having one as a "ground truth" for another.

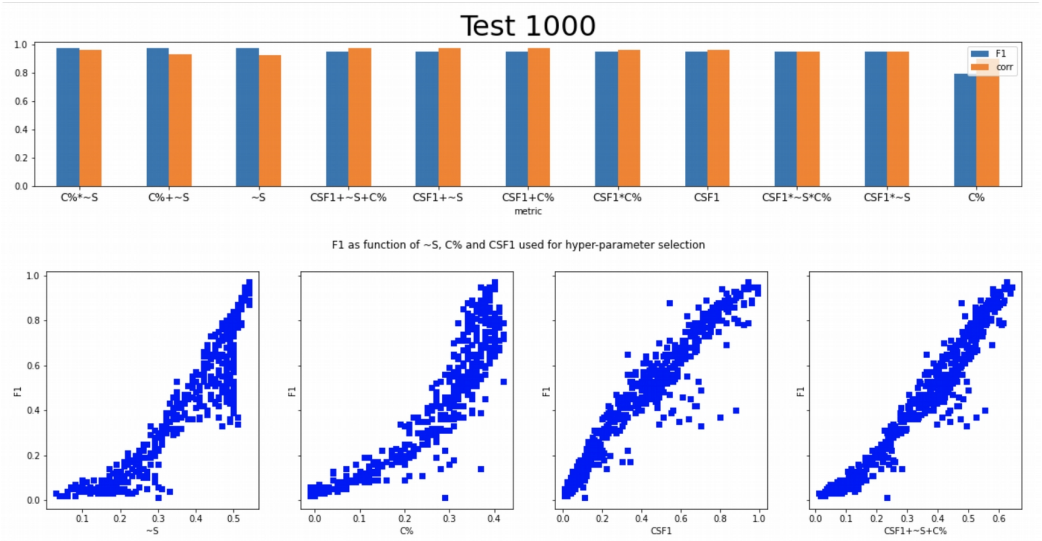


Fig.2. For English language and test set of 1000 lines. Upper row: blue bars correspond to the *F1* score obtained based on set of hyper-parameters corresponding to the maximum target metric on horizontal axis, blue bars correspond to Pearson correlation between the *F1* score and respective metric.

2.2 Hyper-Parameters for Tuning

The experimental setup involved finding conventional *F1* score metrics for tokenization of test set used in the referenced work [3], using the "transition freedom" from the same work as a metric driving tokenization, given the three-dimensional grid of three hyper-parameters such as follows:

1) The combination of N ranks N_s used to perform model graph traversal and the "mean" metric computation based on a specified subset of N-grams. We have explored every possible individual value of N in range from 1 to 7 (1 to 3 in case of Chinese) as well as combinations of N-values.

2) Model compression threshold T_{mc} used to remove low-frequency N-grams (corresponding to vertices and transitions between them on the model graph). We have used the following values: 0.0 (corresponding to no compression at all), 0.0001, 0.001, 0.01, and 0.1.

3) Tokenization metric threshold T_{tm}: the value of a metric exceeding this level would correspond to a token boundary. We have used the following values: 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9.

Along with the *F1* scores for each point in 3-dimensional space of hyper-parameters, the target metrics *C%, S̃* and *CSF1* were evaluated with the same values of hyper-parameters. We also compute end evaluate composite metrics based on the primary three ones, such as additive *C%+S̃* or *C%+S̃+CSF1* and multiplicative *C%∗S̃* or *C%∗S̃∗CSF1*.

2.3 Training Sets and Model Building

For the purpose of model building, according to the referenced work [3], we have used public English, Russian and Chinese corpora - same as in the former paper.

For English, we have used Brown training corpus (6M bytes size) was downloaded from http://www.sls.hawaii.edu/bley-vroman/brown$_$nolines.txt .

For Russian, we have used RusAge corpus downloaded from https://www.kaggle.com/datasets/oldaandozerskaya/fiction-corpus-for-agebased-text-classification , having the largest one called Previews (825M bytes size) used for model building.

For Chinese, we have used CLUE Benchmark News 2016 dataset as downloaded from https://github.com/brightmart/nlp_chinese_corpus . When downloaded, the folder new2016zh was found to have two files, "news2016zh_valid.json" and "news2016zh_train.json". The latter larger file (8930M

bytes size) was processed programmatically (parsing JSON; selecting title, desc, and content fields; and saving each of the fields as individual lines), so plain text file was produced and used for model building.

The model building has been performed following exactly the same procedure as described in the referenced work [3] with three models for each of the languages obtained.

2.4 Test Sets

For the purpose of the *F1* score and *CSF1* metric determination, two test sets were involved for each of the languages.

First, there was test set of 100 text lines for every language obtained from parallel Chinese/English corpus of 100 multi-sentence statements related to personal finance was downloaded from Magic Data (https://magichub.com/datasets/chinese-english-parallel-corpus-finance) as a tab-delimited text file with individual columns for Chinese and English versions, entitled zh and en, respectively. The corpus has been extended with Russian translations of it it, with only one column entitled ru containing the Russian translations, as found in the file (https://github.com/aigents/pygents/blob/main/data/corpora/Russian/magicdata/zh_en_ru_100/CORPUS_ZH_EN_RU.txt) in the Aigents/Pygents open source project project.

Second, there was test set of 1000 text lines for every language randomly selected from entire text corpora for English, Russian and Chinese, respectively.

2.5 Experimental Approach Summary

In the course of our studies, we were able to obtain the functions $F1(N_s, T_{mc}, T_{mc})$, $C\%(N_s, T_{mc}, T_{mc})$, $\tilde{S}(N_s, T_{mc}, T_{mc})$ and $CSF1(N_s, T_{mc}, T_{mc})$ and explore correlations and correspondences between them for English, Russian, and Chinese languages, for test sets of 100 and 1000 text lines relying on models and pipelines described in the cited work [3], its appendices and code found in referenced Aigents/Pygents open source project (https://github.com/aigents/pygents/).

Also, while we were using the latter three functions of target metrics and their derivatives to find the combinations of hyper-parameters corresponding to them, and using these hyper-parameters to perform ”expectedly the best” tokenization, computing the *F1* score and associating the value of it with this metric.

At the same time, we were using all points in the three-dimensional space of hyper-parameters to perform tokenizations of the test sets with these parameters and compute the Pearson correlation values between the *F1* scores of such tokenizations and respective target and composite metrics computed based on them.

3 Experimental Results

The presented table 1 shows the Pearson correlations between all target metrics pairwise, for all explored combinations of hyper-parameters in the three-dimensional space of them. The table 2 presents Pearson correlations between the target metrics and some of the composite metrics based on them.

Table 1. Pearson correlations between target metrics for different languages, based on larger test set of 1000 lines.

Language	Metric1	Metric2	Correlation
English	$CSF1$	$\tilde{S}$	**0.90**
English	$CSF1$	$C\%$	**0.86**
English	$\tilde{S}$	$C\%$	**0.92**
Russian	$CSF1$	$\tilde{S}$	**0.76**
Russian	$CSF1$	$C\%$	**0.64**
Russian	$\tilde{S}$	$C\%$	**0.83**
Chinese	$CSF1$	$\tilde{S}$	**0.36**
Chinese	$CSF1$	$C\%$	**−0.31**
Chinese	$\tilde{S}$	$C\%$	**0.48**

It is seen that, according to the table 1, that $\tilde{S}$ has significantly positive correlations with both *C%* and *CSF1* across all languages. Also, all metrics are highly correlated for English and Russian, where the correlation of the $\tilde{S}$ with the other two is especially high. In turn, for Chinese, all correlations are generally lower while *CSF1* does not show correlation to any other metric.

Moreover, the table 2 is rendering highest correlations with *F1* score for composite *C%*+$\tilde{S}$ while other metrics perform better for some languages and worse for others, as discussed in the language-specific sections below.

3.1 English

In case of English we find it is possible to self-tune hyper-parameters based on almost any of the metric having the *CSF1,* $\tilde{S}$, and their composites especially good for the purpose, referring to the table 2, figure 1 and figure 2.

It should be noted that distributions of points on figure 1 and figure 2 corresponding to quite different test sets are pretty close and can be thought as a ”signature” describing specific language model.

Table 2. Pearson correlations between metrics and tokenization F1 score for different languages, based on larger test set of 1000 lines.

Language	Metric	Correlation
English	$CSF1$	$\mathbf{0.95}$
English	$C\%$	$\mathbf{0.83}$
English	$\tilde{S}$	$\mathbf{0.94}$
English	$C\%+\tilde{S}$	$\mathbf{0.93}$
English	$C\%*\tilde{S}$	$\mathbf{0.93}$
Russian	$CSF1$	$\mathbf{0.79}$
Russian	$C\%$	$\mathbf{0.56}$
Russian	$\tilde{S}$	$\mathbf{0.87}$
Russian	$C\%+\tilde{S}$	$\mathbf{0.67}$
Russian	$C\%*\tilde{S}$	$\mathbf{0.68}$
Chinese	$CSF1$	$\mathbf{-0.09}$
Chinese	$C\%$	$\mathbf{0.85}$
Chinese	$\tilde{S}$	$\mathbf{0.79}$
Chinese	$C\%+\tilde{S}$	$\mathbf{0.95}$
Chinese	$C\%*\tilde{S}$	$\mathbf{0.83}$

3.2 Russian

For Russian we find it substantially less reliable but still possible, relying on *CSF1,* $\tilde{S}$, and their composites (as in case of English), referring to the table 2, figure 3 and figure 4.

As in case of English, distributions of points on figure 3 and figure 4 corresponding to different test sets are still quite similar pretending to serve as a ”signature” of the language model.

3.3 Chinese

For Chinese results, given the same size of the test set, are much more vague than for English and Russian. However, blending the *C%* with $\tilde{S}$ makes automatic determination of hyper-parameters still possible, referring to the table 2, and figure 5.

It should be noted that using the test set of 1000 lines did not provide sufficiently reasonable result for Chinese at all and has not been presented in this paper.

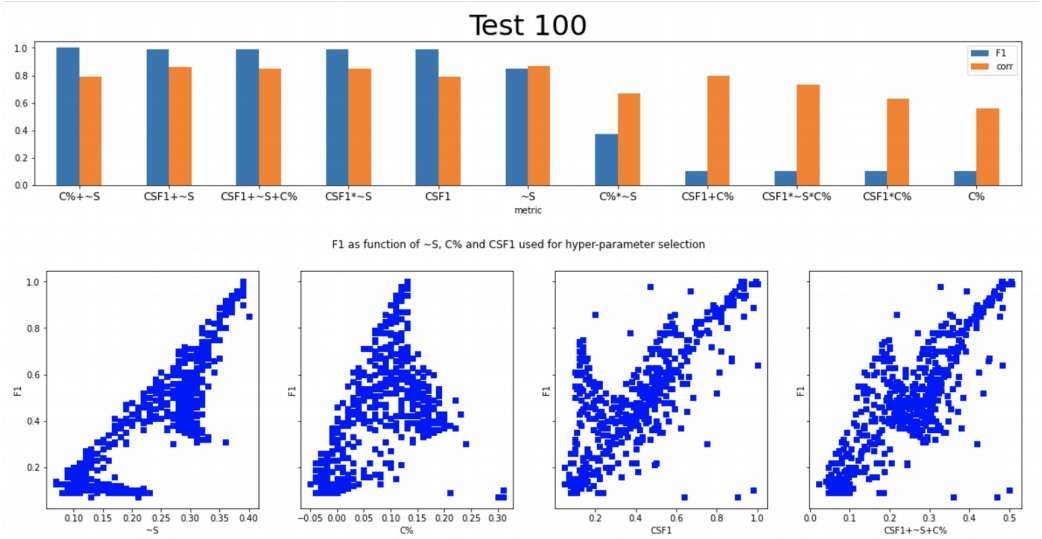


Fig.3. For Russian language and test set of 100 lines. Upper row: blue bars correspond to the F 1 score obtained based on set of hyper-parameters corresponding to the maximum target metric on horizontal axis, blue bars correspond to Pearson correlation between the F1 score and respective metric.

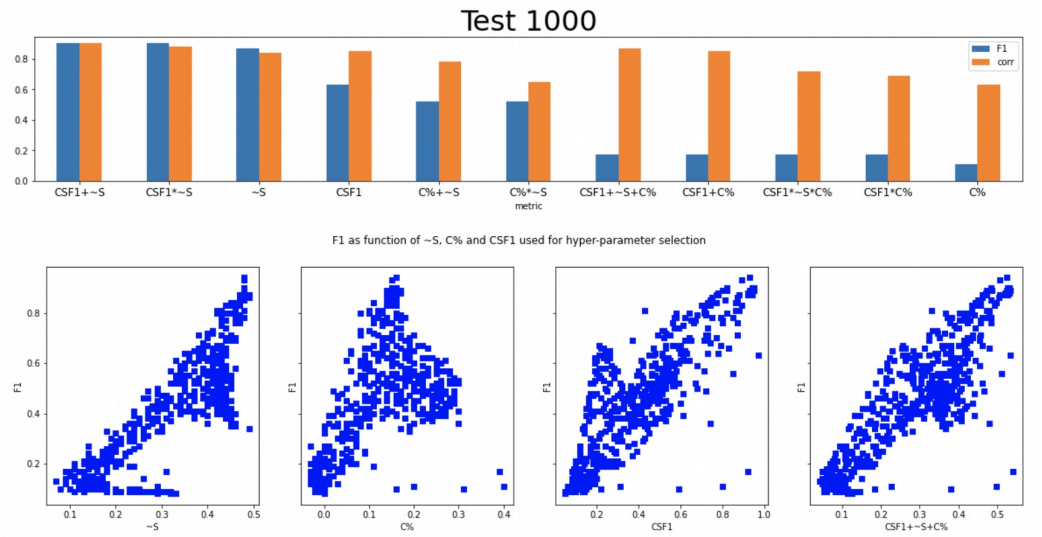


Fig.4. For Russian language and test set of 1000 lines. Upper row: blue bars correspond to the *F1* score obtained based on set of hyper-parameters corresponding to the maximum target metric on horizontal axis, blue bars correspond to Pearson correlation between the F1 score and respective metric.

4 Conclusion

We have discovered a fairly good mutual Pearson correlations between all three introduced human-agnostic metrics such as ”compression factor”, ”normalized anti-entropy” and so-called ”cross-split *F1* score” for English and Russian languages, in the space of hyper-parameters of fully-unsupervised tokenization of raw natural language texts. Also, we have found high Person correlations between these parameters and their composites against accuracy of unsupervised tokenization process performed with respective set of hyper-parameters.

In the case of Chinese language, we find a significant correlation between the *F1* score and the ”compression factor” and ”normalized anti-entropy” as well as composites of the two.

Our results suggest the possibility of robust unsupervised tokenization of low-resource and dead languages and allow us to think about human languages in terms of the evolution of efficient symbolic communication codes with different structural optimization schemes that have evolved in different human cultures.

At the same time it opens the way for end-to-end unsupervised language learning from unannotated and unsegmented corpora as it has been suggested by Vepstas and Goertzel [9] and further developed by Glushchenko et. al. [10].

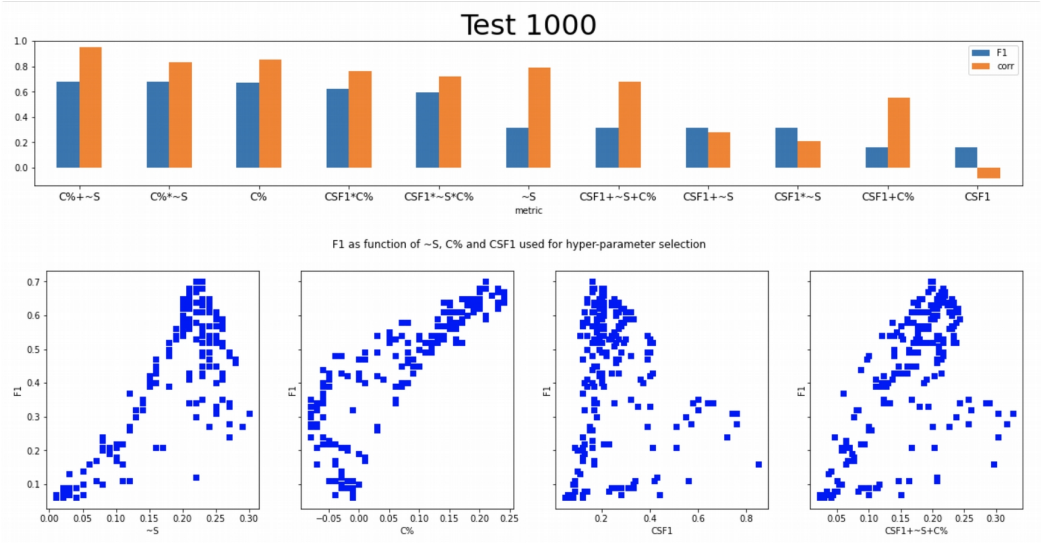


Fig.5. For Chinese language and test set of 1000 lines. Upper row: blue bars correspond to the *F1* score obtained based on set of hyper-parameters corresponding to the maximum target metric on horizontal axis, blue bars correspond to Pearson correlation between the *F1* score and respective metric.

5 Limitations

The major limitation of this work appears to be that the corpora used for the three languages are way different in terms of size and topic. The English and Russian corpora are literary while the Chinese one is based on crawled news data. Further, the English corpus is 100 times smaller than the Russian one while the Chinese corpus is 10 times larger then the latter. Each of these facts or both of them might serve as an explanation of better results for English compared to Russian and Russian compared to English.

In our further work we are going to re-run the same experiment based on models built upon parallel multi-lingual corpora on the same topic and of the same size across all languages involved in the experiment.

The other direction of further work might be running the same experiment for each of the languages with training corpora of different size to see if the size of the corpora might be another hyper-parameter to drive the performance of the hyper-parameter self-tuning.

6 Ethics Statement

There are no apparent negative ethical implications associated with this work. We rather suggest that our work has a positive ethical benefit as it promotes the inclusiveness of cultures based on so-called ”low-resource” languages and dialects that are not easily studied using current linguistic approaches. We believe that the proposed technology can facilitate the learning of such languages by providing initial lexical vocabularies based on raw field data and thus paving the way for further learning of these languages and their grammars. Another long-term positive ethical impact comes from the ”interpretable” nature of this work. Our model facilitates the transition to open, transparent and human-friendly linguistic models that can be developed for any human language and put into production, thereby preventing the potential degradation of the quality of life of ”black box” NLP models. No harmful social impact from our work is anticipated.

7 Acknowledgments

We are grateful to Sergey Terekhov and Nikolay Mikhaylovskiy for valuable questions, critique, recommendations and suggestions during the course of work.

The paper preprint is available as https://arxiv.org/pdf/2303.02427.pdf .

References

[1] Logan Kearsley. A hybrid approach to cross-linguistic tokenization: Morphology with statistics. In Brigham Young University, Theses and Dissertations, 5984. 2016.

[2] Jesse O. Wrenn, Peter D. Stetson, and Stephen B. Johnson. An unsupervised machine learning approach to segmentation of clinician-entered free text. In Proceedings of the AMIA Annual Symposium, pages 811–5. 2007.

[3] Anton Kolonin and Vignav Ramesh. Unsupervised tokenization learning. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, page 3649–3664. 2022.

[4] Simon Kirby, Monica Tamariz, Hannah Cornish, and Kenny Smith. Compression and communication in the cultural evolution of linguistic structure. Cognition, 141:87–102. 2015.

[5] Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Entropy minimization in emergent languages. arXiv:1905.13687 [cs.CL]. 2019.

[6] John R. Searle. Language as Biological and Social. In Making the Social World: The Structure of Human Civilization. Oxford University Press. 2010.

[7] Abdurrahman Usman, Adi Mahmud, Abdulhalim Daud, and Suratman Dahlan. Language as a social instrument. Edukasi, 18:259–276. 2022.

[8] Georges Hansel, Dominique Perrin, and Imre Simon. Compression and entropy. In STACS 92, pages 513–528, Berlin, Heidelberg. Springer Berlin Heidelberg. 1992.

[9] Linas Vepstas and Ben Goertzel. Learning language from a large (unannotated) corpus. 2014. arXiv:1401.3372 [cs.CL]. 2014.

[10] Alex Glushchenko, Andres Suarez, Anton Kolonin, Ben Goertzel, and Oleg Baskov. Programmatic link grammar induction for unsupervised language learning. In Artificial General Intelligence, pages 111–120, Cham. Springer International Publishing. 2019.

A brief recollection of the Multi-Level Model, and some of its properties which distinguish it from the Standard Model

Levichev A.V.[1,2], Palyanov A.Yu.[2,3]

[1]*Sobolev Institute of Mathematics,acad. Koptyug pr. 4, Novosibirsk, 630090, Russia.*
[2]*Ershov Institute of Informatics Systems, acad. Lavrentiev pr. 6, Novosibirsk, 630090, Russia.*
[3]*Novosibirsk State University, Pirogova str. 2, Novosibirsk, 630090, Russia.*

alevichev@gmail.com, palyanov@iis.nsk.su

Abstract: *In the paper we discuss a few key points of the Multi-Level Model (=MLM) which was suggested by A. Levichev as a mathematical description of quarks and gluons. Major part of the necessary MLM-terminology is recalled while certain MLM-findings (for quarks and leptons) are compared to the theoretical and experimental data as accepted by the Standard Model (=SM). Segal's Compact Cosmos (or space-time)* ***D*** *is a Lie group U(2) with causal structure determined by an invariant Lorentzian form on the Lie algebra u(2). MLM is based on* ***D*** *and on the sequence of embeddings: U(2) into U(3), U(2) into U(4), etc. These groups were called the levels (of matter): U(2) - the 0th (that is, our mundane world), U(3) - the 1st, U(4) - the 2nd, etc. Such a convention matches the SM-quarks' generations list. Each SM-quark is interpreted as either a sunken proton or as a captured proton. The MLM-proton is elementary and indestructible (hence no need for confinement). For MLM-quarks, in terms of the introduced notion of a ruling group, flavor and color are defined mathematically. The number of colors (and of flavors) is level-dependent. For levels U(3) through U(6) the correspondence with the SM-quarks is established. Three new quarks are predicted. The SM-puzzle of quarks' and leptons' generations is solved. Using the Han-Nambu scheme, the notion of quark's electric charge is expressed in terms of color charges. Certain proton's properties are suggested to study on the basis of its wave function. Quark matter at different levels is modeled on the basis of the captured proton notion.*

Keywords: Segal's Compact Cosmos; Standard and Multi-Level Models; Wigner-Segal method; ruling group; chronometric proton's wave function

1 Introduction

In [1], with a goal of understanding the quark-gluon media, the Multi-Level Model (MLM, for short) was suggested. As regards MLM-gluons, there was no any development since they have been defined in [1, 2], while the current paper is mostly dedicated to fermions (including quarks, leptons, and electrons). Below, starting with the next section, MLM's main tenets and terminology are recalled (mostly from [1, 2]). To mathematically model elementary particles, Segal's Chronometric Theory is applied (see, especially, the section *Indecomposable Elementary Particle Associations* in [3] which is in some sense a summary of Segal's findings. The method of [3] can be interpreted as the one generalizing the suggestion by Wigner [4] to model particles on the basis of certain representations of the *Poincare group* (denote this 10-dimensional Lie group by *P*). In [3] Segal treats certain representations of $G = SU(2,2)$. This latter Lie group (sometimes called the *conformal* one) can be defined as follows. Denote by *S* the diagonal 4 by 4 matrix with diagonal entries 1, 1, -1, -1. The group *G* consists of all 4 by 4 matrices *g* (with complex entries allowed) having determinant 1 and satisfying the condition $g^*Sg = S$. *G* contains both the (11-dimensional) *extended Poincare group* P^+ and its subgroup *P*. The Chronometry is fundamentally based on the (famous!) fractional-linear *G*-action on the unitary group $U(2)$. Here we only mention that the universal cover of **D** is, essentially,

Einstein's static universe. In [5] it was suggested to name the approach of [3], on how to model elementary particles, as the *Wigner-Segal method*.

The abbreviation SM stands for the (currently widely accepted) Standard Model. In [1, 2] the MLM was described as an alternative to the SM. On the basis of further studies (see [5–10]), **MLM** can now be viewed **as the marriage of Segal's Chronometry with SM**. One of the main goals now is to put together concise and, hopefully, convincing arguments in order to support such a far-reaching view. To finish the Introduction, let us mention that the current paper is significantly based on: Levichev, A.; Palyanov, A. The Multi-Level Model for Quarks and Leptons as the Symbiosis of Segal's Chronometry with the Standard Model. *Preprints* **2022**, 2022020280 (doi: 10.20944/preprints202202.0280.v2).

2. From Chronometry to the MLM-quarks of the $U(3)$-level

First, the Segal's Chronometry has to be outlined. Recall that A. Levichev has surveyed it in [11] (in Russian). Also, it is surveyed in [12] which is online. Mathematically, Chronometry deals with a slightly larger totality of space-time events than the Minkowski space-time M has. Namely, the *compact chronometric world* **D** (or the *Segal's compact cosmos*), as a manifold, is the unitary group $U(2)$ which is defined by the (standard) formula (3.3) of [9]. Here (and on) we use the world as a synonym of *space-time*. It is well-known that **D** is a natural alternative to M, the unique four-dimensional manifold with comparable properties of causality and symmetry. Free particles are considered to be associated with positive-energy representations in bundles of prescribed spin over **D** of the group of causality-preserving transformations, and with corresponding wave equations.
The embedding of M into **D** via the Caley transform is well-known, see formula (5.2) of [12]. The Lorentzian metric (or inner product) in **D** was introduced by Segal (and is given in Section 3.1 of [12]). This metric is left-invariant as well as right-invariant on the Lie group $U(2)$. The above-mentioned Poincare group P is the totality of all isometries of the (pseudo-Euclidian) M. The group P^+ can be obtained from P by adding scaling transformations; hence P is a subgroup of P^+.

The main group of (causal) transformations in **D** is (15-dimensional) $SU(2,2)$, see formula (2.2) from [2]; in our Figure A5 we choose a generic element g_2 of $SU(2,2)$ and we reproduce its action on a generic element z of $\mathbf{D}=U(2)$. It is the linear-fractional action, see Figure A5. When one switches to (an earlier mentioned) **D**'s universal cover, one has also to switch from $SU(2,2)$ to its universal cover. In this regard we only mention that there is a canonical mathematical way of treating such a situation: when G acts on U, then there is a canonical action of the G's universal cover on the U's universal cover. In our paper, it is enough to stay with $G = SU(2,2)$ and with its linear-fractional action on **D**.

Certain representations (see [3]) of $SU(2,2)$ give rise to *chronometric particles* but we notice, once and for all, that we only need to deal with just one particular representation: it is introduced right before our Theorem 2, below. The adjective *chronometric* indicates that we are within the Chronometry. The particles suggested by this theory should be compared to *relativistic* particles, where the (10-dimensional) Poincare group P is the main symmetry group, or to *Galilean* particles, when (also 10-dimensional) Galilean group is the main one. For our research it is important to remember that the SM deals with relativistic particles while the MLM, since it uses many findings of Segal's School, treats chronometric particles, essentially. In [2] (Section 2) it was outlined how a property (one or another) of a chronometric particle can be interpreted in relativistic terms.

For more details on Chronometry a reader is advised to visit Section 2 of [2] or suitable references therein. However, on the basis of the findings in [10], we point out that certain corrections (of what Segal claimed in [3], and what was reproduced from [3] in Section 2 of [2]) have to be made. The research [10] can be viewed as a discussion of, and supplement to, Segal's list of chronometric elementary particles of spin 1/2 [3]. Let us stress that when we, below, associate certain mathematical objects with specific particles; we try (as much as possible) to stay in line with what Segal has done before in this regard. Here is the only significant exception to the above: around 2010 A. Levichev suggested [11] (Section 7) that it is rather the proton p than a hypothetical neutral particle, the exon.

Meanwhile we only state the following (it is an extract from [10]): there is a certain (infinite-dimensional) Hilbert space F_p (of functions on the Minkowski space-time M) which is interpreted as the set of all (theoretically possible) *states* (or *wave functions*) of the chronometric proton p (see formula (20) of [10] and Theorem 3.1 of [9]). (More details on the space F_p are given in our Subsection 6.3). The group $G = SU(2,2)$ acts on F_p. According to the terminology of [9] (Remark 4.1), this G is the *ruling group*, G_r, since it *rules* (or *governs*) the behavior of the particle (the latter thus being the proton p in our case). We are now in a position to start a description of the MLM. This model plays with the sequence of *canonical* (that is, based on principal minors of the matrices involved) embeddings of groups: $U(2)$ into $U(3)$, $U(2)$ into $U(4)$, $U(2)$ into $U(5)$, etc. Each of these embeddings is an isomorphism onto the corresponding subgroup which (later) plays an essential role on how to relate our MLM-quarks to the SM-quarks (right now, for a quicker understanding of the MLM, a reader is advised to look at Figure A1 of the Appendix A). The above groups were called *levels*: $U(2)$

is the 0^{th} level, $U(3)$ – the 1^{st}, $U(4)$ – the 2^{nd}, etc. In our Section 3 we will see that such a convention matches the standard quarks' generations' list. Recall that each matrix group $U(n)$ is defined quite similarly to how $U(2)$ was defined by formula (3.3) of [9]. Embeddings A_{12}, A_{13}, A_{23} of (the Segal's compact cosmos) $\mathbf{D} = U(2)$ into $U(3)$ were introduced as follows: under each of these three embeddings a matrix Z from D becomes a certain principal 2 by 2 minor of the corresponding 3 by 3 matrix from $U(3)$. Namely, let us denote by D_{12} the image of the embedding A_{12} where A_{12}, itself, is defined as follows (also, these embeddings are illustrated in the top portion of Figure A1):

Each Z from **D** is now an upper 2 by 2 principal minor of the 3 by 3 matrix $A_{12}(Z)$ in $U(3)$; the third diagonal entry of $A_{12}(Z)$ is 1; in the $A_{12}(Z)$, all other entries vanish. The two remaining embeddings, A_{13} and A_{23} are defined quite similarly. Clearly, D_{12}, D_{13}, and D_{23} are $U(2)$–subgroups in $U(3)$. Recall that the group $U(2)$ is closed w.r.t. the complex conjugation, and w.r.t. the matrix transposition. The transposed matrix Z^T can be viewed as a mirror of Z w.r.t. reflection in the principal diagonal. From this it follows that each of the D_{12}, D_{13}, D_{23} is invariant w.r.t. any of the two mentioned operations in $U(3)$. Also, to enumerate all D_{ij}, it is enough to consider the cases $i< j$, only. We supply each D_{ij} with a Lorentzian metric by the demand that each A_{ij} be an isometry.

In the totality of all m by m matrices, introduce P_m, as the symmetry w.r.t. the opposite diagonal. Clearly, when Z is in $U(2)$, then $P_2(Z)$ is also in $U(2)$. From this it follows that the subgroup D_{13} is P_3-invariant in $U(3)$ while $P_3(D_{12})=D_{23}$, $P_3(D_{23})=D_{12}$. That enables us to view the embeddings A_{12} and A_{23} as equivalent (one becomes the other when composed with P_3 and P_2 – see Figure A1). **This relates to the 'presence of two SM-*u*-quarks in' a proton, while the A_{13} relates to the presence of an SM-*d*-quark in that proton**. These embeddings make it possible to introduce a notion of a flavor of a MLM-quarks of level $U(3)$. The last two phrases are 'non-mathematical', we relate to physics here. We discuss this in more detail later in this section (right after Theorem 1).

Definition 1. Below, in many instances, we use the name **cell** for each of these D_{ij}.

This seems to be a reasonable way of attaching names: compare to such a straightforward (but the one which is less 'compact') alternative as of "a minor (one or another) of the (corresponding) matrix of a certain size". It should always be stated (or should be clear from the context) which $U(n)$-level such a cell is considered to be in.

In [2] (Section 3), similarly to the way of introducing (in $U(3)$) of the $U(2)$-subgroups D_{12}, D_{13}, D_{23}; the $SU(2,2)$-subgroups G_{12}, G_{13}, and G_{23} of $SU(3,3)$ have been defined. A reader is now advised to go to our Section 4 where (right after Table 1) we define G_{12}, G_{13} at the $U(3)$-level. Namely, a generic element (of each of the two subgroups) is explicitly shown at Fig. A5. At each level $U(n)$, any subgroup G_{ij} is defined in our Section 4 (right before Proposition 2 there) - quite similarly to how it has been done for the $U(3)$-level.

Definition 2. Below, in many instances, we use the name **ruling group**, or the ***r*-group**, for each of these G_{ij} (the latter being an $SU(2,2)$-subgroup of $SU(n,n)$, in general). It should always be stated (or should be clear from the context) which $U(n)$-level such an r-group is associated with. From [5], we now reproduce the following statement [5] (Theorem 4.1).

Theorem 1. *An action of each of the subgroups G_{12}, G_{13}, and G_{23} on any of the subgroups D_{12}, D_{13} and D_{23} is defined. In particular, each of the following three actions, G_{12} on D_{12}, G_{13} on D_{13}, and G_{23} on D_{23}, is the linear-fractional one.*

Let us now say more about the mathematical meaning (and about the physical interpretation) of the following phrase (from above): "The embeddings A_{12} and A_{23} relate to the presence of two SM-*u*-quarks in a proton, while the A_{13} relates to the presence of an SM-*d*-quark in that proton".

First, let us recall that (according to the SM) a proton consists of two *u*-quarks and one *d*-quark. The detection of three point-like centers (of highly inelastic electron-proton scattering, see [13]) served as an experimental basis for such a conclusion about the structure of a proton. However, after several decades of intense search, the majority of the Physics community has submitted to the view that "free quarks cannot be detected."

Before we introduce the MLM-quarks in $U(3)$ (and later, with the increase of n, in each of the levels $U(n)$), we need to present more mathematical findings (from [9, 10], mostly) related to Chronometry. Above, closer to the beginning of the Section, we have been discussing chronometric particles of spin ½. They originate from an induced representation of the group $G = SU(2,2)$ defined by formula (5) from Section 2 of [10]. In our current paper (starting with the next page, and in Subsection 6.3, and elsewhere throughout the paper), we provide more details about this representation. In [10] (Section 9), it was concluded that exactly three chronometric spin ½ fermions have been mathematically detected (and interpreted as the proton p, electronic neutrino υ_e, and electron e). It was mentioned in [10] (right before Remark 1.1 there) that such a conclusion did not involve the MLM, per se. On the basis of formula (16) from [10] (Section 4), the space F of the induced representation (sometimes referred to as the *spannor representation*) of the group $G = SU(2,2)$ has been introduced. In F, there exist two non-trivial invariant subspaces with no invariant complement. One of those

subspaces was denoted by F_p and it was supplied with a Hilbert space structure. It has been proven (see Section 5 of [10]):

Theorem 2. *The restriction of the induced representation to F_p is unitary and irreducible. In F_p, the energy-positivity condition holds.*

The space F_p has been interpreted as the totality of all (theoretically possible) wave functions of the chronometric proton. Notice that F_p is of *conformal weight* (or *conformal dimension*) 5/2, see [10] (Section 5).

Remark 1. Let us repeat that in [10] (Section 6), it was stated that F_p does not have an invariant complement. It means that we deal with the case where the Wigner method is not applicable. According to the Wigner-Segal method, we now have to deal with the quotient space $W = F/F_p$ and with the factor-representation in it. In [10] (Section 6), a minimal non-trivial invariant subspace F_υ in W has been introduced, F_υ has been supplied with the unitary structure and it has been interpreted as the totality of all wave functions of the chronometric electronic neutrino. In [10] (Section 7), the quotient space $W/F_\upsilon = F_e$ has been interpreted as the totality of all wave functions of the chronometric electron (since the corresponding factor-representation turned out to be irreducible and unitarizable, and the conformal weight is 3/2 now). This was the final step in the proof of the main finding of [10]: **there are exactly three elementary chronometric spin ½ particles**.

Remark 2. Having in mind Remark 9.1 of [10], from now and on it seems plausible to associate the Hilbert space F_υ with the electronic antineutrino, rather than with electronic neutrino. Let us recall that the authors of [10] followed Segal (see [3]) and they interpreted this ('middle') sector of the 3-step composition series as the one corresponding to the electronic neutrino but (also in [10]) they have envisaged a possibility of the antineutrino interpretation. Such an interpretation can serve as a mathematical reason allowing a return (as it has been claimed in [14]) to an 'old model' for the neutron as consisting of a proton, of an electron, and of an electronic anti-neutrino – see the $U(2)$-part of the Figure A4 (in our Appendix A).

For the level $U(3)$, recall that each A_{ij} is an isometry, and that each D_{ij} is a space-time isometric to the Segal's compact cosmos **D**. From here (and on the basis of both the above Theorem 2 and of its interpretation), we conclude that a spin ½ fermion ('living in' D_{ij}) is mathematically defined. If D_{12} or D_{23} is a support of its wave functions then, as part of the MLM, we associate this fermion with an *u*-quark. If D_{13} is such a support – then the particle is announced to be a *d*-quark. It means that in the MLM we have introduced two flavors for quarks of the 1st level, $U(3)$, and (by merely keeping the SM-terminology) we have established the correspondence of MLM-quarks to the SM-quarks, and vice versa. It is assumed that a Hilbert structure in the corresponding spaces H_{ij} is introduced via the isometries A_{ij} from the original $H = F_p$. For each of our fermions of the level $U(3)$, its *r*-group G (recall that '*r*' here is for 'ruling', from Definition 2, above) can be any of G_{12}, G_{13}, or G_{23}. Such a convention was a mathematical basis for defining the notion of a *color* of an MLM-quark (see Sections 3 and 4 of [5], as well as our Section 5, below). Clearly, each MLM-quark is as fully described, as the chronometric proton was – see our Theorem 2, above. One is thus tempted to think of an MLM-quark "as if" being a state of the (chronometric) proton.

Remark 3. Our "as if" cannot be dropped here; at least (if dropped), additional explanation should be provided. Here is why: a proton is described by one pair (H, G) while an MLM-quark – by another. In this regard, let us recall that the SM allows usage of different (however, isometric, only) Hilbert spaces of wave functions for the same particle. Hence, let us recall that the following holds: the two *r*-groups (one for the proton, another for the MLM-quark) consist of matrices of different sizes; in the first case – those are 4 by 4 matrices (since our proton lives in $U(2) = \mathbf{D}$), while the *r*-group G (for the quark) is a subgroup of $SU(3,3)$. The latter means that the corresponding matrices are 6 by 6 ones. We have thus explained why, formally, we should not view MLM-quarks as states of a proton. For a pair of different MLM-quarks (of the same MLM-level), the respective *r*-groups (consisting of the matrices of the same size), are, however, different subgroups of $SU(3,3)$ – or of $SU(n,n)$, if $U(n)$ is the level in question. To finalize the current remark, we would like to stress that the main reason to not consider a quark as the state of a proton is that the two particles 'live' in different MLM-levels.

Rather than to use 'a state of a proton' terminology for an MLM-quark, let us *word* the process (of highly inelastic electron-proton scattering) in question as follows. As the result of such a scattering, a proton gets (from $U(2)$) to a 'deeper' level, $U(3)$. In $U(3)$ it gets to one particular cell (of the total of three available ones: D_{12}, D_{13}, or D_{23}) thus becoming an MLM-quark. Recall that the notion of a cell has been mathematically introduced above, in Definition 1. In terms of Physics, a possible description (see our Subsection 6.4, below) could use the following wording: "our proton pushes 'deeper' (that is, to the $U(4)$-level) the 'former occupant' of this cell. However, to better understand such a wording, it is recommended to read our Section 5, first.

Since we have introduced equivalence (w.r.t. operators P_3 and P_2 - see Figure A1 of our Appendix A) between D_{12} and D_{23}, we only have two flavors rather than three. The word combinations, 'gets to one particular cell' and 'becomes an MLM-quark', merely mean that this stage

of scattering has to be described in terms of the Multi-Level Model rather than using the SM-language. In a formal agreement with the SM, our description is that of elastic scattering (of the original electrons hitting the proton) on quarks (in our case – on MLM-quarks). In particular,within the $U(3)$ level, we should use another Hilbert space and another r-group. These last two issues might be less, or even not at all, significant in comparison to a change of state of our proton as it becomes an MLM-quark. Such a change seems to be an unavoidable one.

A reader is now advised to go to Figures B1, B2, and B3 (of our Appendix B) in order to get a clue on what a state of a proton may look like. Seemingly, the thus suggested description of the highly inelastic electron-proton scattering does not, per se, contradict the detection of 'three point-like components in a proton'. Also, in the MLM-approach one can directly apply the combinatorial SM-methods to calculate relations between certain scattering probabilities. A word of caution: in our current paper we discuss the 'sunken vs captured' proton issue (see Section 5). This develops the earlier (see [2]) notion of a 'captured proton'.

Let us provide more MLM-details. If our proton gets into the cell D_{12}, then (instead of the former F_p) we have to exploit the space F_{12} of wave functions defined on D_{12}, rather than on the original **D**.

Let us recall that due to an isometry A_{12} between **D** and D_{12}, the Hilbert spaces F_p and F_{12} are unitarily equivalent. In [5, Section 3] the notation $q(1;1,2)$ has been used for such an MLM-quark: the first '1' is the level number (here it means the $U(3)$ level), while the pair (1,2) specifies the cell (being D_{12} here).

3. The MLM-quarks of levels $U(4)$ and $U(5)$

Introduce embeddings of $\mathbf{D}=U(2)$ into $U(4)$, first. Here they are (see Figure A2 of Appendix A): A_{12}, A_{13}, A_{14}, A_{23}, A_{24}, A_{34}; the notation mimics the one which has been already used in the (3)-case. To determine equivalences, consider the (applicable to any 4 by 4 matrix) operator P_4: the symmetry w.r.t. the opposite diagonal. Clearly (as the Figure A2 illustrates), A_{12} is equivalent to A_{34}, and A_{13} is equivalent to A_{24}. Each of the subgroups D_{14} and D_{23} is P_4–invariant. Relate A_{14} to an SM-quark s, and A_{23} – to an SM-quark c. At this (that is, at the second) level, A_{12} (which is equivalent to A_{34}) relates to an SM-quark u while A_{13} (equivalent to A_{24}) relates to an SM-quark d. Hence, SM-quarks of both *generations* (one and two) 'live' on the 2nd MLM-level, $U(4)$.

Now, for an interested reader (even if this person does not have much knowledge about currently known elementary particles), we give an overview of SM-quarks' generations, as well as of the topic in general. Namely, by 2018 it was known that there exist (at least) three generations of quarks and three generations of leptons. These fundamental particles are thought to be adequately modeled as the 'point-like' ones. Both quarks and leptons have spin ½ which means that they are fermions. By convention, a *fermion* is a particle with half-integer spin, while a *boson* is a particle with an integer spin. Mathematically, a spin of a particle is a certain constant which is present in the (describing this very particle) representation of the group G – see, i.e. [15, p. 348]. By G here (as well as above), we mean the main group of transformations acting in the space-time which our particle 'lives in'. The word *generation* is part of the SM-terminology. (As it should be already clear, generations, essentially, are naturally 'built into' the MLM.)

According to SM, the first generation of quarks consists of u and d; the first generation of leptons consists of the electronic neutrino υ_e, and of the electron e. The second generation: the quarks c and s; the leptons – muonic neutrino υ_μ and the muon μ. The third generation: the quarks t and b; the leptons - tau neutrino υ_τ and the tau particle τ. Also, for each particle, there is an antiparticle. The SM-quarks are considered to have *color* and to have (fractional) *electric charge*. (Within the MLM, we will discuss both notions (color and electric charge) below, see our Section 4). The SM-leptons (we have just listed all of them) are colorless particles having charge zero - as regards neutrinos, and charge 1 (or negative 1) - as regards the remaining leptons. Matter is built of atoms which are believed to be built of the first generation fermions. Within the SM, it is not quite clear why Nature needs the 2nd and the 3d generation fermions. As it will be seen below, the MLM subdivides the above fermions into generations by its very structure. It is very difficult to create and to detect fermions of generations 2 and 3: powerful charged particles' accelerators are to be used for that. Besides, the created particles decay almost instantly. Within generations two and three, only neutrinos are believed to have an unlimited lifetime. Currently the search for the 4th generation of particles is going on (our Figure A3 indicates 'where' to look for quarks of the 4th generation).

Part of the above presented description of fermions and of their role as seen by the SM is a (translated from Russian) portion of

http://phys.vspu.ac.ru/forstudents/TSOR/Kutseva/pokolenie_leptonov_i_kvarkov.html.

We now return to the MLM-description of quarks. Here is the list of all $\mathbf{D}=U(2)$ embeddings into $U(5)$: A_{12}, A_{13}, A_{14}, A_{15}, A_{23}, A_{24}, A_{25}, A_{34}, A_{35}, A_{45}. To get a quick grasp of the picture in

question, our reader can easily think of an illustration, similar to Figure A2. Clearly, $P_5(D_{12}) = D_{45}$; we relate it to an SM-*u*-quark. $P_5(D_{13}) = D_{35}$, we relate it to an SM-*d*-quark. $P_5(D_{14}) = D_{25}$, we relate it to an SM-*s*-quark. $P_5(D_{23}) = D_{34}$, we relate it to an SM-*c*-quark. Each of the following two subgroups, D_{15} (*t*-quark) and D_{24} (*b*-quark), is P_5–invariant. Hence, 'MLM-copies' of the SM-quarks of all three generations 'live' on the 3[d] MLM-level *U*(5).

From now and till the end of Section 4, let us write the word 'quark' everywhere instead of the earlier used 'MLM-quark'. Recall (from [2, 5]) the notation *q*(n; i, j) for the corresponding quark of the *n*-th level; one can always assume that i<j. Such a convention results in the following relations between MLM- and SM-quarks:

q(1; 1, 2) = *q*(1; 2, 3) = *u*, *q*(1; 1, 3) = *d*; *q*(2; 1, 2) = *q*(2; 3, 4) = *u*, *q*(2; 1, 3) = *q*(2; 2, 4) = *d*,
q(2; 2, 3) = *c*, *q*(2; 1, 4) = *s*; *q*(3; 1, 2) = *q*(3; 4, 5) = *u*, *q*(3; 1, 3) = *q*(3; 3, 5) = *d*,
q(3; 2, 3) = *q*(3; 3, 4) = *c*, *q*(3; 1, 4) = *q*(3; 2, 5) = *s*, *q*(3; 2, 4) = *b*, *q*(3; 1, 5) = *t*, "top quark".

Recall that '2' (in, say, *q*(2; 1, 4)) is the number of the level, while the pair (1,4) indicates the cell D_{14}. Also, compare to an example at the end of Section 2 and see Figure A3.

Remark 4. The SM-quarks of higher generations were detected later (than SM-quarks of lower generations), with the increase of the accelerators' typical energies. Hence, it is natural to interpret the 'deepening', with the increase of *n*, of the *U*(n)-levels as corresponding to the increase of the scattering typical energy.

As regards the total number of MLM-quarks at a given MLM-level, we reproduce (from [1, 2]) the following

Theorem 3. *On the level U*(n), *suppose an U*(2)*-subgroup* D_{ij} *be not* P_n*-invariant. Then* D_{ij} *corresponds to a quark from a lower level. The recurrent* (1) *and explicit* (2) *formulas* (*for the total number* m_n *of quarks at the U*(n)-level) *hold*:

$$m_2 = 1,\ m_n = m_{n-1} + [n/2], \quad (1)$$
$$m_n = \{n(n-1)/2 + [n/2]\}/2. \quad (2)$$

Here, [x] denotes the greatest integer part (roof) of a real number x. We are thankful to the Reviewer who has noticed that (2) can be simpler expressed as $[n^2/4]$.

To conclude our Section 3, we indicate that our 'selection criterion' (that is, why we are quite satisfied with our list of MLM-quarks) is the one which establishes an explicit correspondence with the SM-quarks. In levels *U*(3), *U*(4), *U*(5) the MLM-quarks are in precise correspondence with SM-quarks as they are currently agreed upon. In level *U*(6), three new SM-quarks are predicted, as illustrated in our Figure A3.

4. On fractional electric charges of SM-quarks and the Han-Nambu scheme

Most part of this Section's content is a translation of (certain) portions of [5].
According to the Standard Model, the electric charge of each quark *u* (or *c*, or *t*) is 2/3, while the electric charge of each quark *d* (or *s*, or *b*) is minus 1/3. There is an approach with integer quarks' charges, [16]. Such an approach is known as the **Han-Nambu scheme** (in [16] there is a reference to the original Han-Nambu publications). Our current section reproduces (from [5]) the adaptation of the Han-Nambu scheme to the MLM.

Consider the *U*(3)-level, first. Let us insert our notation into the following Table (which we reproduce from p.1 of [16]).

Table 1.

f \ c	**12**	**23**	**13**
u: 12, 23	1	1	0
d: 13	0	0	-1

In the first row of this Table, **f** is for flavor, that is, *u* or *d*; **c** is for color (not to be confused with the notation for the quark *c*!). The last three entries in the first row are abbreviations for G_{12}, G_{23}, and G_{13}. This notation for corresponding *SU*(2,2)-subgroups of *SU*(3,3) was introduced in our Section 2; however, a 'shorter' one, **12**, **23**, **13** should not be confusing either. Similarly (in the beginning of the Table's second row) we use 12, 23 instead of D_{12}, D_{23}, etc.

Before we state a possible 'color-related' interpretation of the above Table 1, let us provide more details (from [5], mostly) on how the subgroups G_{ij} have been introduced.

It is well-known that each matrix g_n (in $G_n = SU(n,n)$) is composed from *n* by *n* blocks A_n, B_n, C_n, D_n. Also, there is a fundamental (linear-fractional) G_n-action on *U*(n). For the *U*(3)-level, recall

(from our Section 2) the embedding A_{12}. It designates the rows 1 and 2, and it designates the columns 1 and 2. Such a designation (choice) of rows and columns uniquely specifies an $SU(2,2)$-subgroup G_{12} in G_3. Namely, each g_{12} in G_{12} is composed (from blocks A_2, B_2, C_2, D_2 of the original matrix g_2 from $G_2 = SU(2,2)$) as follows: A_2 is the upper principal minor in the 6 by 6 matrix g_3; D_2 is the principal minor located at the 4 and 5 row-column intersections; one more (not a principal) minor B_2 is located on the intersection of rows (1 and 2) with columns (4 and 5); C_2 is located on the intersection of rows (4 and 5) with columns (1 and 2). The remaining entries of the matrix g_{12} are as follows: 1 (if on the main diagonal) and 0 (when off the main diagonal). A reader is now advised to have a look at Fig. A5. Subgroups G_{13} and G_{23} are defined quite similarly. The following statement can be easily verified.

Proposition 1. The subgroup G_{12} acts linear-fractionally on D_{12}; similarly: G_{13} on D_{13}, and G_{23} on D_{23}.

Remark 5. Actually, we can (canonically) define an (equivalent to linear-fractional) action of each (of the three possible) G_{ij} on D_{sk}. As an example, to define such an action of G_{13} on D_{12}, we conjugate G_{13} by the corresponding elementary matrix E. Now, if g is from G_{13} and z is from D_{12}, then the result of the g-action on z is, by definition, the result of the (linear-fractional) action of EgE on z. From now and on, we assume that each G_{ij} acts (in the way which has been just indicated) on any of D_{sk}.

Having the Remark 5 in mind, we define the **color** of an MLM-**quark** D_{sk} as G_{ij} (which can be chosen from G_{12}, G_{13}, G_{23}). In other words, the color of an MLM-quark is defined by the choice of its ruling group (or, even more formally, the color (as a **symbol**) of an MLM-quark is (the symbol of) its ruling group. It goes without saying that the ruling group also acts in the Hilbert space of wave functions of the quark in question (see what we have stated earlier, right before Remark 3). Clearly, there are three colors for quarks of the $U(3)$-level.

The notion of a color for an arbitrary $U(n)$, with the integer n not less than 3, can be introduced quite similarly. The following statement has been proven in [2].

Proposition 2. Each G_{ij} is a subgroup of G_n; each G_{ij} is isomorphic to $SU(2,2)$.

For each level $U(n)$, with $n > 2$, the MLM-quark (of a certain flavor and color) is defined as an ordered triple (D_{pq}, G_{ij}, m). Here m is either 1, or negative 1 (depending on whether we deal with a particle or with an antiparticle). The subgroup D_{pq} in $U(n)$ determines **flavor**, while the subgroup G_{ij} in $SU(n,n)$ determines **color**. An implicit part of this definition is a well-defined representation space, p-space H, which the proton's wave function belongs to. The ruling group G_{ij} acts in this H.

Here is another statement from [2]:

Proposition 3. The total number of colors at the $U(n)$-level is $n(n-1)/2$.

Returning to the $U(3)$-level, let us now present the following possible interpretation in terms of physics: when a proton (participating in a highly inelastic scattering) 'finds itself' in a D_{13}-cell (and it stays there for a moment, name it a *sunken proton*), then its color may be one of G_{12}, G_{23}, G_{13} but changing between them with huge speed, presumably. As stated in the above Table 1, in D_{13} these ruling subgroups generate electric charges 0, 0, negative 1, in that order. It means that such a quark d has an average charge of 1/3. Similarly, a u-quark has a charge 2/3. Notice that (from now and on) we use *sunken* instead of *sank* (which appeared in [2]). On the basis of our (chronometric) Remark 2 (in Section 2), and on the just suggested MLM-approach to the values of SM-quarks' electric charges, the following seems to be logically non-contradictory:

Conjecture 1. The electric charges of the proton and of the electron (viewed either 'inside' neutron, or separately) originate as the result of the corresponding action (in their Hilbert spaces F_p and F_e) of the ruling group G (represented, essentially, by $SU(2,2)$).

Hopefully, such a (to a certain extent, philosophical) view might serve as an answer (preliminary, at least) to the question "What are the origins of electric charge?" According to the SM (as well as to the MLM), there are color charges, too. In our Section 4 the color charges were introduced via the 'MLM-modification' of the Han-Nambu scheme. The electric charges of the SM-quarks are fractional. As based on our Section 4 findings, it is possible to interpret the chronometric proton's electric charge as a special case of the MLM-quarks' color charges: in the $U(2)$-level there is just one ruling group for the proton which means that there is just one color. We interpret this color charge as the electric charge of the chronometric proton.

For the $U(4)$-level, consider the following

Table 2.

f \ c	12	13	14	23	24	34
u: 12, 34	1	0	1	1	0	1
d: 13, 24	0	-1	0	0	-1	0
s: 14	0	-1	0	0	-1	0
c: 23	1	0	1	1	0	1

Again, applying the Han-Nambu scheme, we deduce 'correct' electric charges for MLM-quarks *u*, *d*, *c*, *s* at this level. The total number of colors is 6.

At the *U*(5)-level, there are 10 colors (see Table 3) which implies charges of 7/10 for each of *u*, *c*, *t* and negative 3/10 charges for each of *d*, *s*, *b*. Hence, at the level of *U*(5) the SM-data and the MLM-conclusion (assuming the validity of the Han-Nambu scheme) are contradictory (as regards quarks' electric charges).

Table 3.

f \ c	12	13	14	15	23	24	25	34	35	45
u: 12, 45	1	0	1	1	0	1	1	1	0	1
d: 13, 35	0	-1	0	0	-1	0	0	0	-1	0
s: 14, 25	0	-1	0	0	-1	0	0	0	-1	0
c: 12, 45	1	0	1	1	0	1	1	1	0	1
b: 14, 25	0	-1	0	0	-1	0	0	0	-1	0
t: 12, 45	1	0	1	1	0	1	1	1	0	1

At the *U*(6)-level the number of colors is 15, see Table 4. The MLM-quarks on this level get three new flavors. The electric charges of the new quarks are as follows: $f(4; 1, 6) = f(4; 3, 4) = 2/3$, $f(4; 2, 5) = -1/3$. The charges of other MLM-quarks on this level are the same as of the corresponding SM-quarks.

Table 4.

f \ c	12	13	14	15	16	23	24	25	26	34	35	36	45	46	56
u: 12, 45	1	0	1	1	0	1	0	1	1	1	0	1	1	0	1
d: 13, 35	0	-1	0	0	-1	0	-1	0	0	0	-1	0	0	-1	0
s: 14, 25	0	-1	0	0	-1	0	-1	0	0	0	-1	0	0	-1	0
c: 12, 45	1	0	1	1	0	1	0	1	1	1	0	1	1	0	1
b: 14, 25	0	-1	0	0	-1	0	-1	0	0	0	-1	0	0	-1	0
t: 12, 45	1	0	1	1	0	1	0	1	1	1	0	1	1	0	1
q(4;1; 6)	1	0	1	1	0	1	0	1	1	1	0	1	1	0	1
q(4;2; 5)	0	-1	0	0	-1	0	-1	0	0	0	-1	0	0	-1	0
q(4;3; 4)	1	0	1	1	0	1	0	1	1	1	0	1	1	0	1

5. A 'sunken' vs 'captured' MLM-proton

In our Section 4 (right after Proposition 3) we have provided more details on '*a quark as a sunken proton*' approach (as originally articulated in [2]). Namely, any subgroup G_{ij} of $SU(n,n)$ induces the color charge of 1, -1, or 0 in each 'cell' D_{sk} of the *U*(n)-level. In the current Section 5 we discuss MLM-quarks (and MLM-leptons which can be associated with them) as **stable** ingredients of 'matter at deeper (than *U*(2)) levels'. Such stability presumes that **a proton stays at the d_{sk}-cell not just for a tiny moment** (which could have happened as the result of its participating in highly inelastic scattering – see the portion of our Section 4, between Proposition 3 and Remark 6) **but for an observable** (and even theoretically unlimited, if there is no influence by another interaction) **time interval** – name it a ***captured proton***. Let us **postulate** that such **a capture may happen only if** the **ruling group G_{ij} 'matches' the cell D_{sk}**: i = s and j = k. In particular, such a hadron p_{ij} has an electric charge of 1 or of negative 1.

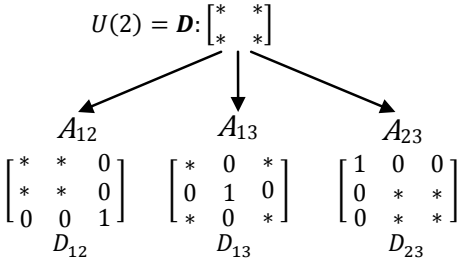
The findings of [17] are graphically presented: Figures B1, B2, and B3. Clearly, it is important to move from this toy example to a more realistic one where f depends on all four variables (notice that, obviously, f(x,y,z,t) is positive everywhere).

If the value (as a function of time) of the mass operator at such a state is almost constant, then we can set this value to be the proton's mass. We believe that such a realistic wave function will be detected. In that case, it will be possible to relate our preliminary analytic findings in the proton's space F_p to certain attempts of research, dedicated to the radius of a proton or, more broadly, to the shape of nucleons: [18, 19].

Funding: The research by A. V. Levichev was partly funded by the State Program of fundamental scientific research of the Sobolev Institute of Mathematics (SB RAS, project No FWNF-2022-0006).

Appendix A

Figure A1.

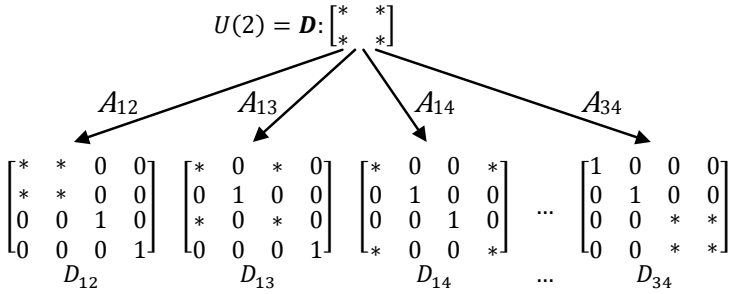


D_{12}, quark u; D_{13}, quark d; D_{23}, quark u – subgroups in $U(3)$ isomorphic to $U(2)$.

$$P_2: \begin{bmatrix} z_1 & z_2 \\ z_3 & z_4 \end{bmatrix} \to \begin{bmatrix} z_4 & z_2 \\ z_3 & z_1 \end{bmatrix}; \quad P_3: \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \to \begin{bmatrix} i & f & c \\ h & e & b \\ g & d & a \end{bmatrix}.$$

$$A_{23} = P_3 \circ A_{12} \circ P_2, \quad P_3(D_{12}) = D_{23},$$
$$P_3(D_{23}) = D_{12}, \quad P_3(D_{13}) = D_{13},$$

Figure A2.



$A_{24} = P_4 \circ A_{13} \circ P_2, \quad A_{34} = P_4 \circ A_{12} \circ P_2, \quad P_4(D_{13}) = D_{24}, \quad P_4(D_{24}) = D_{13},$

$P_4(D_{12}) = D_{34}, \quad P_4(D_{34}) = D_{12},$
$P_4(D_{14}) = D_{14} -$ quark s; $\quad P_4(D_{23}) = D_{23} -$ quark c.

Figure A3.

$$U(6): \begin{cases} D^u_{12} & D^d_{13} & D^s_{14} & D^t_{15} & D^{q(4;1,6)}_{16} \\ & & & & 2/3 \\ D^c_{23} & D^b_{24} & D^{q(4;2,5)}_{25} & D^t_{26} & D^{q(4;3,4)}_{34} \\ & & -1/3 & & 2/3 \\ D^b_{35} & D^s_{36} & D^c_{45} & D^d_{46} & D^u_{56} \end{cases}$$

$$U(5): \begin{cases} D^u_{12} & D^d_{13} & D^s_{14} & D^t_{15} & D^c_{23} \\ 2/3 & -1/3 & -1/3 & 2/3 & 2/3 \\ D^b_{24} & D^s_{25} & D^c_{34} & D^d_{35} & D^u_{45} \\ -1/3 & -1/3 & 2/3 & -1/3 & 2/3 \end{cases}$$

$$U(4): \begin{matrix} D^u_{12} & D^d_{13} & D^c_{23} & D^d_{24} & D^u_{34} & D^s_{14} \\ 2/3 & -1/3 & 2/3 & -1/3 & 2/3 & -1/3 \end{matrix}$$

$$U(3): \begin{matrix} D^u_{12} & D^u_{23} & D^d_{13} \\ 2/3 & 2/3 & -1/3 \end{matrix}$$

The fractional electric charges are entered in accordance with the SM. Please compare with our Section 4 where the Han-Nambu scheme has been applied. In that case, the charges of the $U(5)$-quarks would be different. Also, here we interpret each MLM-quark as a 'sunken proton' (rather than as a captured one).

Appendix B

Figure B1.

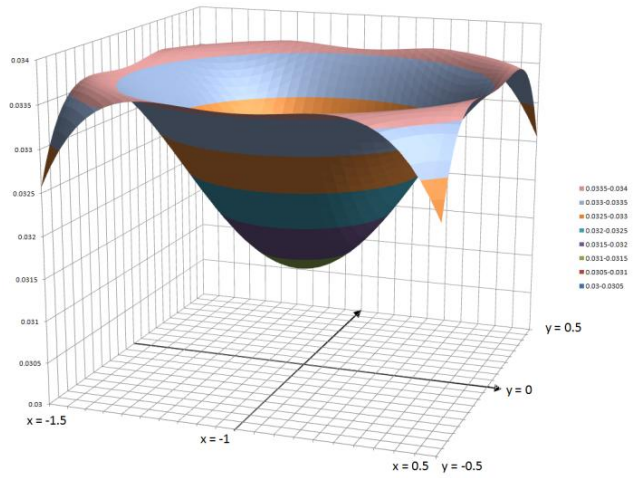


Figure B2.

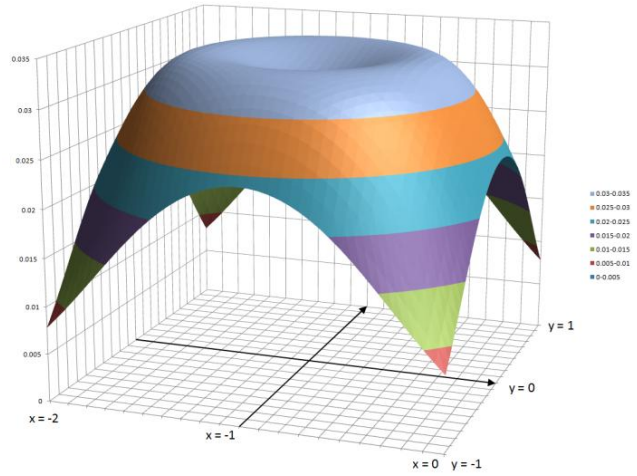
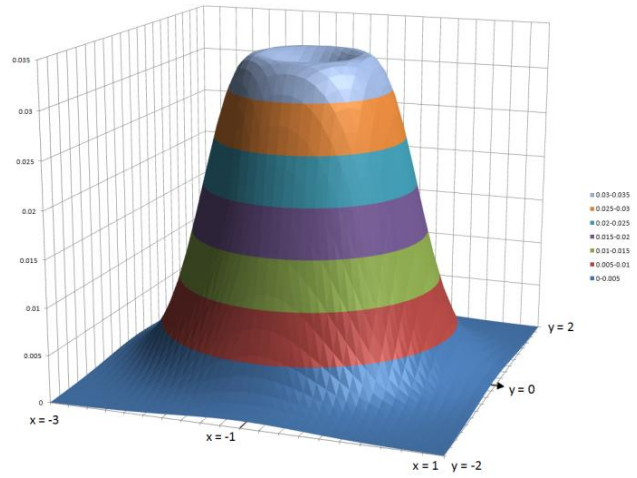


Figure B3.



References

[1] Levichev, A.V. Towards a matrix multi-level model of quark-gluon media. *J. Progressive Res. Math.* **2016**, *10*, pp. 1493 --1496.

[2] Levichev, A.V. One possible application of the Chronometric Theory of I.E. Segal: A toy model of quarks and gluons. *J. Phys.: Conf. Ser.* **2019**, **1194** 012071

[3] Segal, I.E. Is the Cygnet the quintessential baryon? *Proc. Natl. Acad. Sci*. **1991**, *88*, pp. 994—998.

[4] Wigner, E.P. On unitary representations of the inhomogeneous Lorentz group. *Ann. of Math*. (2) **1939**, *40*, pp. 149—204.

[5] Levichev, A.V.; Palyanov, A.Yu. On colors and electric charges of quarks: modeling in terms of groups U(n) and SU(n,n). *Mathematical Structures and Modeling.* **2020**, *4*(*56*), pp. 31--40, in Russian.

[6] Levichev, A.V. On key properties of the intertwining operators ornament in the matrix multi-level model of the quark-gluon media. In Proceedings of the All-Russia conference with the international participation "Knowledge-Ontology-Theories" (KONT-2017), Novosibirsk, Sobolev Institute of Mathematics of the Siberian Division RAS, Russia, 6-8 October 2017; Volume 2, pp. 41–47.

[7] Kon, M.; Levichev, A. Parallelization analysis of space-time bundles and applications in particle physics. In Proceedings of the All-Russia conference with the international participation "Knowledge-Ontology-Theories" (KONT-2019), Novosibirsk, Sobolev Institute of Mathematics of the Siberian Division RAS, Russia, 7--11 October 2019; pp. 385-–392.

[8] Levichev, A.; Palyanov, A. Standard charges of quarks determination in terms of the multi-level model. In Proceedings of the All-Russia conference with the international participation "Knowledge-Ontology-Theories" (KONT-2019), Novosibirsk, Sobolev Institute of Mathematics of the Siberian Division RAS, Russia, 7--11 October 2019; pp. 222--226.

[9] Jakobsen, H.P.; Levichev, A.V.; Palyanov, A.Yu. The Wigner-Segal method as applied to the problem of quarks' and leptons' generations. . In Proceedings of the All-Russia conference with the international participation "Knowledge-Ontology-Theories" (KONT-2021), Novosibirsk, Sobolev Institute of Mathematics of the Siberian Division RAS, Russia, 8--12 November 2021; pp. 344--352. Online at http://math.nsc.ru/conference/zont/21/index.htm

[10] Jakobsen, H.P.; Levichev, A.V. The representation of SU(2,2) which is interpreted as describing chronometric fermions (proton, neutrino, and electron) in terms of a single composition series. *Rep. on Math. Phys.* 2022, V. 90, no. 1, pp. 103-121.

[11] Levichev, A.V. Segal's Chronometry: its development, application to the physics of particles and their interactions, further perspectives. *Poisk matematicheskih zakonomernostei Mirozdania*, M.M.Lavrent'ev, V.N.Samoilov, Eds.; GEO: Novosibirsk, Russia, 2010; Volume 7, pp. 66-99, in Russian.

[12] Levichev, A.V. Pseudo-Hermitian realization of the Minkowski world through DLF theory. *Physica Scripta*. 2011, *83*, pp. 1—9. Online at https://iopscience.iop.org/article/10.1088/0031-8949/83/01/015101

[13] Breidenbach, M. *et al*. Observed Behavior of Highly Inelastic Electron-Proton Scattering. *Phys. Rev. Lett*. 1969, *23*, pp. 935--939.

[14] Barut, A.O. A. Return to 1932: proton, electron and neutrino as true elementary constituents of leptons, hadrons and nuclei. *Quantum Theory and the Structures of Time and Space*, 1981, Volume 4, pp. 152--163. Carl Hanser Press, Munich.

[15] Varadarajan, V.S. Geometry of Quantum Theory; Springer: New York, USA, 1985.

[16] Faessler, M.A. Weinberg Angle and Integer Electric Charges of Quarks. https://arxiv.org/abs/1308.5900 , 2013.

[17] Levichev, A.V.; Klevtsova, Yu.Yu.; Palyanov, A.Yu. A. D. Alexandrov would have been 110, and a contribution to Chronometry. *Mathematical Structures and Modeling.* 2022, *2*(*62*), pp. 66--75, in Russian.

[18] Khabarova, K.Yu.; Kolachevsky, N.N. Proton charge radius. 2021. *Uspekhi Fizicheskikh Nauk*. *191*(*10*), pp. 1095-1106, in Russian, https://ufn.ru/ru/articles/2021/10/d/

[19] Nekrasov, M. L. On the Shape of Nucleons at High Energies. *Particles*. 2021, *4*(*3*), pp. 381-390.

Evaluating Student Knowledge Levels through Fuzzy Logic in the "Smart School" Information System

S.A.Masharipov[1], L.S.Rakhimova[1], X.I.Akhmedova[2]

[1]*Urgench branch of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, al-Khorezmi street 110, Urgench city, Uzbekistan*

[2]*Alisher Navo'i Tashkent State University of the Uzbek language and Literature, 103, Yusuf Xos Khojib street, Tashkent city, Uzbekistan*

sanatbek@gmail.com, laylorakhimova@gmail.com, xolisa9029@mail.ru

Abstract: In the fast-paced realm of contemporary information technologies, it is crucial to address issues pertaining to the efficacy of education quality in secondary schools, the transparency of teacher evaluations, and the analysis of several other indicators with the assistance of artificial intelligence. In this analytical method, initial values are inputted by school experts. These values comprise of a student's standing in a particular category among their peers and their academic grades. "Category" here refers to various criteria, including students with differing levels of cognitive activity, underperforming students, frequent absentees, and those whose learning levels are high or low within certain subjects or topics. The "Smart School" program generates arrays from the number of grades and students' marks, according to the criteria set by experts, and is capable of sorting values similar to the inputted ones. The problem addressed in this article is elucidated both mathematically and practically. The business process is modelled based on the proposed mathematical models, a business process software product is developed, and statistical data is acquired.

Keywords: Smart school, information system, mathematical model, array, set elements, similarity indicators, association rule, time series, fuzzy logic.

1 Introduction

In The growing student population in schools is leading to a significant disparity in the teacher-student ratio. It is evident that a single teacher cannot always actively participate in evaluating each student's knowledge level, identifying topics that are challenging to comprehend, and other education-related processes. This limited one-on-one interaction may precipitate a significant decline in the quality of education.

Elevating the quality of education in Uzbekistan and upgrading the educational system through the implementation of cutting-edge information technologies, specifically artificial intelligence, is deemed a critical mission at present. Ensuring the cognitive and physical development of students, providing a foundation for them to fully harness their potential in the future, and aiding them in finding their place in society are amongst the pressing matters of today.

A rising trend in research on this subject is observable, as new programs and systems powered by artificial intelligence are steadily permeating the education sector. The automation of the educational process in higher and secondary education systems poses a significant challenge. However, this issue has been partially addressed through the development of an automated information system for secondary education, which is currently operational in 10 public schools in Fergana city, Uzbekistan.

The "Smart School" program utilizes artificial intelligence to evaluate education quality, teacher evaluation transparency, and other key indicators in schools. It also identifies subjects that students find difficult to understand. Systematic measures are taken on challenging topics, and individual students who show signs of inactivity in class or struggle with understanding are identified. This allows for effective, personalized work with each student. Regular updates about a child's academic progress are provided to parents in a histogram format through a mobile application, enabling advice to be given to teachers and parents about students who frequently miss classes, and the potential reasons behind this.

The program bridges the gap between the school, student, and parents, providing opportunities in the following ways:

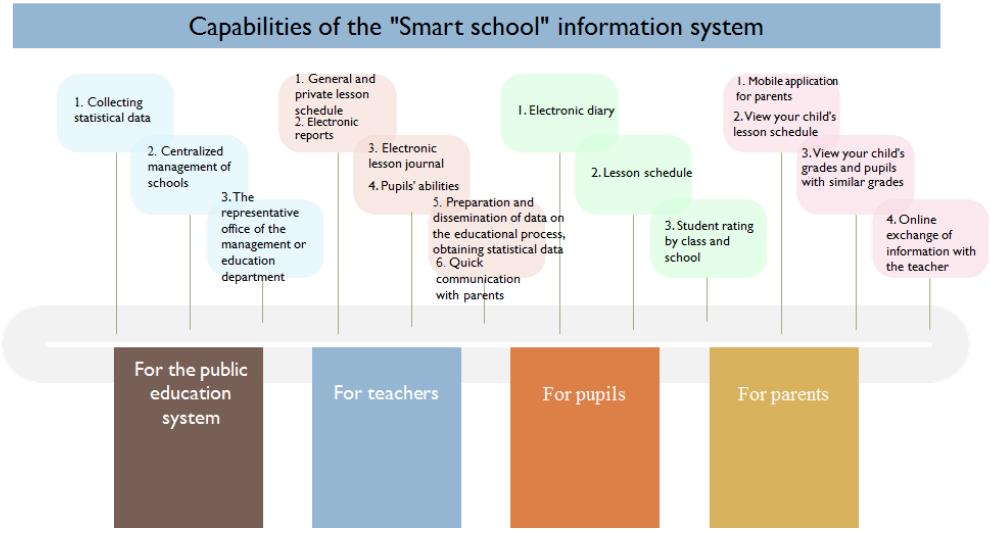


Fig.1. Capabilities of the "Smart school" information system

The Smart School program is a software tool engineered for electronic administration and work process analysis in schools, encompassing logs and daily records. Its primary function is to facilitate the operations of school administrators, teachers, and other staff. The software employs Artificial Intelligence (AI) systems, including associative rule-based searches, time series forecasting, and fuzzy logic, to analyze students' cognitive capabilities and physical development.

Features such as electronic journals, digital diaries, online information exchange with parents, access to exam schedules and results, and online class schedules assist students in fully engaging in classes and improving learning efficiency. Notably, the program can analyze a student's physical development and physical training in relation to their age, suggest standard physical exercises, and recommend activities to ensure the student's health.

The "Smart School" project incorporates AI systems like associative rule-based searches, time series predictions, and similarity identification, effectively identifying multiple students with grades similar to a randomly selected student. During the development of this functionality, the problem of similarity identification was encountered.

The Smart School program is a software tool engineered for electronic administration and work process analysis in schools, encompassing logs and daily records. Its primary function is to facilitate the operations of school administrators, teachers, and other staff. The software employs Artificial Intelligence (AI) systems, including associative rule-based searches, time series forecasting, and fuzzy logic, to analyze students' cognitive capabilities and physical development.

Features such as electronic journals, digital diaries, online information exchange with parents, access to exam schedules and results, and online class schedules assist students in fully engaging in classes and improving learning efficiency. Notably, the program can analyze a student's physical development and physical training in relation to their age, suggest standard physical exercises, and recommend activities to ensure the student's health.

The "Smart School" project incorporates AI systems like associative rule-based searches, time series predictions, and similarity identification, effectively identifying multiple students with grades similar to a randomly selected student. During the development of this functionality, the problem of similarity identification was encountered.

Research Methodology

Various scientific works on similarity identification were examined. It was found that the method of identifying similarities largely depends on the object in question. For instance, Haipeng Chen and Xiwen Yang from China used four principles to detect the forgery of replicated artworks:

- Frequency domain-based method
- Size reduction-based method
- Local binary pattern-based method
- Texture-based method

Additionally, several researchers like Varun Chandola, Luke Michael Febriansyah, Farhan Ullah, Gudkov P.A, have conducted studies on similarity coefficients in diverse directions. Importantly, whether it involves identifying similarities between bodies or arbitrary objects, decisions are often made based on elements of mathematical statistics and probability theory.

In this context, a few considerations become significant:

- Associative rules serve as guidelines for determining the correlation probability between elements in large databases. They can help ascertain the dependency probability of grades achieved by students in one subject on their grades in other subjects.

• Time series prediction involves data collection based on a certain time sequence. Time series analysis allows for future predictions based on past numerical variabilities.

• Fuzzy logic, an AI branch, mimics human intelligence to aid in decision-making. Fuzzy logic models in the "Smart School" program are used to assess students' knowledge levels in subjects.

The significant teaching load complicates the process of assessing students' knowledge levels. This issue can be mitigated through AI systems. In the "Smart School" program, AI systems analyze students' grades in each subject to identify the subjects in which the student is active or struggling.

In this paper, the term "category" refers to various criteria like students with varying mental activity levels, students who are performing poorly, missing many classes, or demonstrating high or low learning levels in specified subjects or topics. The "Smart School" program forms arrays of grades and pupils' grades in subjects based on the expert-entered criteria and can sort values similar to the entered value.

The solutions proposed in this paper include giving parents access to the "Smart School" information system, where they can view their child's grades and identify other students at the same level as their child.

To address this problem, we require an array of grades of students in one category, for example, the grades of 7th class students in subjects such as algebra, geometry, physics at the end of the school year. Assuming parents or experts wish to ascertain the grades of a specific student and other similar students during an optional time of the school year, we consider a two-dimensional array consisting of the total grades of students from the academic year to the current time as an object.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \qquad (1)$$

Here $a_{ij} \in \{2,3,4,5,\emptyset\}$, a_{ij} $i-$ is the grade of the i-pupil on the j-subject, n∈Z is the number of subjects in the class denotes the number of subjects and m∈Z denotes the number of pupils in the class. Table 1 below shows the elements of array A

Table 1: Array of pupils' grades

	Theme 1	Theme 2	Theme 3	Theme 4	Theme 5	…	Theme n
P 1	5	4	4	3	5	…	4
P 2	4	3	5	5	4	…	5
P 3	5	5	4	3	4	…	5
P 4	4	4	5	5	3	…	5
P 5	3	5	4	4	5	…	4
…	…	…	…	…	…	…	…
P m	4	5	4	5	4	…	5

Operations are performed on selected elements of matrix A. The elements of matrix A consist of 5 different grades 0, 2, 3, 4, 5 (where the number 0 means not graded on the subject). Now the number of grades for each pupil is counted and as a result, matrix B is formed

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ \cdots & \cdots & \cdots & \cdots \\ b_{m1} & b_{m2} & b_{m3} & b_{m4} \end{pmatrix} \qquad (2)$$

Here b_{i1} – number of 5 grades of i th pupil, b_{i2} – number of 4 grades, b_{i3} – number of 3 grades, b_{i4} – number of 2 grades. That is to say:

$$\mathbf{B}=\begin{pmatrix} 2 & 4 & 5 & 2 \\ 5 & 3 & 1 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 7 & 0 & 0 & 0 \end{pmatrix}$$

a matrix consisting of such values is formed.

Let's assume that the system is required to identify pupils whose grades are similar to the grades of k-pupil:

$$x_k = \{b_{k1}, b_{k2}, b_{k3}, b_{k4}\}$$

x_k array C consisting of elements of array B satisfying the condition $|b_{k4} - b_{i4}| \leq 1$ between the number of grades of the pupil and the number of grades of other pupils is created.

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ \cdots & \cdots & \cdots & \cdots \\ c_{y1} & c_{y2} & c_{y3} & c_{y4} \end{pmatrix} \qquad (3)$$

Here 1<y<m.

Array D is formed by dividing the elements of the created array C with the condition $|b_{k3} - b_{i3}| \leq 1$, the difference between the numbers of 3 values in the array C consists elements of the set {-1;0;1}

$$\mathbf{D} = \begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ \cdots & \cdots & \cdots & \cdots \\ d_{z1} & d_{z2} & d_{z3} & d_{z4} \end{pmatrix} \quad (4)$$

Here 1<z<y

In the next step the 2nd column of the array D, i.e. the number of 4 grades of the pupils, the array E consisting of elements satisfying the condition $|b_{k2} - b_{i2}| \leq 1$ is created.

$$\mathbf{E} = \begin{pmatrix} e_{11} & e_{12} & e_{13} & e_{14} \\ e_{21} & e_{22} & e_{23} & e_{24} \\ \cdots & \cdots & \cdots & \cdots \\ e_{p1} & e_{p2} & e_{p3} & e_{p4} \end{pmatrix} \quad (5)$$

Here 1<p<z

Now it's time to calculate the differences between the 5 grades of the pupils and this operation is performed on the elements of the E array. As a result, an array F consisting of elements satisfying the condition $|b_{k1} - b_{i1}| \leq 1$ of the array E is created.

$$\mathbf{F} = \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ \cdots & \cdots & \cdots & \cdots \\ f_{l1} & f_{l2} & f_{l3} & f_{l4} \end{pmatrix} \quad (6)$$

Here 1<l<p

F is the resulting array, the differences between the values of the current k-pupil $x_k = \{b_{k1}, b_{k2}, b_{k3}, b_{k4}\}$ consisting elements of the set {-1,0,1} are extracted. Now, according to the order of the array F, the elements between the elements of the array A are separated

$$\boldsymbol{A'} = \begin{pmatrix} a'_{11} & a'_{12} & \ldots & a'_{1n} \\ a'_{21} & a'_{22} & \ldots & a'_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a'_{l1} & a'_{l2} & \ldots & a'_{ln} \end{pmatrix} \quad (7)$$

The resulting array of $\boldsymbol{A'}$ elements is required to be sorted when presenting the results to the user. It is expedient to perform the sorting according to the sum of elements of the array $\boldsymbol{A'}$. Let S be ordered matrix

$$\mathrm{S} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_l \end{pmatrix} \quad (8)$$

Elements of the array $\boldsymbol{A'}$ are displayed which satisfying the condition $s_i = f_{i1} + f_{i2} + \cdots f_{in}$ and $s_1 \geq s_2 \geq s_3 \geq \cdots \geq s_l$

We will consider the process of solving this problem based on the above mathematical expressions. This business process is modeled as in Fig.2.

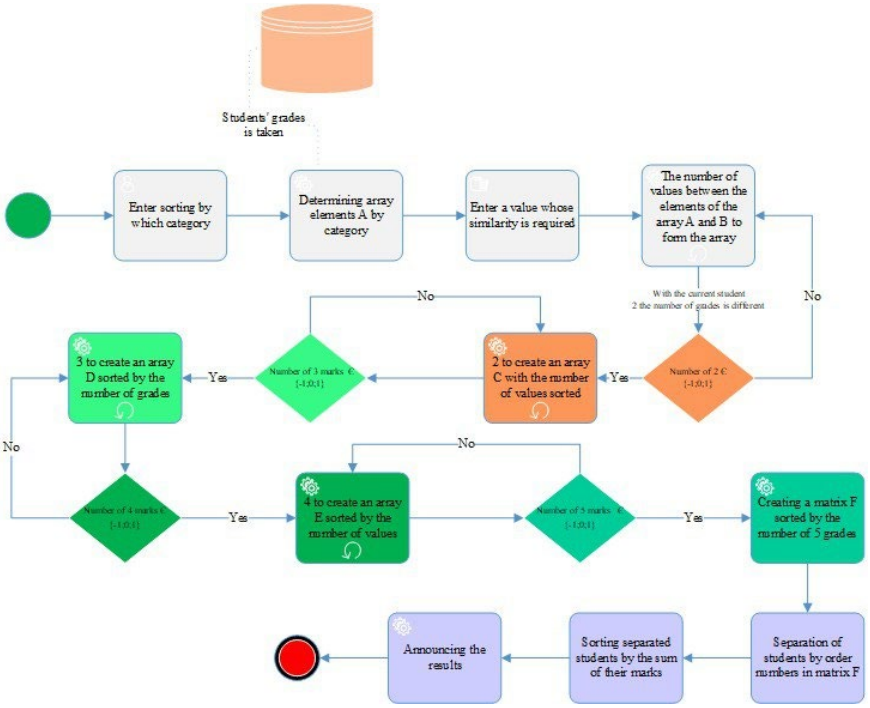


Fig. 2. Business process

A program will be created based on the developed business process. The “Smart school” information system is developed in the Java programming language in which the interfaces are initially written. Individually written interfaces are inherited and generated methods are filled and passed to the browser. The interface that creates the elements of array A can be composed of the following methods.

```
List<Fan> getFanListAtThatDateBySinfId(int sinfId, String sana, String lang);
List<Jurnal> getJurnalListByGuruhIdAtThatDate(int sinfId, String sana, String
lang);
List<Jurnal> getJurnalListGuruhIdByBetweenDate(int guruhId, String sana);
List<Jurnal> getJurnalForBahoMap(String sana, int sinfId, int fanId, double soat,
String lang);
List<Jurnal> getJurnalForHisobot(String sana, int sinfId, int fanId, double soat, String
baho);
List<Jurnal> getJurnalByFanIdAndSinfId(int fanId, int sinfId, int param, String
begin,String end);
List<Jurnal> getJurnalByFanIdAndSinfIdInSoat(String sana, int fanId, int sinfId);
List<Jurnal> getJurnalByFanIdSanaSinfSoat(int fanId, String sana, int sinfId, double
soat);
```

Interface methods are written according to the necessary categories. Pupils' grades were obtained from the database with the help of written methods. Let's write one of the similar methods such as

```
public List<Jurnal> getJurnalByFanIdAndSinfId(int fanId, int sinfId, int param,
String begin,String end) {
        String sql = "";
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");
        LocalDate localDate = LocalDate.now();
        String presentsDay = localDate.format(formatter);
//        System.out.println(begin);
        if (param != 0) {
            System.out.println(param + "param");
            System.out.println(begin + "  " + presentsDay);
            sql = "SELECT j.fan_id,f.fan_name_uzb as fanName, j.sana,j.soat,
u.uquvchi_id, j.baho  FROM jurnal j" +
                    " left join uquvchi u on j.uquvchi_id=u.uquvchi_id " +
                    " left join fan f on f.fan_id=j.fan_id" +
                    " where j.fan_id=? and u.sinf_id=? and sana between ? and ?
order by sana";
            return (List<Jurnal>) jdbcTemplate.query(sql, new Object[]{fanId, sinfId,
begin, presentsDay}, new SanaRowMapper());
        } else {
            System.out.println(param + "param");
            sql = "SELECT j.fan_id,f.fan_name_uzb as fanName, j.sana,j.soat,
u.uquvchi_id, j.baho  FROM jurnal j" +
                    "   left join uquvchi u on j.uquvchi_id=u.uquvchi_id " +
                    " left join fan f on f.fan_id=j.fan_id" +
                    "  where j.fan_id=? and u.sinf_id=? ORDER BY sana";
            return (List<Jurnal>) jdbcTemplate.query(sql, new Object[]{fanId,
sinfId}, new SanaRowMapper());
        }
    }
```

Methods for other similar categories are written like this and elements of array A are created. An array B is created based on the generated array and a one-dimensional array x_k is also created based on the entered value. In the next step, arrays B and x_k are compared. Comparing two arbitrary arrays can be done using the code below

```
public class Test1 {
 private static double data1[] =
{0.7,0.57,0.81,0.32,0.33,0.68,0.91,0.24,0.2,0.5,0.0,0.64,0.12,0.43,0.74,0.28,0.38,0.3
,0.0,0.4,0.9,0.68,0.75,0.8,0.9,0.6,0.75,0.38,0.16,0.26};
 private static List<List<Double>> testList = new ArrayList<>();
 private static List<List<Double>> resultList = new ArrayList<>();
 public static void main(String[] args) {
  double minValue = 0;
  double maxValue = 0;
```

```
  for (int i = 0; i < 10; i++) {
   List<Double> testData = new ArrayList<>();
   for (int j = 0; j < 30; j++) {
    testData.add(Double.parseDouble(String.format("%.2f", Math.random()).replace(',',
'.')));
   }
   testList.add(testData);
  }
  for (List<Double> dList : testList) {
   System.out.println(dList);
   List<Double> fResult = new ArrayList<>();
   for (int i = 0; i < dList.size(); i++) {
    double d = 1-dList.get(i);
    double d1 = Double.parseDouble(String.format("%.2f", d). replace(',', '.'));
    fResult.add(d1);
   }
   resultList.add(fResult);
   System.err.println(fResult);
  }  } }
```

Using this method, the arrays are compared and the necessary results are presented to the user.

The results of research and discussions

Based on the mathematical models we created and the business processes modeled after them, we integrated a similarity detection function into the "Smart School" information system. In the 10 schools where the system is in use, students are accurately identified based on the information input into the program. If the function needs to cover the grades of 10 school students, it's necessary to input the resultant number of students. For instance, if the system needs to identify how many students have similar grades to a particular student, the user is required to input the desired number of students. In addition, the system identified the following learning indicators in schools:

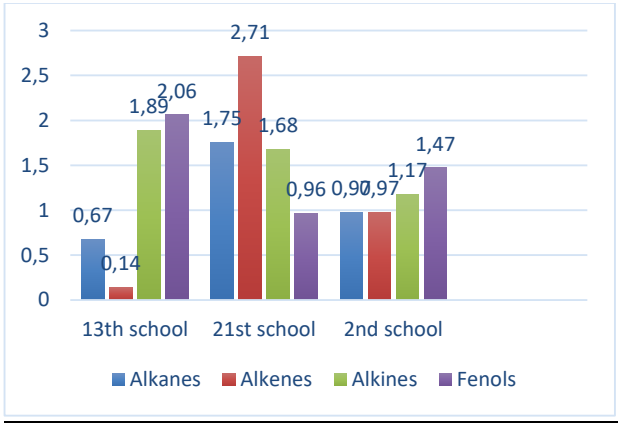
1. After analyzing the mathematics learning of 10th-grade students in schools 2, 21, and 40, we found that certain topics were challenging for the students to grasp(Table 2):

Table 2. Statistics of research

Schools	Themes		
	Negation, disjunction and conjunction	Simple percentage, complicated percentage	Simple rational equations and their systems
School №40	4.18	3.18	3.82
School №21	1.77	0.37	0.89
School №2	1.91	1.48	2.67

Table 2 shows that in 2 schools it is difficult for pupils of the 10th grade to master these topics in algebra, while in the school №40, these topics are mastered relatively well. In this case, studying the methodology of teaching math in school №40 and applying it to other school may lead to master the subject in all other schools.

2. In addition, the "Smart School" information system has the ability to determine the mastery coefficients of subjects that are difficult for pupils to master. This possibility can be seen in the diagram below:



From the diagram, we can observe that the assimilation coefficients of subjects identified as challenging within the school framework have been detailed. In the public education system, it is advisable to revise the teaching methods for these subjects to make them more comprehensible. Similarly, the system can identify other subjects where learners struggle with particular topics.

CONCLUSIONS

The designed information system identifies students with high academic performance based on criteria such as the number of grades received by students, the frequency of class participation, and mastery indicators of subjects. For instance, looking at the example of School No. 10, it was found that there are four students in the 5th "A" class and three students with a high level of knowledge in the 6th "G" class. Furthermore, in School No. 10, the mother tongue proficiency level of the 5th "E" grade students was 4.9 out of a maximum of 5 points, and the class's average indicator was calculated by the program according to each student's level. With this analysis method, it will be possible to evaluate the level of students and classes at the school level.

REFERENCES

1. Asemi, A. (2006). Information Technology and National Development in Iran. Paper presented at the 2006 International Conference on Hybrid Information Technology (ICHIT'06).
2. Zaenab S. Majeed Zailin Shah Yusoff. *Are We 'Smarter' Now? Case Study of Smart School Implementation in a Developing Nation*. Journal of Studies in Education· August 2015
3. Henry, M., & Abala, A. (2009). *E-learning research methods and approaches.Paper presented at the Proceedings of the First Kuwait* Conference on e-Services and e-Systems.
4. Siavash Omidinia, Maslin Masrom, Harihodin Selamat. *An Examination of the Concept of Smart School: An Innovation to Address Sustainability.* 2nd International Conference on Advances in Computer Science and Engineering (CSE 2013*)*
5. Коротенков Юрий Григорьевич. *Образовательные Технологии В Smart-Образовании*. г. Москва, Россия
6. Mingsep Sampebua, Septiani Mangiwa. *The Design Smart School Application to Increase Education in Junior High School* International Journal of Computer Science and Information Security (IJCSIS), Vol. 15, No. 10, October 2017
7. Hennessy, S., Ruthven, K., & Brindley, S. (2005).*Teacher perspectives on integrating ICT into subject teaching: commitment, constraints, caution and change*.Journal of Curriculum Studies, 37(2), 155-192.
8. Mahmudi, J., Nalchigar, S., & Ebrahimi, S. B. (2008). *Challenges of Smart Schools in Iran*. Quarterly Journal of Educational Innovations, 7(27), 78-61
9. Puteh, M., & Vicziany, A. M. 2004. *How smart are Malaysia's smart schools?* Paper presented at the 4th Global Congress on Engineering Education,Bangkok, Thailand, 5 – 9 July.
10. Muslimov N.A va boshqalar. Innovatsion ta'lim texnologiyalari. O'quv metodik qo'llanma. –T.: Sano standart, 2015.-208b.
11. Tolipov U.K, Usmonboyeva M. Pedagogik texnologiya: nazariya va amaliyot. –T.:Fan, 2005.-205b.
12. https://yuz.uz/uz/news/fargonadagi-aqlli-maktablarda-oquvchilar-bilimini-suniy-intellekt-baholamoqda
13. Bustanov X.A, Xasanov U. Oliygohlarda zamonaviy axborot ta'lim muhiti va ulardan foydalanishning metodik aspektlari.
14. HAIPENG CHEN, XIWEN YANG , YINGDA LYU "Copy-Move Forgery Detection Based on Keypoint Clustering and Similar Neighborhood Search Algorithm"// SPECIAL SECTION ON INNOVATION AND APPLICATION OF INTERNET OF THINGS AND EMERGING TECHNOLOGIES IN SMART SENSING, 2020
15. Chandola, V., Banerjee, A., and Kumar, V. 2009. Anomaly detection: A survey. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages. DOI = 10.1145/1541880.1541882 http://doi.acm.org/10.1145/1541880.1541882
16. Luke Michael Febriansyah 2019. Analysis winnowing algorithm for text plagiarism detection using three method similarity. PROXIES VOL. 2 NO. 2, TAHUN 2019
17. Farhan Ullah, Abdullah Bajahzar, Hamza Aldabbas, Muhammad Farhan, Hamad Naeem, S. Sabahat H. Bukhari, Kaleem Razzaq Malik, 2020. An E-Assessment Methodology Based on Artificial Intelligence Techniques to Determine Pupils' Language Quality and Programming Assignments' Plagiarism.// Intelligent Automation And Soft Computing, 2020 Vol. 26, no. 1, 169–180 https://doi.org/10.31209/2019.100000138
18. Гудков П.А. Методы сравнительного анализа. Учеб. пособие. – Пенза: Изд-во Пенз. гос. ун-та, 2008 – 81 с.
19. R. Agrawal, T. Imielinski, A. Swami, *"Mining Association Rules between Sets of Items in Large Databases"* in SIGMOD '93 Proceedings of the 1993 ACM SIGMOD international conference on Management of data, Washington DC, USA, 1993, vol. 22, no. 2, pp. 207-216.
20. Hu, Y.C., Chen, R.S., Tzeng, G.H.: Mining fuzzy associative rules for classifications problems. Computers and Industrial Engineering, 43 (4) pp. 735-750 , 2002.

21. Box, G., Jenkins, G.M., Reinsel, G.C., 2008. Time Series Analysis: Forecasting and Control, fourth ed. Hoboken, N.J., Wiley, New York.
22. García-Honrado, 2012. Fuzzy logic at schools and high schools?," 2012 Annu. Meet. North Am. 3 Fuzzy Inf. Process. Soc. NAFIPS 2012
23. M. Ismail and L. Syaiful, 2015. Affective assessment in learning using fuzzy logic, 2015 IEEE Conf. e-Learning, e-Management e-Services, pp. 98–102,
24. A. Albayrak, M. Albayrak, and R. Bayir, 2015. Design of Matlab/Simulink based development board for fuzzy logic education, IEEE Int. Conf. Fuzzy Syst., vol. 2015–November, 2015.
25. Zadeh, L.A. (1965). Fuzzy Sets. Information and Control, 8: 338–353.
26. Мелихов А.Н., Берштейн Л.С., Каровин С.Я. Ситуационные советующие системы с нечеткий логикой // М. наука.Гл.ред. физ.-мат. лит., 1990.-278 с.-ISBN 5-02-014144-5
27. Sun'iy intellekt asosida yoshlarni kasbga to'g'ri yo'naltirish tizimini ishlab chiqish. Muhammadsolayev A.Sh, Masharipov Sanatbek Atanazarovich, Iskandarov S.Q., Sharifboyeva R.B. International scientific conference "INFORMATION TECHNOLOGIES, NETWORKS AND TELECOMMUNICATIONS" ITN&T-2022 Urgench, 2022y April 29-30

Development of Efficient PON Architecture Models for Small Areas

U.K.Matyokubov[1], D.A.Davronbekov[2], T.A.Matqurbanov[1]

[1]Urgench branch of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, al-Khorezmi street 110, Urgench city (Uzbekistan)
[2]Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan.

otkir_matyokubov89@mail.ru, d.davronbekov@gmail.com, tulkinmatkurbanov2020@gmail.com

Abstract.
This article presents models of the legacy PON architecture as well as new models of the PON architecture. The backup method has been widely analyzed and used to provide reliability indicators. A comparative analysis of the presented protection schemes was carried out. Analytical expressions and models of each protected circuit are developed. Using the developed analytical expressions, calculations of reliability parameters were carried out, and based on this, the analytical results were depicted in the form of diagrams. Using the analyzed models and algorithms, it was found that the overall reliability index of the PON network can be predicted.
Keywords: optical fiber, reliability, Passive optical network (PON), optical line terminal (OLT), model, splitter, additional protection, spare line, availability ratio

1 Introduction

Today, Passive Optical Network (PON) technology is becoming more and more popular in the field of communication. As a result, the need to maximize the use of frequency bands is becoming increasingly important. Network reliability has become a key criterion for service providers today. At the same time, the reliability of the traditional architecture of optical access networks is low. This results in data loss when optical components fail in both the equipment and the transmission medium. Therefore, it is necessary to develop architectures that provide the required reliability without sacrificing network bandwidth. The issue of reliability is especially important in the application of long-distance access networks in small areas, which allow to increase the scale of deployment compared to classical PONs [1, 2]. This is because the reliability coefficient directly depends on the network elements and architecture to be deployed. The highest requirements for resilience to network failures are placed by data providers, for whom a short-term loss of traffic can cause significant financial losses and affect the reputation of the network operator. Therefore, they always value the high network availability guaranteed by the telecom operator at a high price. At the same time, the operator tries to find a compromise between high network availability and capital and operating costs [3-7, 21-23].

The purpose of this article is to analyze various PON-based network protection schemes from the point of view of their reliability and consider the possibilities of using reserved schemes. In addition, it consists in comparing the presented schemes according to other indicators.

2 Main part

FTTH technology belongs to the FTTx (fiber to x optical fiber to point X) technology family. FTTH technology is a solution that provides high throughput from the data transmission device located in the central office to the subscriber's house. FTTH technology uses a passive optical network (PON). Access network line based on FTTH technology uses optical fiber cables. FTTH technology is installed based on two special configurations. The first one is installed up to each subscriber user in the fiber optic cable access network. This type of configuration is called a point-to-point (PTP) network. The second is that one fiber is distributed among a certain number of users, where from four to thirty-two (through a power splitter) can be distributed to a subscriber line. PONs are characterized by one or more splits of the fiber optic cable at the network nodes, as a result of which the fiber is shared between several users.

Fiber in a PON is typically shared by four to thirty-two users. The network is divided by an optical splitter. These splitters can split the fiber anywhere from one to thirty-two times and by their very nature cause significant losses in the network. Therefore, their use is limited, taking into account the power load of the network. In a passive optical network, it would be desirable to use a PLC splitter between an optical splitter and individual end users in close proximity. Experimental measurements using this splitter show a signal loss of 10.612 dB at a wavelength of 1310 nm, and a loss of 10.522 dB at a wavelength of 1550 nm [8]. The model we offer uses a PLC splitter with a ratio of 1:4 and 1:8. Typically, PON can reach subscribers within 20 kilometers (km) of the original transmitter, which covers 98 percent of the population. A single 1x32 splitter has lower loss than 1x2 and 1x16 or 1x4 and 1x8 (2x2 and 2x16 or 2x4 and 2x8 in a 2x32 splitter network) cascaded splitters or any combination of 1x16, 1x8, 1x4 and 1x2 splitters [4].

There are three main types of PON technology: E-PON, A-PON and G-PON. The structural advantage of E-PON and G-PON is that they use two wavelengths by subscriber users. Statistical multiplexing efficiency is considered the most important performance factor because users have the same bandwidth.

PON infrastructure can be divided into three areas: central office, external area, subscriber area (customer premises). All active equipment in the PON network is located at the ends of the channels. The OLT (Optical Line Termination) device is located in the communication node, and the ONT device is located in the subscriber's area. Only fiber optic cable and signal splitters (splitters) passive components are placed between these devices. The structural scheme of the optical communication network is presented in Fig. 1.

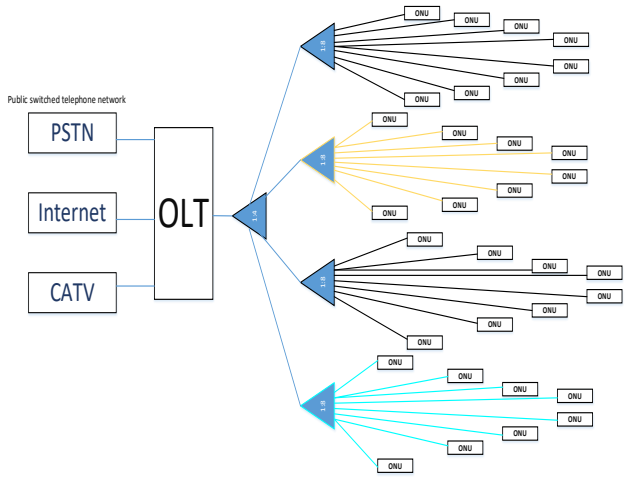


Fig. 1. The structure of the optical communication network

It is possible to detect any fiber communication faults in the PON network. A particular challenge in this regard is that a failure in one of the distributed optical cables must be accommodated without affecting service to other subscribers. To solve this problem, it is necessary to develop various network schemes and perform reliability calculations on network elements. According to the results of the analysis, it is necessary to organize the most optimal method of account books according to the reliability indicator of the communication network for small population areas.

Development of PON network reliability models and analytical expressions. The development of reliable schemes for the implementation of PON networks is still ongoing. In addition to the need to reduce capital costs per user, it is necessary to solve the problem of reducing operational costs associated with the construction of fault-tolerant access networks. This is the main task related to the creation of fault-tolerant and efficient access networks [11-13].

PON is based on point-to-multipoint architecture. Only one ONU can communicate with the OLT. Because of this, failures at different points affect the network differently. For example, the failure of a common intermediate optical fiber causes one ONU to fail, while the failure of the backbone fiber (the fiber between the OLT and the splitter) causes all ONUs to fail [15-17].

Determining reliability metrics is important in effective network design. PON reliability can be affected by various faults. A misconfigured network topology also affects the amount of failures. For this reason, it is necessary to take into account the type of topology when creating a reliability model of a passive optical network [18-20].

In developing models and analytical expressions for reliability estimation between OLT-ONU in passive optical networks in sparsely populated areas, we assume the following:

- Passive optical network works during normal use;
- Rejections and interruptions in a passive optical network obey the exponential distribution law.
- The rejection of any element of the passive optical network will lead to the rejection of the whole system.

A structural reliability diagram is a graphical representation of system reliability that can be used to visualize the effect of using different configurations and failures in the system. The structural diagram of a trustless PON network is shown in Figure 2. The passive optical access network architecture under consideration is described below.

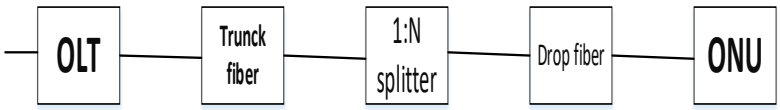


Fig. 2. General architecture of the PON network.

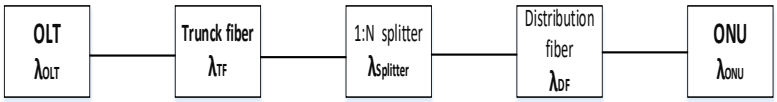


Fig. 3. Reliability structural model of the general architecture of the PON network

When the PON communication line is organized on the basis of optical cables, the rejection intensity of the main optical cable is determined by the expression (1):

$$\lambda_{ok1} = l_{TF} * \lambda_{TF} \quad (1)$$

l_{TF}-the length of the main cable of the PON line, the communication line is organized on the basis of optical cables;

λ_{TF}- Rejection intensity of 1 km long optical cable [1/h*km].

When the PON communication line is organized on the basis of intermediate optical cables, the rejection intensity of the optical cable is determined by the expression (2):

$$\lambda_{ok2} = l_{DF} * \lambda_{DF} \quad (2)$$

l_{DF}- the length of the PON line, where the communication line is organized on the basis of intermediate optical cables;

λ_{DF}- Rejection intensity of 1 km long optical cable [1/h*km].

The rejection intensity of the splitter of the PON communication line is determined by $\lambda_{splitter}$:

$$\lambda_{splitter} = \lambda_{splitter1} + \lambda_{splitter2} \quad (3)$$

where $\lambda_{splitter}$ is the rejection intensity of the splitter on the OLT and ONU side [1/hour].

The rejection intensity of the structural model of the general architecture is expressed by the following analytical expression.

$$\lambda_{total} = \lambda_{OLT} + \lambda_{ok1} + \lambda_{splitter} + \lambda_{ok2} + \lambda_{ONU} \quad (4)$$

The redundant passive optical network model consists of the use of redundant sets of OLT and ONU terminal devices, optical fiber lines, receiver modules and optical splitters. The scheme of passive optical network reserved for the first type of model, i.e. Splitter, is shown in Fig. 4. The OLT central node processes the duplicated signal and transmits a copy of the signal to the trunk. Downstream transmission occurs in a similar manner. When the receivers fail or the optical line is damaged, it is possible to switch over to backup devices and network components in time without interrupting the communication. Fig. 5 presents the reliability structural model of the redundant passive optical network architecture up to the splitter.

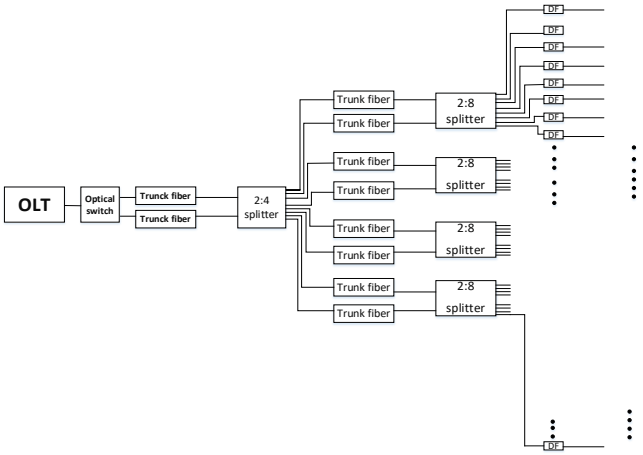


Fig. 4. Reserved passive optical network architecture up to the splitter.

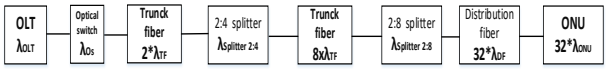


Fig. 5. Reliability structural model of passive optical network architecture reserved up to splitter

$$\lambda_{total} = \lambda_{OLT} + \lambda_{Os} + 2 * \lambda_{ok1} + \lambda_{splitter\ 2;4} + 8 * \lambda_{ok1} + +\lambda_{splitter\ 2;8} + \\ +32 * \lambda_{ok2} + 32 * \lambda_{ONU} \qquad (8)$$

The second type of model, i.e. passive optical network architecture reserved up to 2x16 splitter, is presented in Figure 6 and its reliability structural model is presented in Figure 7. To organize a network according to such an architecture, OLT, optical switch, main fiber optic cable, 2x2 optical splitter (optical splitter), 2x16 optical splitter, distribution fiber optic cable and ONU devices are needed.

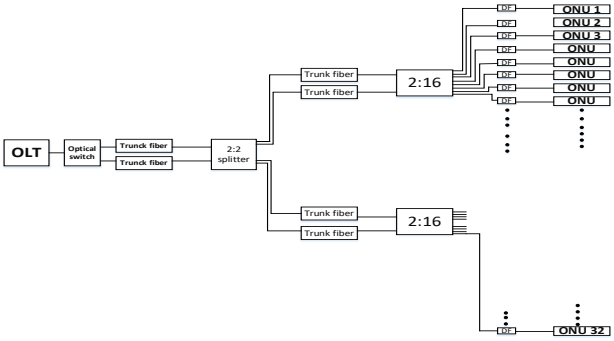


Fig. 6. Passive optical network architecture backed up to 2x16 splitter

Fig. 7. Reliability structural model of redundant passive optical network architecture up to 2x16 splitter

$$\lambda_{total} = \lambda_{OLT} + \lambda_{Os} + 2 * \lambda_{ok1} + \lambda_{splitter\ 2;2} + 8 * \lambda_{ok1} + \lambda_{splitter\ 2;16} + 32 * \lambda_{ok2} + 32 * \lambda_{ONU} \qquad (9)$$

We can see the general architecture of the third type of model, i.e. passive optical network organization reserved up to 2x32 splitter, in Fig. 8 below. To organize a network according to such an architecture, OLT, optical switch, trunk optical fiber cable, 2x32 optical splitter (optical splitter), distribution optical cable and ONU devices are needed.

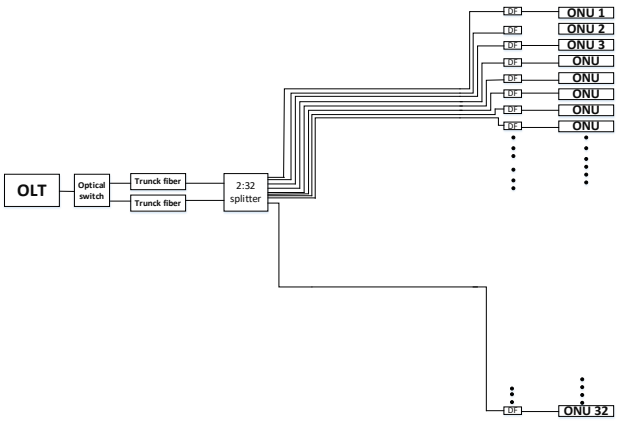


Fig. 8. Passive optical network architecture reserved up to 2x32 splitter

Fig. 9. Reliability structural model of redundant passive optical network architecture up to 2x32 splitter

$$\lambda_{total} = \lambda_{OLT} + \lambda_{Os} + 2 * \lambda_{ok1} + \lambda_{splitter\ 2;32} + 32 * \lambda_{ok2} + 32 * \lambda_{ONU} \qquad (10)$$

The unavailability ratio of the network is one of its most important indicators, and its determination is important for the assessment of capital and operating costs. For this reason, in this article, we calculate the unavailability coefficients for four cases according to the distance and grouping of subscriber users from the central data transmission center.

Cases:

Case 1. Location of users in sparsely populated areas: TF = 19 km; DF = 1 km.

Case 2. Location of users in sparsely populated areas: TF = 16 km; DF = 4 km.

Case 3. Dense location of users in sparsely populated areas: TF = 15 km; DF = 5 km.

Here TF-trunck fiber, DF-Distribution fiber.

Table 1 shows the reliability indicators of the network elements used in the object under consideration, and the calculations were made according to these indicators [9, 10].

Table 1. Reliability indicators of network elements

The name of the device	**Reliability indicator**
OLT	$5{,}12*10^{-7}$
Optical switch	$4*10^{-7}$
Optical splitter 2x2	$0{,}5*10^{-7}$
Optical splitter 2x4	$3*10^{-7}$
Optical splitter 2x8	$3{,}5*10^{-7}$
Optical splitter 2x16	$5*10^{-7}$
Optical splitter 2x32	$7{,}2*10^{-7}$
Truck fiber	$1{,}33*10^{-10}$ 1/m
Distribution fiber	$1{,}76*10^{-12}$ 1/m
ONU	$5{,}12*10^{-7}$

For most of the schemes presented, the unavailability factor does not depend on the total number of users in the system. However, backbone fiber optic cable in intermediate zones and, in contrast, distributed, i.e., non-backbone fiber optic cable, have different reliability values over distance. The total cost of reliability depends on the number of network backbone nodes organized according to reserved reliability schemes and the number of distributed users. Account books for such circuits are calculated for 32 users, for different values of the length of fiber optic cables of each reserved circuit. However, the unavailability coefficients for users belonging to different nodes are also different. Thus, the unavailability coefficients for all 3 cases are presented in Table 2.

Table 2. Results of unavailability ratios for 3 cases

Models	**Circumstances**
According to the 1st type model	Case 1: TF = 19 km; DF= 1 km.
	Case 2: TF = 16 km; DF = 4 km.

	Case 3: TF = 15 km; DF = 5 km.
According to the 2nd type model	Case 1: TF = 19 km; DF = 1 km.
	Case 2:TF = 16 km; DF = 4 km.
	Case 3: TF = 15 km; DF = 5 km.
According to the 3rd type model	Case 1: TF = 19 km; DF = 1 km.
	Case 2: TF = 16 km; DF = 4 km.
	Case 3: TF = 15 km; DF = 5 km.

Figures 9, 10, 11 show the results for three cases.

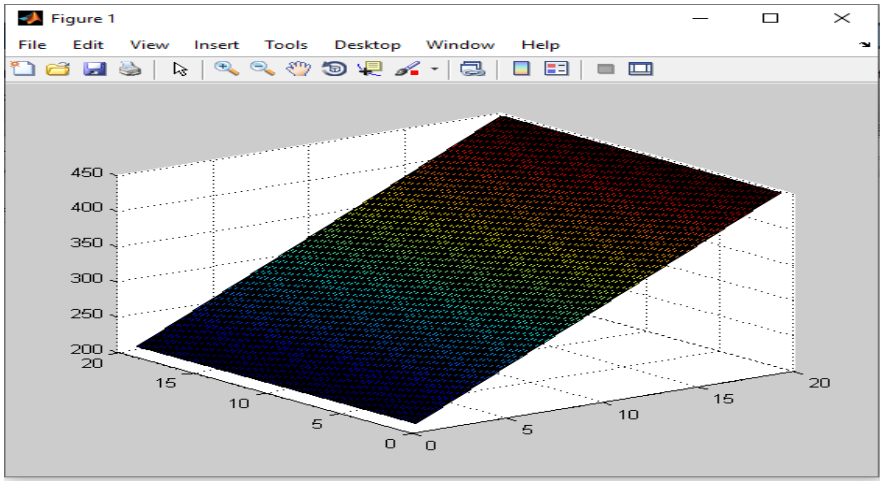


Fig. 10. Visual representation of calculations according to model 1

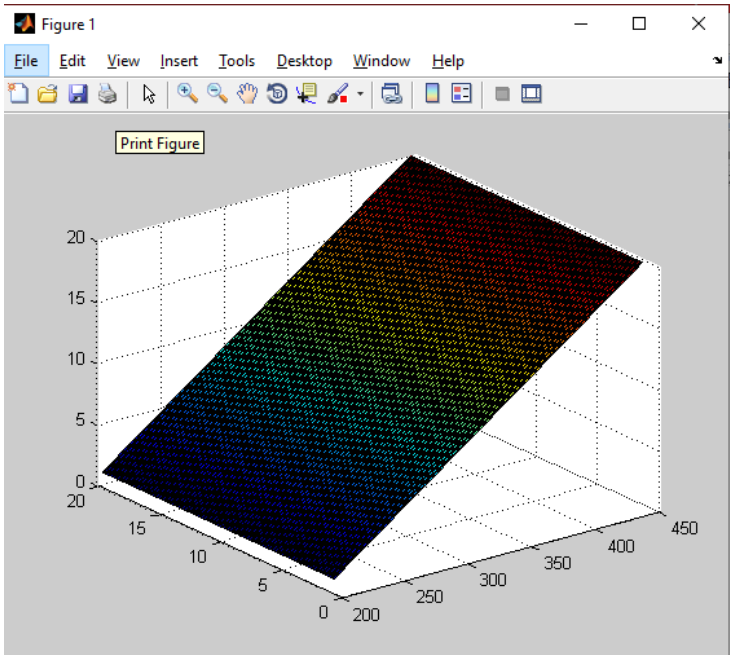


Fig. 11. Visual representation of calculations according to the 2nd model

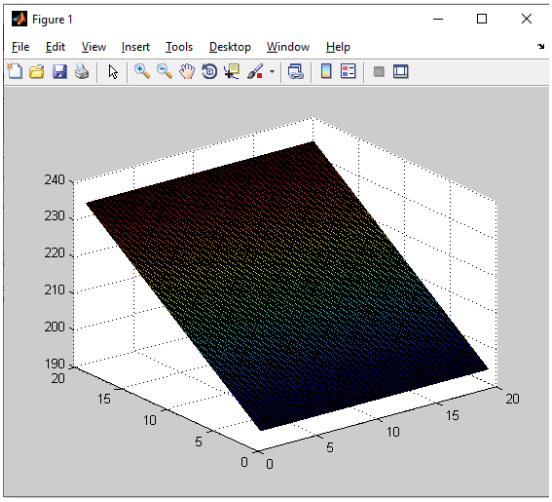


Fig. 12. Visual representation of calculations according to the 3rd model

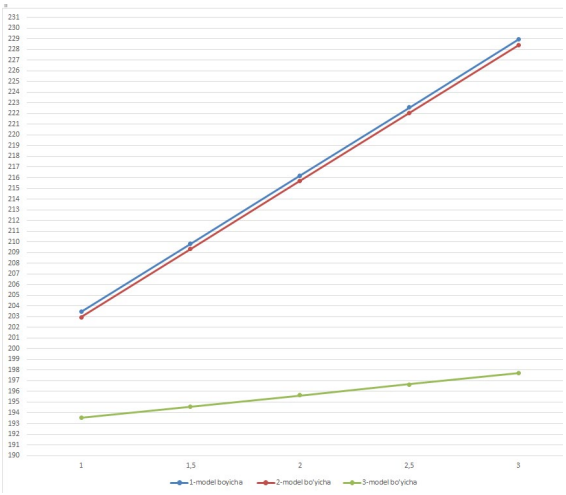


Fig. 13. Charts of each model's unavailability versus trunk line length

3 Conclusion

In the presented article, various network reliability schemes were considered and unavailability coefficients were determined for each of them. A comparative analysis of all presented schemes was conducted. Three reliability PON architectures are proposed using the models presented in this work. The three architectures differ in the degree of protection of the optical cables that divide the network from the ONU device to the front splitter. Type 1 reserved architecture is based on 2x4 and 2x8 splitter, type 2 is based on 2x2 and 2x16 splitter. Type 3 is organized on the basis of a 2x32 splitter. The above-mentioned architectures have a high-level optical splitter, and the length of the common communication line is 20 km. Based on the probable nature of optical fiber and device failures, analyzes were performed on connection availability and the number of connected customers. The actual level of connectivity improvement associated with deploying survivable architectures depends on a number of factors, including backbone fiber length and the type of survivable architecture. The number of failed connections or clients that are not connected depends not only on the type of architecture and the length of the trunk fiber to be maintained, but also on the ratio of splitting (splitters). Each of the additional reserved architectures has additional reliability per user. In conclusion, the proposed redundant type 3 PON architecture has been proven reliable compared to the other two types of redundant architecture models to meet the future needs of broadband customers and network carriers. Currently, survivable architectures that achieve high reliability performance to fully protect the network also depend on the length of the communication line to the residential customers and the splitter used for the communication network.

References

[1] Kavian Y.S., Leeson M.S. Resilient optical network design. Advances in fault-tolerant methodologies. USA: Information Science Reference. 2012, pp. 228–250.

[2] Farjady F. et al. First WP2 Progress Report. Available at: http://cordis.europa.eu/docs/projects/cnect/7/318137/08 0/ deliverables/001-DISCUSD22Final.pdf (07 October 2017))

[3] Ignatov A.V., Shuvalov V.P. The reliability of subscriber access networks LR-PON. T-Comm: Telekommunikatsii i transport [T-Comm]. 2015, vol. 9, no. 5, pp. 25–30. (In Russian)

[4] Gromov Yu.Yu., Ivanova O.G., Mosyagina N.G., Nabatov K.A. Nadezhnost' informatsionnykh sistem [Reliability of information systems]. Tambov: TGTU Publ., 2010, 160 p. (In Russian)

[5] Shuvalov V.P., Timchenko S.V. Metody rezervirovaniya i vosstanovleniya v tele-kommunikatsionnykh setyakh [Redundancy and recovery methods in telecommunication networks]. Nadezhnost' funktsionirovaniya i problemy informatsionnoi bezopasnosti telekommunikatsionnykh sistem zheleznodorozhnogo transporta: mezhvuzovskii tematicheskii sbornik nauchnykh trudov [Operation reliability and information security problems of telecommunication systems of railway transport: inter-university thematic collection of scientific articles]. Omsk: OmGUPS Publ., 2009, pp. 40–44. (In Russian)

[6] Egunov M.M., Shuvalov V.P. Structural reliability analysis of transport network. Vestnik SibGUTI [Proceedings of Siberian State University of Telecommunications and Information Science]. 2012, no. 1, pp. 54–60. (In Russian)

[7] Egunov M.M., Shuvalov V.P. Reservation and restoration in telecommunication networks. Vestnik SibGUTI [Proceedings of Siberian State University of Telecommunications and Information Science]. 2012, no. 2, pp. 3–10. (In Russian))

[8] Ivaniga T. et al. Optical Networks FTTx and Reduced Attenuation Balance with Passive Optical Splitter //Carpathian Journal of Electronic and Computer Engineering. – 2014. – Т. 7. – №. 1. – С. 30

[9] Wosinska L., Chen J., Larsen C. P. Fiber access networks: reliability analysis and Swedish broadband market //IEICE transactions on communications. – 2009. – Т. 92. – №. 10. – С. 3006-3014.

[10] Chen J. et al. Cost vs. reliability performance study of fiber access network architectures //IEEE Communications Magazine. – 2010. – Т. 48. – №. 2. – С. 56-65.

[11] Matyokubov U.K., Davronbekov D.A. The Impact of Mobile Communication Power Supply Systems on Communication Reliability and Viability and Their Solutions // International Journal of Advanced Science and Technology. Vol.29, No.5, (2020), pp.3374–3385. (Scopus). http://sersc.org/journals/index.php/IJAST/article/view/12023

[12] Davronbekov D.A., Matyokubov U.K. Reliability of the BTS-BSC System with Different Types of Communication Lines Between Them // International Journal of Advanced Trends in Computer Science and Engineering. Vol. 9, No. 4, (2020), pp. 6684–6689. (Scopus) http://www.warse.org/IJATCSE/static/pdf/file/ijatcse362942020.pdf DOI:10.30534/ijatcse/2020/362942020

[13] D.A.Davronbekov, U.T.Aliev, Sh.U.Pulatov, J.D.Isroilov, F.X.Fayzullaev. Features of Technologies for Transmission of Radio and Television in 4G / 5G Networks // 2021 International Conference on Information Science and Communications Technologies (ICISCT), DOI: 10.1109/ICISCT52966.2021.9670182

[14] D.A. Davronbekov; Z.T. Khakimov. Joint Application of A Runiing Wave Amplifier and Acousto-Optical Configurable Filter for Linearization of The Passage Spectral Characteristics of Focles // 2020 International Conference on Information Science and Communications Technologies, ICISCT - 2020. – p.1-4.

[15] Davronbekov D.A., Matyokubov U.K., Abdullaeva M.I. Evaluation of reliability indicators of mobile communication system bases // Acta of Turin Polytechnic University in Tashkent, 10(3):10, 2020. pp. 22-27.

[16] D.A.Davronbekov, U.K.Matyokubov. Influence of Communication Lines on Reliability in Mobile Communication Systems // 2021 International Conference on Information Science and Communications Technologies (ICISCT 2021), DOI: 10.1109/ICISCT52966.2021.9670377

[17] D.A.Davronbekov, U.K.Matyokubov. Algorithms for Calculating the Structural Reliability of a Mobile Communication System // 2021 International Conference on Information Science and Communications Technologies (ICISCT 2021), DOI: 10.1109/ICISCT52966.2021.9670315

[18] Davronbekov D.A., Matyokubov U.K. The role of network components in improving the reliability and survivability of mobile communication networks // Acta of Turin Polytechnic University in Tashkent, 10(3):10, 2020

[19] Davronbekov D.A., Matyokubov U.K. Calculate the reliability of the BTS-BSc network with the reserved line connection // Chemical Technology, Control and Management: Vol. 2021 : Iss. 1 , Article 11.

[20] Davronbekov D.A., Matyokubov U.K. The effect of the number of backup communication lines in the BTS-BSC system on reliability // Scientific Collection «InterConf», №40. Proceedings of the 2 nd International Scientific and Practical Conference «Scientific community: interdisciplinary research», January 26-28, 2021, Hamburg, Germany: 2021. - pp. 679-684.

[21] Tanwar, Sudeep & Khujamatov, Halim & Turumbetov, Bairam & Reypnazarov, Ernazar & Allamuratova, Zamira. (2022). Designing and Calculating Bandwidth of the LTE Network for Rural Areas. International Journal on Advanced Science, Engineering and Information Technology. 12. 437. 10.18517/ijaseit.12.2.14950.

[22] Davronbekov, D., Isroilov, J., & Akhmedov, B. (2019). Principle Of Organizing Database Identification On Mobile Devices By Imei. 2019 International Conference on Information Science and Communications Technologies (ICISCT), 1-5.

[23] Khujamatov, Kh.E & Toshtemirov, T & Lazarev, A.P. & Raximjonov, Q.T.. (2021). IoT and 5G technology in agriculture. 1-6. 10.1109/ICISCT52966.2021.9670037.

Application of neural networks in the field of information security

Mahmudjon Sadikov[1], Bahrom Sobirov[1]

[1]*Urgench branch of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, al-Khorezmi street 110, Urgench city (Uzbekistan)*

imaxmaxmud@gmail.com, ggalaktika5@gmail.com

Abstract.

This article explores the application of neural networks in the field of cryptography, in the context of the emerging field known as "Neurocryptography". Neurocryptography is the branch of cryptography that deals with the use of neural network algorithms in encryption and cryptanalysis.

This paper presents a historical perspective on the advent of neurocryptography, starting with research by Wolfgang Kinzel and Ido Kantler, which demonstrated the learning of two neural networks through interaction with each other. The experiment showed that two neural networks could work in harmony, producing complementary but opposite results, which became the basis for their use as encryption keys.

Since then, various types of neural networks have been developed, such as "End of the circle neural network", "Feedback neural network", and "Chaotic neural network", each with its own characteristics and applications.

Neural networks are currently being used effectively in various areas of cryptography, including the development of public key cryptosystems, encryption algorithms, cryptanalysis, shorthand, and hash functions. The article discusses in detail the application of neural networks in each of these areas.

Keywords: linear law of the lesson, quasilinear method, Sweep method

1 Introduction

Today, the development and automation of information technologies, the implementation of artificial intelligence technologies, and the usage of neural network technologies to solve complex tasks contribute significantly to the conquest of science. In the field of cryptography, the application of neural networks and their use in solving cryptographic problems is gradually expanding. With the use of neural networks in cryptography, the term "Neurocryptography" entered the science. Neurocryptography is a branch of cryptography that studies the use of neural network algorithms in encryption and cryptanalysis.

The first information about interconnected neural networks in the field of cryptography is presented in the article "Neural networks find application in cryptography". It talks about the necessity of the process of learning and gaining experience for neural networks and the practice that two neural networks carry out this process through each other. Wolfgang Kinzel, an employee of the "Institute of Theoretical Physics" in Wolsburg, Germany and Ido Kantler, a researcher at the "Minerva" Center Israel Ramat-Gane will run neural networks with reversely connected input and output parts through a computer network. In the process of training neural networks, the task of unique categorization is set, presenting random fragments of information, the results of which are clear. After each round, the results were tallied. Soon the two networks began to work in harmony, with the networks producing complementary but opposite results. Then the results of one neural network were reversed and the two neural networks began to work in a mutually compatible manner. The result of this experiment was useful to the field of cryptography. Because of this, currently, encrypted information is transmitted through computer networks, and the key needed to solve the ciphertext is also sent through this network. This creates a risk of stealing the key itself and cracking the ciphertext. And these two interconnected neural networks can use their information as a key, because these two neural networks operate in a mutually synchronous and compatible manner.

As a result of the development of neurocryptography, neural networks of the type "End of the circle neural network", "Feedback neural network", "Chaotic neural network" with their own characteristics and work have appeared.

Neural networks are currently being used effectively in the following areas of cryptography:

• the development of open-key cryptosystems.

• encryption algorithms.
• cryptanalysis.
• shorthand.
• hash functions.

This article provides information on how neural networks can be used in the fields listed above.

2 Main part

The working principle of the neural network generating public keys is given below. This neural network is two-stage and consists of n networks and n^2 output neurons. Each outgoing neuron is a certain element of the matrix $D=MB+C$. The value of the given matrix P is the connection value of the neural network, and the input signal is λ_i.

For example, let's consider a cryptosystem that encrypts information consisting of three components. The components of the cryptosystem are defined.

$$B=\begin{matrix}5&3&1\\4&7&8\\11&2&10\end{matrix}\quad P=\begin{matrix}2&3&5\\4&7&1\\3&7&9\end{matrix}\quad C=\begin{matrix}11&8&7\\4&5&3\\6&13&3\end{matrix}$$

In this $m_1 = 83 > (2 + 4 + 3)^2 = 81$, $m_2 = 293 > (3 + 7 + 7)^2 + 289$,
$m_3 = 227 > (5 + 1 + 9)^2 = 225$
Modulo multiplication $m=m_1m_2m_3$ =5 520 413 and a constant is W=64.
Then $M=mW-1=353\ 306\ 431$.

Selected m_i based on which a constant outgoing signal is determined λ = (4 256 704, 1 205 824, 1 556 416). Updating the system parameters for the mentioned modules can be done by changing the coefficients in the given range, the sum of the components and the square of the sum of the elements of the matrix C.

To generate the key A, we need to determine the connection coefficients of the neural network. In the general case, the coefficients of the matrix W consist of n rows and n^2 columns. In this example, n = 3, the matrix W for each component of A has 3 rows and 9 columns.

$$W = \begin{pmatrix} p_{11}p_{11} & p_{11}p_{12} & p_{11}p_{13} & p_{11}p_{21} & p_{21}p_{12} & p_{21}p_{13} & p_{11}p_{31} & p_{31}p_{12} & p_{31}p_{13} \\ p_{12}p_{21} & p_{12}p_{22} & p_{12}p_{23} & p_{22}p_{21} & p_{22}p_{22} & p_{22}p_{23} & p_{32}p_{21} & p_{32}p_{22} & p_{32}p_{23} \\ p_{13}p_{31} & p_{13}p_{32} & p_{13}p_{33} & p_{23}p_{31} & p_{23}p_{32} & p_{23}p_{33} & p_{33}p_{31} & p_{33}p_{32} & p_{33}p_{33} \end{pmatrix}.$$

Figure 1 shows the structure of a public key computation neural network. For the example above, it looks like this.

$$A = \begin{pmatrix} 1\,833\,321\,798 & 1\,127\,077\,317 & 474\,207\,686 \\ 1\,480\,383\,744 & 2\,601\,894\,078 & 2\,950\,625\,019 \\ 4\,007\,271\,995 & 830\,786\,443 & 3\,756\,529\,721 \end{pmatrix}.$$

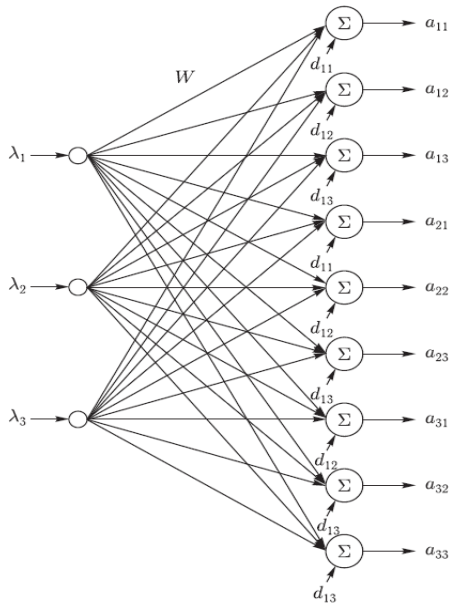


Figure 1. Structure of public key generation neural network

"Deep Hashing learning", which has entered the field of hash functions, is a new generation of hash functions, and the use of such hash functions is expanding in cryptography. There are also two types of deep hashing, symmetric and asymmetric. They have been improved and continue to be developed using artificial neural networks in the hashing process.

3 Conclusion

Based on the above information, it can be said that artificial neural networks are taking bold steps in the field of cryptography. Neural networks can be used not only in any field of cryptography, but also in any other field. Developing new areas, such as Deep Learning, Turing Neural Machine, Differential Neural Computer (DNC), Deep Hashing (Deep hashing) and other areas require the use of technologies in cryptographic issues. It is necessary to create two interconnected neural networks for data encryption and transmission through channels, and to train two neural networks in such a way that no other system can understand them. Information is exchanged between two neural networks through a protocol known only to them. The protocol is created when neural networks train each other. Open data is provided to the input stream of the neural network.

References

[1] Eisenstein E., Kanter I., Kessler D.A., Kinzel W. Generation and Prediction of Time Series by a Neural Network // Phys. Rev. Lett. 1995. V. 74. P. 6–9.

[2] Червяков Н.И., Евдокимов А.А., Галушкин А.И., Лавриненко И.Н., Лавриненко А.В. Применение искусственных нейронных сетей и системы остаточных классов в криптографии. Москва – 2012г.

[3] Бекмуратов Т. Ф., Мухамедиева Д.Т. Теория, методы и алгоритмы синтеза нейро-нечетких моделей принятия решений при интеллектуальном анализе данных. Ташкент-2016г.

[4] Чмора А.Л. Современная прикладная криптография. — М.: Гелиос АРВ, 2017. — 256 с.

[5] Ростовцев А. Г., Маховенко Е. Б. Введение в криптографию с открытым ключом. — СПб. Мир и семья, Интерлайн, 2013. — 356 с.

[6] Саломаа А. Криптография с открытым ключом: Пер. с англ. — М.: Мир, 1995. — 318 с.

[7] Kinzel W., Kanter I. Interacting neural networks and cryptography // Advances in Solid State Physics. Springer Verlag, 2002. V. 42. P. 383–391.

[8] Turdiev T. T; Ismailov Sh; Akbar B. Parallel Computational Algorithms for Solving Boundary Value Problems for Two-Dimensional Equations of Parabolic Type 2021. https://ieeexplore.ieee.org/abstract/document/9670272

[9] Madaminov U.A, Sadikov M.A, Kutliev S.P, Allaberganova M.R . Development and application of computer graphics training software in information technology. 2021 International Conference on Information Science and Communications Technologies (ICISCT)

[10]Olimov I, Boriyev Y, Sadikov M, Xudoyberdiyev A, Ismanaliyev J. Analysis of existing standards for information security assessment. 2021 International Conference on Information Science and Communications Technologies (ICISCT).
[11]Sadikov M.A, Olimov I.S, Karimov A.A, Tursunov O.O, Yusupova Sh.B, Tojikabarova U.U. A Creation Cryptographic Protocol for the Division of Mutual Authentication and Session Key. 2020 International Conference on Information Science and Communications Technologies (ICISCT)
[12]Sadikov M.A, Olimov I.S, Internet of things architecture and security challenges. 2020 International Conference on Information Science and Communications Technologies (ICISCT)

Методы извлечения знаний о ситуациях и событиях из текста естественного языка

Архипова Влада Игоревна

Новосибирский государственный университет, Россия.

v.arkhipova1@g.nsu.ru

Рассматривается проблема извлечения знаний о ситуациях и событиях из текста естественного языка. Методы извлечения информации из текста становятся все более актуальными в условиях необходимости эффективной обработки больших объемов информации. Исследуются различные подходы к извлечению знаний, включая использование семантических технологий и онтологических моделей.

Ключевые слова: естественные языки, извлечение знаний, онтология

В настоящий момент в Интернете представлен колоссальный объем информации, которую трудно извлекать в структурированном виде. [1] Целью работы является автоматизация извлечения информации, представленной в неструктурированных текстах естественного языка, для разработки онтологий различных предметных областей. В работе описаны методы извлечения знаний о ситуациях и событиях из текста естественного языка [2, 4, 5]. Эти методы основаны на формальном представлении знаний при помощи фрагментов атомарных диаграмм алгебраических систем. [3]

Извлечение знаний о ситуациях и событиях из текста основано на создании онтологических моделей [4], которые описывают семантические отношения между сущностями и событиями [6]. Эти модели используют RDF для представления данных и OWL для определения семантических отношений и ограничений [7, 8]. Затем SPARQL применяется для выполнения сложных запросов к этим данным, позволяя точно извлекать информацию о ситуациях и событиях.

Результаты этой работы имеют потенциал повысить качество и эффективность интеллектуальных помощников, которые будут способными обогащать знания и предоставлять пользователю более точные и информативные ответы. Таким образом, с использованием семантических технологий и методов извлечения знаний о ситуациях и событиях, мы можем усовершенствовать процессы анализа и понимания текстовых данных, что имеет важное значение в современной информационной среде.

Литература

[1] Власов Д.Ю., Пальчунов Д.Е., Степанов П.А. Автоматизация извлечения отношений между понятиями из текстов естественного языка // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2010. Т. 8. № 3. С. 23-33.

[2] Dmitry E. Palchunov. Virtual catalog: the ontology-based technology for information retrieval. In: Knowledge Processing and Data Analysis. LNAI 6581, Springer-Verlag Berlin Heidelberg, 2011, pp. 164–183.

[3] Деревянко Д.В., Пальчунов Д.Е. Формальные методы разработки вопросно-ответной системы на естественном языке // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2014. Т. 12. № 3. С. 34-47.

[4] A. Orlovsky and D. Palchunov. Development of Automated Methods for the Domain Ontology Population with the Help of a Virtual Assistant, 2021 IEEE 22nd International Conference of Young Professionals in Electron Devices and Materials (EDM), 2021, pp. 537-541, doi: 10.1109/EDM52169.2021.9507641.

[5] D. E. Palchunov. Methodological Aspects of the Application of Model Theory in Knowledge Engineering and Artificial Intelligence, 2022 Ural-Siberian Conference on Computational Technologies in Cognitive Science, Genomics and Biomedicine (CSGB), 2022, pp. 210-215. doi: 10.1109/CSGB56354.2022.9865602.

[6] Allemang, D., & Hendler, J. (2008). Semantic Web for the working ontologist. Morgan Kaufmann Publishers.

[7] Методы извлечения информации из текста [Электронный ресурс]. (2019). Режим доступа: https://cyberleninka.ru/article/n/metody-izvlecheniya-informatsii-iz-teksta/viewer

[8] RDF Primer [Электронный ресурс]. (2004). World Wide Web Consortium. Режим доступа: https://www.w3.org/TR/rdf-primer/

Извлечение эмоциональных оценок из текстов естественного языка

С. А. Балашова

ФГБУН Институт математики им. С. Л. Соболева СО РАН, 630090, г. Новосибирск, пр. Академика Коптюга, 4

s.balashova@g.nsu.ru

Аннотация. *Эмоции играют важную роль в жизни человека, а их анализ представляет собой особый интерес. Одним из важных аспектов при извлечении эмоциональных оценок является передача эмоционального состояния автора текста, его отношение к происходящему в тексте. В данной статье рассматриваются основные подходы, используемые для извлечения эмоций из текста.*

Ключевые слова: распознавание эмоций, машинное обучение, описание эмоций, обработка естественного языка

1 Введение

Эмоции играют важную роль в жизни людей, как для одного человека, так и для группы людей, живущих или работающих вместе. Различные медиа предоставляют людям возможности поделиться своими эмоциями. Анализ эмоций может использоваться с целью прогнозирования политической ситуации, определения группы людей со сходными интересами, извлечения эмоционального отклика в отзывах различных потребителей, настраивания работы различных информационных систем и т.д.

Сложность определения эмоций в тексте обосновывается следующими причинами:

1) Выражение эмоций сильно зависит от контекста.

2) Частота слов важнее их расположения.

3) Использование различных стилистических приемов.

2 Эмоциональные оценки

На сегодняшний момент тема распознавания эмоций в тексте очень актуальна. В рамках этой темы можно выделить некоторые задачи:

1) Анализ тональности текста - выявление в тексте эмоционально окрашенной лексики, то есть выражений, которые позволяют выразить личные отношения человека к поступкам, действиям и вещам. Слова в данных выражениях отражают чувства, эмоции и настроения и несут в себе негативную либо позитивную оценку смысла. К примеру, такие слова

как выродок, балбес, набедокурить, имеют отрицательную оценку. А благородный, куколка - положительную.

Эмоционально окрашенная оценочная лексика не может встречаться в текстах, написанных в официально-деловом и научном стилях. Но ее можно увидеть в разговорной речи или материалах, написанных в публицистическом стиле [1].

2) Анализ субъективности - определить, субъективен ли текст, то есть предназначен для информирования об определенном факте без отражения мнения автора, или объективен, то есть выражает мнения, чувства или идеи автора.

Эти подходы являются основными задачами анализа эмоциональной окраски текста. Первый каждому слову присваивает оценки: +1 и -1 в случае позитивной и негативной оценки соответственно. Второй подход определяет текст, подлежащий анализу, а именно тот, в котором выражается мнение автора.

3) Классификация по эмоциям - определить, какая конкретная эмоция выражена в тексте. Исследования Пола Экмана показывают, что используются 6 базовых эмоций: радость, злость, отвращение, страх, грусть и удивление.

4) Извлечение мнений - определить, кто по отношению к кому или чему выразил мнение и какова тональность этого мнения.

Основываясь на классификации Экмана по основным эмоциям, можно выделить краткое их описание:

- радость: состояние, вызванное ощущением удовольствия, которое возникает от восхищения чего-то красивого и необычного, от наслаждения неудачи другого человека, от гордости за свои достижения.

- злость: состояние, вызванное неуспехом, сильным неудобством, разочарованием на многократные неудачи.

- отвращение: состояние, возникающее на ужасный вкус или запах, которое побуждает избегать что то неприятное.

- грусть: состояние, которое меняется от уныния и безнадежности, до страданий и чувств утраты.

- удивление: состояние, которое возникает на неожиданные события.

- страх: состояние, возникающее на ужасные и неприятные ситуации в жизни, а также на возможную опасности [2].

3 Обработка естественного языка

Для решения проблемы исследования эмоциональных оценок из текстов естественного языка используется искусственный интеллект, в частности машинное обучение и обработка естественного языка. Обработка естественного языка дает компьютерам возможность интерпретировать, манипулировать и понимать человеческий язык, а также позволяет применять алгоритмы машинного обучения для текста и речи.

Предварительно текст обрабатывается, что включает в себя:

- Токенизация – это процесс разделения текста на компоненты;

- Нормализация. Процесс предназначен для того чтобы убрать из исходного текста грамматическую информацию, оставляя смысловую составляющую. Два основных алгоритма нормализации это стемминг и лемматизация:

- Стемминг – упрощенный алгоритм морфологического разбора слова;

- Лемматизация – более тонкий процесс, который использует словарь и морфологический анализ, целью которого является привести слово к его канонической форме – лемме.

- Фильтрация. При анализе текста его лучше перевести в низкий регистр (за исключением собственных имен и аббревиатур), удалить знаки препинания, удалить неинформативные слова;

- Векторизация - преобразовать в числа и выполнить извлечение признаков [3].

После используются различные подходы извлечения эмоциональной оценки.

Можно выделить несколько подходов:

1. Основанный на лексике (классифицирует текст используя имеющий лексикон)
2. Основанный на ключевых словах (сопоставление шаблона и ключевых слов)
3. Машинное обучение (обучение классификатора с учителем или без).
4. Смешанный подход (комбинирует 2 или 3 подхода) [2].

Для обучения алгоритмов машинного обучения необходима выборка, которая нужна для тренировки данной модели. Выборку можно разбить на 7 классов эмоций: шесть базовых эмоций по Полу Экману и седьмой - нейтральный.

После обработки текста идет построение вектора слов на основе заранее подготовленного словаря. Затем этот вектор попадает в нейронную сеть, которая может работать в двух режимах: обучения и предсказания. В режиме обучения подается тип класса, к которому относится данный текст, на основе чего классификатор должен обучиться. В режиме предсказания программа пытается предсказать класс поданного текста на основе предыдущего текста [4].

Существуют методы для более глубокого анализа, которые извлекают эмоции из текстов - модель OCC для извлечения эмоций. Данная модель извлекает эмоции из английских текстов, но также может быть использована для извлечения эмоций из русскоязычных текстов. Модель OCC позволяет распознавать 22 типа эмоций, идентифицируя их путем оценки реакций на события, агентов или объектов, содержащихся в анализируемом тексте [5].

В дальнейшем планируется разработка системы, предположительно на нейронных сетях, которая будет обрабатывать текст естественного языка, а также давать ему эмоциональную оценку. Данная система, в отличие от других, будет показывать то, что вызывает определенную эмоцию, а также рассматривать такие случаи, когда одна ситуация вызывает несколько различных эмоций.

4 Заключение

Проблема распознавания эмоций в текстах является в настоящее время весьма важной и актуальной. Для ее решения используются различные методы и алгоритмы, в частности машинное обучение и обработка естественного языка, которая в первую очередь направлена на повышение качества машинного перевода и его анализ. Все они обладают своими достоинствами и недостатками, но комбинирование методов позволяет более точно извлекать эмоциональные окраски из текстов естественного языка, тем самым помогает настраивать информационные системы, а также передавать эмоциональные состояния в тексте.

Литература

[1]ЛитРес: Эмоционально окрашенная лексика: сайт. - URL: https://services.selfpub.ru/poleznoe-dlja-pisatelej/jemocionalno-okrashennye-slova (дата обращения 22.09.2023)

[2] Дворников С. В. Распознавание эмоций в текстовом сообщении // CYBERLENINKA: сайт. - URL: https://cyberleninka.ru/article/n/raspoznavanie-emotsiy-v-tekstovom-soobschenii/viewer (дата обращения 22.09.2023)

[3] Золушкин Ю. А. Обработка естественного языка // Библиотека материалов по теме выпускной работы: сайт. - URL: https://masters.donntu.ru/2021/fist/zolushkin/library/article1/1.htm (дата обращения 22.09.2023)

[4] Изотова Е. Д. Интеллектуальное извлечение эмоционального отклика потребителей лекарственных средств из отзывов на естественном языке // CIBERLENINKA: сайт. - URL: https://cyberleninka.ru/article/n/intellektualnoe-izvlechenie-emotsionalnogo-otklika-potrebiteley-lekarstvennyh-sredstv-iz-otzyvov-na-estestvennom-yazyke/viewer (дата обращения 22.09.2023)

[5] Слапогузов А. П., Цопа Е. А. Извлечение эмоций из русскоязычных текстов // Конгресс молодых ученых: сайт. - URL: https://kmu.itmo.ru/digests/article/2329 (дата обращения 22.09.2023)

Реализация графического инструмента трассировки системных вызовов на основе gVisor

Э. А. Вартазарян

Новосибирский государственный университет, ул.Пирогова, д. 1, г. Новосибирск, 630090, Россия

e.vartazaryan@g.nsu.ru

В условиях современности, где скорость работы программного обеспечения играет важную роль, разработка графического инструмента для удобной трассировки системных вызовов представляет собой значимый шаг в развитии инструментария разработчика.

Предлагаемый инструмент даст возможность быстро и наглядно просматривать системные вызовы, выявлять возможные ошибки и проводить оптимизацию кода. В дополнение к этому, этот инструмент также будет полезен для анализа вредоносного ПО, которое также использует системные вызовы для совершения атак. Традиционные инструменты трассировки, такие как strace, не всегда удобны для обыденного использования и могут быть легко обнаружены вредоносным ПО. Данный инструмент будет интегрирован в gVisor, что позволяет “маскироваться” как запуск обычного контейнера и оставаться незамеченным для злоумышленника.

Целью данной исследовательской работы является разработка графического инструмента, интегрированного в gVisor, для анализа системных вызовов. Для достижения этой цели были поставлены следующие задачи: исследование существующих решений, разработка прототипа описанного инструмента и его тестирование.

В рамках проведенной работы был успешно разработан прототип этого инструмента, предоставляющий возможность анализировать системные вызовы через веб-интерфейс. В настоящий момент он умеет подключаться к gVisor, принимать его логи, имеет функционал для настройки фильтров для просмотра интересующих системных вызовов и возможность детально их изучать, например, узнать возвращаемые значения.

Благодарности

Работа выполнена при поддержке АО "СберТех" и ПАО "СБЕРБАНК" на базе студенческой учебно-научной лаборатории факультета информационных технологий Новосибирского государственного университета.

Знания-Онтологии-Теории (ЗОНТ-2023)

Разработка методов определения эмоций пользователя на основе анализа текстовых и звуковых данных

Гуртуева Ольга Александровна

Новосибирский национальный исследовательский государственный университет, 630090, г. Новосибирск, ул. Пирогова, 2

o.gurtueva@g.nsu.ru

Аннотация. *Цель данной работы состоит в изучении существующих методов анализа эмоциональной экспрессии текстовых и звуковых данных естественных языков. В рамках исследования предполагается создание и разработка методики для определения тональности текста и эмоциональной выразительности звука, которая будет использоваться для анализа тональности диалогов с пользователями на основе имеющейся выборки данных.*

Ключевые слова: анализ тональности, семантический анализ, онтология, обработка естественного языка

1 Введение

Речь является основным и непосредственным способом взаимодействия между людьми. Она содержит в себе большое количество различной информации и может передавать богатый спектр эмоций, позволяющих наиболее точно воспринимать говорящего.

Автоматическое распознавание эмоций путем анализа человеческого голоса и выражения лица стало предметом многочисленных исследований благодаря тому, что системы автоматического распознавания применимы в различных областях и для различных целей. Особенно актуальной эта задача стала в связи с автоматизацией большинства процессов и широким использованием виртуальных помощников. Применение методов анализа эмоций при взаимодействии чат-ботов с пользователями позволяет наиболее точно выстраивать сценарии для общения, производить оценку эффективности их работы: формировать различный отчеты, производить оценку по заданным метрикам (например, об удовлетворенности пользователя взаимодействием с ботом) и строить различные графики.

В данной работе рассматривается задача построения подобного интерфейса в контексте интеллектуального помощника, основанного на трехуровневой семантической модели пользователя, представленного в системе в виде знаний, представлений о текущей ситуации и намерений.

2 Постановка задач

Целью настоящей работы является исследование существующих подходов к решению задач анализа эмоций пользователя, а также разработка программной системы, осуществляющей автоматическое определение эмоций на основе текстовых и звуковых данных с помощью анализа речевого сигнала (интонации голоса и т.п.), а также анализа тональности текста (сентимент-анализ, англ. Sentiment analysis), которая позволяет в любой момент времени отображать спектр эмоциональных параметров пользователя.

Для выполнения данной цели были поставлены следующие задачи:

1. Исследование особенностей взаимодействия пользователей с чат-ботами, анализ существующих подходов автоматизированного определения эмоций.
2. Поиск данных для обучения системы.
3. Определение метрики для оценки эффективности работы системы.
4. Построение программной архитектуры системы.
5. Разработка системы и оценка ее качества.
6. Разработка способов улучшения метрики.

3 Заключение

В ходе проведенной работы была разработана система распознавания эмоций пользователя в рамках работы виртуального помощника. По результатам проведенных экспериментов оценка точности данной системы по выбранной метрике составила 78%, что является хорошим результатом среди аналогов, однако не является пределом.

Литература

[1] Пальчунов Д. Е., Целищев В. В. Проблема извлечения знаний в системе взаимодействия человека и компьютера (онтологии и пресуппозиции) // Философия науки. 2012. No 4 (55). С. 20–35.

[2] Панг Б. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval / Б. Панг, Л. Ли. – Москва : Вильямс, 2008. – 235 с

[3] Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие / Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В. — М.: МИЭМ, 2011. — 272 с.

[4] Махина, Е.Д., Пальчунов, Д.Е. Программная система для определения речевых действий в текстах естественного языка. Вестник Новосибирского государственного университета. Серия: Информационные технологии, 2018. Т.16, №4 — 2018, № 4. — С. 96–102.

[5] Sumathi, C. P. Automatic facial expression analysis a survay / C. P. Sumathi, T. Santhanam, M. Mahadevi // International Journal of Computer Science & Engineering Survey (IJCSES). — 2012. — Vol. 3, No 6. — P. 47-57.

[6] Neurodata Lab [Электронный ресурс]. – URL: neurodatalab.com/ (дата обращения: 13.02.2022)

Знания-Онтологии-Теории (ЗОНТ-23)

Применение опроса для рекомендации косметической продукции с учетом индивидуальных черт пользователя

Демчук Марина Евгеньевна

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия
Novosibirsk State University, 2 Pirogova st., Novosibirsk, Russia

m.demchuk@g.nsu

Основная задача рекомендательной системы - понять предпочтения пользователя и предложить ему самый подходящий товар или услугу [1]. Сегодня такие системы встречаются в каждой сфере. Так и в индустрии косметических и уходовых средств. Обычный человек теряется среди огромного многообразия косметических и уходовых товаров, поэтому тратит большое количество денег и времени на поиск подходящего продукта.

Современные рекомендательные системы основываются на семантическом анализе предметной области [2, 3]. Такие системы, основываясь на оценках и отзывах пользователей [4], предоставляют самый популярный продукт, однако эти системы не учитывают особенности человека и то, что совместное использование продуктов может быть гораздо эффективнее.

Данная работа посвящена разработке рекомендательной системы, которая будет основываться на особенностях пользователя и рекомендовать ему подходящий продукт или набор продуктов. В зависимости от средства, который ищет пользователь, генерируется опрос, который включает в себя вопросы, способные выделить индивидуальные черты пользователя. Зная об особенностях человека, мы можем предложить ему продукт, который будет наиболее подходящим. Также будут рекомендованы средства, которые в наборе дадут самый лучший результат. В итоге, мы решаем проблему “самого популярного товара” и основываемся на особенностях конкретного пользователя, а не на оценках огромного количества разных людей.

После применения алгоритма анализа к ответам пользователя, система выдаст подходящие ему продукты, а также сохранит предпочтения, чтобы в дальнейшем быстрее выдавать результат.

Литература

[1] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor, Recommender Systems Handbook \\ Springer New York, 2010.

[2] Yakhyaeva G.E. Application of Boolean Valued and Fuzzy Model Theory for Knowledge Base Development \\ SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings, 2019, стр. 868–871.

[3] Yakhyaeva, G., Application of the Case Models Restriction for Modeling Argumentation Reasoning \\ 2021 International Symposium on Knowledge, Ontology, and Theory, KNOTH 2021, 2021, pp. 40–44.

[4] Yakhyaeva G., Skokova V. Subjective Expert Evaluations in the Model-Theoretic Representation of Object Domain Knowledge \\ Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2021, 12948 LNAI, стр. 152–165.

Разработка методов создания модели пользователя интеллектуального помощника

Дробышевский Иван Александрович

ФГБУН Институт математики им. С. Л. Соболева СО РАН, 630090, г. Новосибирск, пр. Академика Коптюга, 4

i.drobyshevskii@g.nsu.ru

Аннотация. Новизна заключается в создании семантической модели, которая объединяет в себе модель пользователя, сервис для обработки естественного языка Dialogflow, использующий нейронные сети и машинное обучение, и онтологическую модель предметной области. Целью данной работы является разработка интеллектуального помощника на основе модели и знаний о намерениях пользователя на примере помощника секретаря кафедры. В процессе работы были выполнены следующие задачи: обзор предметной области, анализ существующих аналогов, разработана программная система общения с пользователем, разработана трехуровневая модель пользователя, произведено тестирование системы. Обработка естественного языка и выявление намерений осуществлялась посредством сервиса Dialogflow. Хранение базы знаний пользователя производится в онтологической модели, представленной в ВКР студентки Олейник. А. О., куда отправляются запросы для получения информации. В результате работы была получена система для общения с пользователем, которую планируется внедрить на кафедру общей информатики для автоматизации бизнес-процессов кафедры, связанных с запросом, получением и предоставлением кафедральных документов студентам и научным руководителям.

Ключевые слова: модель пользователя, интеллектуальный помощник, обработка естественного языка, онтологический подход.

1 Введение

В настоящее время огромное количество людей по всему миру пользуются интеллектуальными помощниками, такими как Siri, Алиса и т.д. [1, 2]. Но каждая беседа с таким ассистентом происходит как общение с новым человеком. Ассистент начинает новый разговор, совершенно забывая про старый. Это в свою очередь вызывает необходимость пояснительных слов со стороны пользователя и дает неточные результаты [3].

Существующие похожие решения реализованы в различных виртуальных помощниках. Однако современные интеллектуальные помощники не запоминают информацию при общении с пользователем, а лишь могут поддержать контекст, соотнеся следующий запрос с предыдущим.

Целью данной работы является разработка интеллектуального помощника на основе модели и знаний о намерениях пользователя на примере помощника секретаря кафедры. Предложенная система будет основываться на созданной семантической модели, которая объединяет в себе модель пользователя, сервис для обработки естественного языка Dialogflow, использующий нейронные сети и машинное обучение, и онтологическую модель предметной области [4, 5].

2 Теоретические проблемы

2.1 Постановка задачи

Для достижения данной цели были поставлены следующие задачи:

- Исследование методов обработки естественного языка.
- Исследование онтологического подхода хранения данных.
- Разработка модели пользователя.
- Проектирование архитектуры программы.
- Разработка программной системы.
- Тестирование разработанной системы.

2.2 Анализ существующих решений

Проанализировав различных интеллектуальных помощников, таких как Алиса, Siri, Google Ассистент и другие, можно сделать вывод, что они очень хорошо обрабатывают естественный язык и также имеют отличное поддержание контекста с собеседником, как и выполнение своих функций и возможностей, но они не очень сильно используют информацию пользователя о своих предыдущих запросах и операциях.

Для достижения цели было прежено разработать систему, которая в отличие от предложенных вариантов обладает онтологическим подходом и моделью пользователя, развиваемой заведующим кафедрой и его студентами.

2.3 Модель пользователя

При ведении диалога с пользователем создается его модель в предметной области, состоящая из трех уровней [3]:

- Уровень знаний пользователя.

Знания — это то, что пользователя знает о предметной области. Этот уровень содержит нужную информацию о кафедре и ее людях.

- Уровень намерений пользователя.

Намерения — это то, с какой целью пользователь общается с чат-ботом. Этот уровень содержит сценарии намерений пользователя.

- Уровень управления диалогом.

Управление диалогом — это процесс ведения диалога с пользователем и предоставление ему ответа.

3 Заключение

В данной работе был сделан анализ предметной области и представлены некоторые из аналогов существующих решений. Была разработана программная система, реализующая чат-бота, которая ведет диалог с пользователем на основе его модели, содержащей знания пользователя о кафедре.

Разработанную программную систему планируется внедрить на кафедру общей информатики для автоматизации бизнес-процессов кафедры, связанных с запросом, получением и предоставлением кафедральных документов студентам и научным руководителям.

В рамках работы были получены следующие результаты:

1. Сделан обзор предметной области.
2. Произведен анализ существующих аналогов.
3. Разработана модель пользователя.
4. Разработана архитектура системы.
5. Разработана программная система.
6. Произведено тестирование системы.

Проблемы, с которыми пришлось столкнуться в процессе прохождения практики, состояли в попытках оптимизировать работу с естественным языком и с определением намерений пользователя, а также в организации клиент-серверного соединения.

Литература

[1] Алиса – голосовой помощник от компании Яндекс [Электронный ресурс] – URL: https://yandex.ru/alice (дата обращения: 22.05.2023).

[2] Siri – Apple (RU) [Электронный ресурс] – URL: https://www.apple.com/ru/siri/ (дата обращения: 22.05.2023).

[3] Деревянко Д.В., Пальчунов Д.Е. Формальные методы разработки вопросно-ответной системы на естественном языке // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2014. Т. 12. No 3. С. 34-47.

[4] Palchunov D.,Yakhyaeva G. Application of boolean-valued models and FCA for the development of ontological models // Proceedings of the 2nd International Workshop on Formal Concept Analysis for Knowledge Discovery (FCA4KD), 2017. CEUR Workshop Proceedings. Vol. 1921, p. 77-87.

[5] Palchunov D. Application of FCA for Domain Model Theory Investigation. In: Kovalev S.M., Kuznetsov S.O., Panov A.I. (eds) Artificial Intelligence. RCAI 2021. Lecture Notes in Computer Science, vol 12948. Springer, Cham, 2021, pp. 119-134.

Проверка корректности экспертных оценок, заданных интервалами

Дымонт Надежда Андреевна

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия

n.dymont@g.nsu.ru

Нечеткие вычисления являются одним из важных направлений в современной информатике и искусственном интеллекте. Они представляют собой методологию, которая позволяет моделировать и решать проблемы, связанные с неопределенностью и нечеткостью [1, 2]. Актуальность нечетких вычислений обусловлена их способностью учитывать неопределенность и нечеткость в реальных задачах и проблемах. Они находят применение в различных областях, где требуется анализировать и обрабатывать нечеткую информацию, принимать решения на основе нечетких данных и создавать интеллектуальные системы [3, 4].

Зачастую оценочные знаний о предметной области формулируются не только как объективная (статистическая) вероятность, но и в виде субъективной вероятности [5]. Кроме того, эксперт может не обладать таким полным знанием, более того множество может оказаться бесконечным, а оценки от разных экспертов могут разниться. Тем не менее, существует конечное множество событий предметной области, о вероятностных значениях которых эксперт может дать оценку [6].

Таким образом, на входе мы имеем конечное множество предложений и их интервальные оценки. Необходим инструмент для проверки согласованности экспертных оценок. События представляются в виде формул логики предикатов. В процессе алгоритма проверки используются таблицы истинности формул, на основе которых формируется система интервальных линейных уравнений. Затем полученная система подвергается анализу на предмет ее совместимости. Для этого используются алгоритмы интервального анализа [7]. Экспертные оценки считаются согласованными только в случае, если система интервальных уравнений также оказывается совместной.

Литература

[1] Yakhyaeva G., Logic of fuzzifications. \\ Proceedings of the 4th Indian International Conference on Artificial Intelligence, IICAI 2009, 2009, pp. 222–239

[2] Пальчунов Д.Е., Яхъяева Г.Э. Нечеткие логики и теория нечетких моделей \\ Алгебра и логика, 54, № 1, 2015, с. 109-118.

[3] Yakhyaeva G.E. Application of Boolean Valued and Fuzzy Model Theory for Knowledge Base Development. \\ SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings, 2019, pp. 868–871

[4] Palchunov D., Yakgyaeva G. Representation of Knowledge Using Different Structures of Concepts. \\ CEUR Workshop Proceedings, 2020, vol. 2729, pp. 69–74.

[5] Yakhyaeva G., Skokova V. Subjective Expert Evaluations in the Model-Theoretic Representation of Object Domain Knowledge. \\ Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2021, 12948 LNAI, pp. 152–165

[6] Малаева Е.Д., Яхъяева Г.Э. Программная система визуализации и проверки согласованности оценочных знаний экспертов. \\ Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2023. Т. 21. № 1. с. 32-45.

[7] Шарый С.П. \\ Конечномерный интервальный анализ. Новосибирск: XYZ, 2021, 654 с.

Возможности разработки и применения гибридных подходов для создания децентрализованных метавселенных доступных в реальном времени

Дьяченко М.А.

Новосибирский государственный университет, ул. Пирогова, 1, г. Новосибирск, 630090, Россия

m.dyachenko2@g.nsu.ru

Ограничение активного развития метавселенных - виртуальных миров, направленых на взаимодействие через них пользователей друг с другом, в современном мире связано со многими проблемами - в том числе с централизацией подобных проектов. Связано это с тем, что при создании децентрализованных виртуальных миров с использованием современных блокчейн-технологий возникает проблема взаимодействия пользователей друг с другом в реальном времени, вызвано это с тем, что подтверждение актуального состояния цепочки блоков всеми участниками блокчейн сети занимает время.

Цель работы: изучение возможностей в разработке и применении гибридных систем, которые объединяли бы быстродействие централизованных систем и децентрализованность и надежность блокчейна. Потенциальным решением является использование блокчейна для хранения состояния мира, его правил и параллельное использование сервиса, обеспечивающего доставку информации об изменениях мира до пользователей в реальном времени. Для выполнения работы сформулированы следующие задачи: анализ различных подходов к проектированию таких систем; разработка и тестирование прототипа системы.

Было принято решение использовать централизованный сервер для обмена информацией в реальном времени и подтверждать ее с помощью блокчейна. Для выполнения работы была использована open-source игра, которая поддерживает мультиплеер с использованием центрального сервера. Досутп к блокчейну осуществлялся с помощью API для Java. Игра, написанная на языке Си, была интегрирована с приложением на Java с помощью механизма Java Native Interface, а обмен информацией о состоянии игрового мира осуществляется с использованием очередей сообщений. Таким образом, информация об изменениях поступает пользователям от сервера в реальном времени, а позже подтверждается с помощью блокчейна.

Однако все ещё существенна проблема надежности информации, получаемой параллельно блокчейн сети. Так полученная информация от ненадежного сервера или игрока не может быть отброшена или подтверждена с помощью блокчейна сразу же, после получения. Поэтому все еще необходимо изучение существующих исследований и концепций создания подобных систем. Стоит также отметить, что сервисы, работающие параллельно блокчейну не обязательно должны быть централизованными, главное условие - скорость доставки информации об изменениях в виртуальном мире. Таким образом появляется возможность использования p2p-сетей или сегментирования виртуального пространства на области, которые обслуживаются различными централизованными сервисами. Это позволяет не отказываться от децентрализации и независимости виртуального мира.

Благодарности

Работа выполнена при поддержке АО "СберТех" и ПАО "СБЕРБАНК" на базе студенческой учебно-научной лаборатории факультета информационных технологий Новосибирского государственного университета.

Комплексная интерпретируемость искусственного интеллекта

Зыкова А.А.[1]

[1]*Новосибирский Государственный Университет, ул. Пирогова, д. 2, г. Новосибирск, 630090, Россия.*

azalekss@mail.ru

Аннотация. Искусственный интеллект (ИИ) лежит в основе многих областей деятельности, которые охватили новые информационные технологии. Сложность систем на базе ИИ в последнее время возросла до такой степени, что для них почти не требуется вмешательства человека. Когда же решения, принимаемые с помощью таких систем, в конечном итоге влияют на жизнь людей, возникает потребность в понимании того, как такие решения принимаются с помощью методов ИИ. Таким образом, в настоящее время одна из самых острых проблем применения искусственного интеллекта – это проблема его объяснимости, наличия явного и прозрачного обоснования результатов. В статье рассмотрены и структурированы понятия, связанные с прозрачностью модели, интерпретируемостью и объяснимостью алгоритмов ИИ, дано комплексное определение интерпретируемости.

Ключевые слова: искусственный интеллект, методы искусственого интеллекта, объяснимость, прозрачность, интерпретирумость, оценка объяснимости, черный ящик, метод интерпретации.

1. Введение

Искусственный интеллект лежит в основе многих областей деятельности, которые охватили новые информационные технологии. На сегодняшний день существует четкое согласие относительно первостепенной важности, которую придают интеллектуальным машинам, наделенным способностями к обучению, рассуждениям и адаптации. Сложность систем на базе искусственного интеллекта в последнее время возросла до такой степени, что для них почти не требуется вмешательства человека [1,2].

Когда же решения, принимаемые с помощью таких систем, в конечном итоге влияют на жизнь людей (в медицине, системе уголовного правосудия, на финансовых рынках), возникает проблема неспособности людей понять некоторые методы искусственного интеллекта [3], модели машинного обучения, вследствие чего возникают недоверие к искусственному интеллекту в целом и полученным с его помощью результатам, сомнение в справедливости принятого решения. Поэтому существует потребность понимания того, как такие решения принимаются с помощью методов ИИ.

Таким образом, в настоящее время одна из самых острых проблем применения искусственного интеллекта – это *проблема его объяснимости*, наличия явного и прозрачного обоснования результатов [4]. Считается, что объяснимость поможет людям с большим доверием относиться к искусственному интеллекту.

Но и на этапе постановки задачи возникает важная проблема – проблема разносторонности самого понятия объяснимости, а также понятий интерпретируемости, прозрачности, достоверности и других. Объяснимый искусственный интеллект сейчас – очень популярная область для изучения, многие университеты и компании пытаются предложить новый подход для объяснимости результатов [5]. Понятия объяснимости используются всегда, когда речь заходит о недоверии к результатам искусственного интеллекта, но существует проблема неопределенности этого понятия. Можно сделать вывод, что авторы статей и различные исследователи негласно понимают, что значит «объяснимость», «интерпретируемость», «прозрачность», но либо при этом не определяют этих понятий вообще,

либо определяют часть из них и сталкиваются с противоречиями с уже существующими определениями.

Например, «интерпретируемость — это степень, в которой человек может понять причину решения» [6]. Или же «интерпретируемость — это степень, в которой человек может последовательно предсказать результат модели» [7]. Интерпретируемость называют объяснимостью и наоборот, из-за чего понятия сливаются в одно и перестают иметь смысл как описывающие разные характеристики понятия.

Цель статьи – рассмотреть и структурировать понятия, связанные с прозрачностью модели, интерпретируемостью и объяснимостью алгоритмов ИИ.

2. Предпосылки к определению понятия интерпретируемости

Итак, когда речь заходит про интерпретируемость, выражаются такие желания, как доверие к результату алгоритма искусственного интеллекта, генерирование гипотез, переносимость обученной модели в новые непредсказуемые для нее ситуации, справедливость и непредвзятость, а также многое другое. Рассмотрим интерпретируемость с разных сторон. Так как обычно она включает в себя различные факторы, стоит рассмотреть понятие «интерпретируемости» более комплексно.

Начнем с того, почему вообще возникают разговоры об интерпретируемости и почему людям становится так важно получать объяснения и убеждаться в интерпретируемости моделей.

2.1.Очевидность корректности результатов

Как уже было указано выше, проблема недоверия – самый популярный фактор, способствующий продвижению сферы искусственного интеллекта в сторону интерпретируемости [8], но и проблема эта не возникает на пустом месте. От простых до сложных объёмных случаев люди сталкиваются с ошибочными результатами.

Например, когда модель присваивает более высокий вес входным данным, которые на самом деле не должны иметь такой вес. Примером такой ситуации является анализ модели, которая классифицирует волков и хаски. Было показано, что она основывала свои выводы на наличии снега на фоне [9].

Также и злоумышленники могут добавлять маленькие искажения в изображения, которые незаметны для человека, но сбивает с толку алгоритм, ранее доказавший свою эффективность на не измененных изображениях [10].

Идея того, что даже самые простые задачи на самом деле могут выдавать некорректные или предвзятые, скомпрометированные результаты, а также во многом зависят от способа и корректности обучения, наталкивает на мысли, что в более сложных ситуациях с более сложными моделями, неподвластными людям без специализации в данной сфере, тем более нельзя доверять результатам искусственного интеллекта. Здесь также играет роль свойство переносимости моделей.

Обычно мы выбираем обучающие и тестовые данные, случайным образом разбивая примеры из одного и того же распределения. Затем мы оцениваем ошибку по разнице ее эффективности на обучающих и тестовых данных. Однако люди демонстрируют гораздо более богатые способности к обучению, проявляя приобретенные навыки в незнакомых ситуации, и хочется реализовать алгоритмы машинного обучения в таких же ситуациях - когда требуются возможности приспосабливаться к ситуации. Точно также развертываются модели в условиях, когда может измениться окружающая среда, сделав недействительными их прогнозы на будущее. Примером является модель, обученная предсказывать вероятность смерти от пневмонии. Она предсказывает меньший риск для пациентов, у которых есть также и астма [11]. То есть астма по мнению модели являлась признаком более низкого риска смерти, но это предсказание связано только лишь с более активным лечением этих пациентов.

Хуже того, модели контролируемого обучения уже регулярно подвергаются манипуляциям. Например, в модели создания кредитных рейтингов более высокие баллы должны означать более высокую вероятность того, что человек способен погасить кредит. Согласно их собственному техническому отчету, FICO обучает кредитные модели с использованием логистической регрессии [12]. Поэтому соискатели кредита могут манипулировать некоторыми из этих переменных функций модели, и это сейчас не является редкостью. Например, максимальную сумму можно улучшить, запрашивая периодически увеличение суммы, при этом структура расходов сохраняется такой же. Сами FICO и Experian также признают, что кредитными рейтингами можно манипулировать и даже предлагают руководства по улучшению кредитной истории. Эти стратегии тем временем принципиально не меняют способность

человека выплатить долг. Тот факт, что люди активно и успешно используют эту систему, может свести на нет ее предсказательную силу.

2.2. Очевидность справедливости и непредвзятости

Также важным фактором является справедливость и непредвзятость алгоритмов. В настоящее время политики, журналисты и исследователи выражают обеспокоенность тем, что результаты моделей необходимо интерпретировать, чтобы оценить, насколько решения, автоматически генерируемые алгоритмами, соответствуют этическим нормам. [13].

Действительно, алгоритмы все больше участвуют в нашей жизни, влияя на новости, которые нам приходится смотреть, наши финансы и возможности карьерного роста. Мы позволяем компьютерным программам утверждать кредиты, просматривать новости и отбирать кандидатов на работу. Суды даже используют такие алгоритмы для прогнозирования рецидива преступного поведения человека [14]. Вполне вероятно, что эта тенденция будет только усиливаться, поскольку прорывы в области искусственного интеллекта быстро расширяют возможности программного обеспечения.

Прогнозы рецидивизма уже используются для определения, кого освободить, а кого задержать, что вызывает этические опасения. Мы не можем быть уверены, что предсказания не зависят от расы или ориентации. Обычные метрики оценки дают мало гарантий того, что модель и ее действия, согласно теории принятия решений, ведут себя приемлемо. Новые правила в Европе предполагают, что лица, которых затронули алгоритмическими решениями, имеют право на объяснение. Но вопрос, какую именно форму может принять такое объяснение или как такое объяснение может быть обосновано, остается открытым. Более того, те же правила предполагают, что алгоритмические решения должны быть оспоримы.

2.3. Построение прогнозов

Исследователи часто используют модели в надежде вывести новые свойства или выдвинуть гипотезы о мире природы. Например, простая модель может выявить тесную связь между применением талидомида и врожденными дефектами или курением и раком легких [15].

Ассоциации, изученные алгоритмами обучения с учителем, не обязательно отражают причинно-следственные связи, однако можно надеяться, что, интерпретируя модели контролируемого обучения, мы сможем сгенерировать гипотезы, которые ученые затем смогут проверить экспериментально.

3. Понятие интерпретируемости

Исходя из описанных предпосылок к определению интерпретируемости, можно сделать вывод, что, во-первых, людям для формирования доверия необходимо понимать, *как* работает модель, и во-вторых, хочется получить *объяснение*, почему она пришла к тому или иному результату.

В источниках литературы также можно встретить такое понятие как прозрачность [16], которое как раз и описывает, как именно работает модель, является некоторой противоположностью «чёрному ящику», о котором принято говорить в рамках машинного обучения, нейронных сетей. При этом речь может идти о прозрачности, когда мы хотим понимать работу модели в целом, её отдельных компонентов и алгоритма обучения. Как и говорилось, даже на этапе определения возникают некоторые противоречия.

Итак, опишем интерпретируемость комплексно, исходя из того, что обычно понимают под ней в различных источниках литературы и исследованиях.

3.1. Симулируемость

Симулируемость - возможность анализа модели человеком, возможность симулировать работу алгоритма с нуля. Наиболее важным критерием для симулируемости является сложность модели. Простые и при этом со слишком большим количеством правил системы не соответствуют этой характеристике в отличие от, например, одиночной нейронной сети персептрона.

Для некоторых же моделей, размер модели может быть больше, чем время исполнения. Например, дерево решений. Это говорит о том, что возможность моделирования может иметь два варианта, один из которых основан на общем размере модели, а другой — на основе вычислений.

3.2. Декомпонуемость (разложимость)

Разложимость означает возможность объяснить (хотя бы интуитивно) каждую часть модели – почему такие или иные входные данные, параметры и выходные данные. Громоздкие функции, например, не соответствуют данному критерию. Это также согласуется со свойством понятности [17].

3.3. Алгоритмическая прозрачность

Алгоритмическая прозрачность — это способность пользователя понимать процессы, происходящие в модели ИИ, и то, что происходит с входными данными. Линейные модели ИИ считаются прозрачными, поскольку их поверхности ошибок (математические интерпретации механизмов обучения ИИ) понятны и могут быть изучены, что дает пользователю достаточное представление о том, как будет работать модель в каждой ситуации, с которой он может столкнуться.

В случае глубоких архитектур это не так, поскольку поверхности ошибок не прозрачны и не могут быть полностью проанализированы, в результате чего для аппроксимации решения приходится использовать эвристическую оптимизацию(например, стохастический градиентный спуск).

4. Понятие объяснимости (апостериорной интерпретируемости)

Апостериорная интерпретируемость представляет собой особый подход к извлечению информации из изученных моделей. Хотя апостериорные интерпретации часто не объясняют, *как* именно работает модель, они, тем не менее, могут предоставить полезную информацию для практиков и конечных пользователей машинного обучения. Подходы к апостериорным интерпретациям включают *объяснения* на естественном языке, визуальные *объяснения* и *объяснение* на примере, поэтому апостериорную интерпретируемость также можно назвать объяснимостью.

4.1. Текстовые объяснения

Люди часто оправдывают решения устно. Точно так же мы можем научить одну модель генерировать прогнозы, а другую модель - генерировать объяснения [18]. В работе над рекомендательными системами также используют текстовые объяснения решений модели скрытых факторов [19].

4.2. Визуализация

Другим распространенным подходом к созданию апостериорных интерпретаций является визуализация, позволяющая определить, чему научилась модель. Одним из распространенных подходов является визуализация многомерных распределенных представлений с помощью t-SNE.

t-SNE — это двумерная техника визуализации, которая отображает соседние точки данных вне посредственной близости друг от друга. [20].

Большинство методов визуализации сопровождаются методами снижения размерности, чтобы сделать модель более понятной для человека. Методы визуализации могут комбинироваться с другими методами для лучшего понимания и считаются наиболее подходящим способом представления сложных взаимодействий между переменными, участвующими в модели [21].

4.3. Объяснение на примере

Одним из методов объяснения решений модели может быть отчет о том, какие другие примеры модель считает наиболее похожими [22]. После обучения глубокой нейронной сети или модели со скрытыми переменными для задачи классификации можно получить доступ как к

предсказаниям, так и к обученным представлениям. Тогда для любого примера, помимо генерации предсказаний, можно определить k-ближайшего соседа на основе близости в пространстве с помощью активации скрытого слоя. Объяснение на примере похоже на то, как люди могут объяснять свои действия по аналогии.

Таким образом, термин интерпретируемости не стоит рассматривать как единичное определение. Чтобы оно имело смысл в рамках любой работы, нужно фиксировать конкретную часть комплексного определения, ведь достичь комплексной интерпретируемости сложно.

Итак, интерпретируемость модели может быть рассмотрена на всех этапах разработки искусственного интеллекта, как для изначально интерпретируемых моделей ИИ, так и для моделей на основе «черного ящика».

4. Заключение

Таким образом, в статье была изучена проблема интерпретируемости искусственного интеллекта, поставлена проблема определения понятий интерпретируемости, объяснимости, выявлены предпосылки определения данных понятий и характеристики, которые они должны отображать. Затем даны комплексные определения понятий интерпретируемости, объяснимости.

Литература

1. **Palchunov D.E., Tishkovsky D.E., Tishkovskaya S.V., Yakhyaeva G.E.** Combining logical and statistical rule reasoning and verification for medical applications. *International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, 2017, p. 309-313.
1. **Palchunov D.E., Yakhyaeva G.E.** Integration of Fuzzy Model Theory and FCA for Big Data Mining. *International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, 2019, p. 961–966. DOI: 10.1109/SIBIRCON48586.2019.8958216
2. **Kim B., Glassman E., Johnson B., Shah J.** iBCM: Interactive Bayesian case model empowering humans via intuitive interaction. *Computer Science and Artificial Intelligence Lab (CSAIL)*, 2015.
3. **Yakhyaeva G.E, Skokova V.** Subjective Expert Evaluations in the Model-Theoretic Representation of Object Domain Knowledge. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021, vol 12948, p. 152–165. DOI:10.1007/978-3-030-86855-0_11
4. **Арстангалиев Д.Т., Яшин В.И., Дронин В.В., Пшиков Г.А., Муравицкий Н.С., Солдатов А.С.** Объясняемый искусственный интеллект. *Состояние и перспективы развития современной науки по направлению" Информатика и вычислительная техника"*, 2020, с. 33-40.
5. **Miller T.** Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*., 2019, vol. 267, p. 1-38. DOI:10.1016/j.artint.2018.07.007
6. **Kim B., Khanna R., Koyejo O.O.** Examples are not enough, learn to criticize! Criticism for interpretability. *Advances in neural information processing systems*., 2016, vol. 29.
7. **Kim B.** Interactive and interpretable machine learning models for human machine collaboration. *Massachusetts Institute of Technology*, 2015, p. 135-143.
8. **Ribeiro M.T., Singh S., Guestrin C.** "Why should i trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.*, 2016, p. 1135-1144. DOI:10.1145/2939672.2939778
9. **Goodfellow I., Papernot N., Huang S., Duan Y., Abbee P.** Attacking machine learning with adversarial examples. *OpenAI Blog*., 2017, vol. 24.
10. **Caruana R., Lou Y., Gehrke J., Koch P., Sturm M.** Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*., 2015, p. 1721-1730. DOI:10.1145/2783258.2788613
11. **Poon M.** Scorecards as devices for consumer credit: The case of Fair, Isaac & Company Incorporated. *The sociological review*, 2007, vol. 55, p. 284-306.
12. **Goodman B., Flaxman S.** European Union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 2017, vol. 38, p. 50-57. DOI:10.1609/aimag.v38i3.2741
13. **Chouldechova A.** Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 2017, vol. 5, p. 153-163, DOI:10.1089/big.2016.0047
14. **Wang H.X., Fratiglioni L., Frisoni G.B.** Smoking and the occurence of Alzheimer's disease: Cross-sectional and longitudinal data in a population-based study. *American journal of epidemiology.*, 1999, vol. 149, p. 640-644. DOI:10.1093/oxfordjournals.aje.a009864.

15. **Roovers R.** Transparency and Responsibility in Artificial Intelligence. *A call for explainable AI*, 2019.
16. **Lou Y., Caruana R., Gehrke J.** Intelligible models for classification and regression. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*., 2012, p. 150-158. DOI:10.1145/2339530.2339556
17. **Krening S., Harrison B., Feigh K.M.** Learning from explanations using sentiment and advice in RL. *IEEE Transactions on Cognitive and Developmental Systems*, 2016, vol. 9, pC. 44-55. DOI:10.1109/TCDS.2016.2628365
18. **McAuley J., Leskovec J.** Hidden factors and hidden topics: understanding rating dimensions with review text. *Proceedings of the 7th ACM conference on Recommender systems*, 2013, p. 165-172. DOI:10.1109/TCDS.2016.2628365
19. **Van der Maaten L., Hinton G.** Visualizing data using t-SNE. *Journal of machine learning research*, 2008, vol. 9
20. **Mordvintsev A., Olah C., Tyka M.** Inceptionism: Going deeper into neural networks. *Google Research Blog*, 2015.
21. **Mahendran A., Vedaldi A.** Understanding deep image representations by inverting them. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, p 5188-5196.
22. **Caruana R., Kangarloo H., Dionisio J.D.** Case-based explanation of non-case-based learning methods. *Proceedings of the AMIA Symposium*, 1999, p. 212.

Разработка сервиса для автоматизированной рассылки электронных писем

Ковалевская Ольга Артемовна

НГУ, ул. Пирогова, д.1, г. Новосибирск, 630090, Россия

o.kovalevskaya@g.nsu.ru

Аннотация. *Разработка сервиса для автоматизированной рассылки электронных писем по адресам из базы данных. Реализация анализа корректности содержания отправленных писем и полученных ответов. В работе представлен обзор существующих программ для автоматизированной рассылки, а также предполагаемая архитектура подобного сервиса.*

Ключевые слова: электронная почта, e-mail, рассылка электронных писем, автоматизация рассылки

1 Введение

В современном мире электронная почта является самым популярным инструментом для передачи данных в Интернете. Ежедневное использование электронной почты подтверждает 69% опрошенных (больше, чем использование социальных сетей среди пользователей всех возрастов), и с каждым годом электронная почта становится все более популярной.
ВУЗы активно применяют электронную почту для обмена информацией между преподавателями, студентами и администрацией и сталкиваются с необходимостью рассылать огромное число электронных писем каждый день. В процессе рассылки электронных писем также возникают определенные сложности, связанные с контролем доставки и отслеживанием ответов. Это приводит к большим затратам человеческих и временных ресурсов на работу с электронной почтой. Для решения данной проблемы существуют сервисы для автоматизации рассылки.

Основные преимущества использования сервиса для автоматизации включают:
1. Значительное сокращение затрат времени и усилий на рассылку.
2. Возможность массовой отправки сообщений большому числу адресатов, что особенно актуально для ВУЗов, где требуется регулярная информационная поддержка студентов и преподавателей.
3. Встроенные инструменты для контроля доставки писем и отслеживания ответов, что позволит эффективно управлять коммуникацией и оперативно реагировать на запросы и обращения.

Таким образом, разработка сервиса по автоматизированной рассылке электронных писем для ВУЗа является актуальной и перспективной задачей, способной значительно улучшить процесс коммуникации в учебном заведении и повысить его эффективность.

1.1 Обзор существующих сервисов для автоматизации рассылки

Существует множество сервисов для автоматизации рассылки электронных писем. Некоторые из них:

MailChimp. Один из самых популярных сервисов для рассылки электронных писем. Предоставляет широкий набор функций: возможность создания и настройки шаблонов писем, автоматическая персонализации писем и отслеживание результатов рассылки.

Brevo. Этот сервис также предоставляет широкий набор функций для автоматизированной рассылки писем. Позволяет создавать и настраивать шаблоны писем, управлять адресными списками, проводить A/B-тестирование и анализировать результаты проведения рассылки.

GetResponse. Сервис, помимо прочего, предлагает функциональность для создания опросов и лендинг-страниц.

MailerLite. Этот сервис предлагает простой интерфейс для создания и отправки электронных писем. Он также предоставляет возможность управления адресными списками и отслеживания результатов.

Это лишь некоторые из множества сервисов, доступных для автоматизированной рассылки электронных писем. Их функционал достаточно схож. Несмотря на множество преимуществ, у этих сервисов также есть недостатки. Некоторые из них включают в себя:

- Ограничения по количеству подписчиков или отправляемых писем в рамках бесплатного плана. Это может быть проблематично для крупных учебных заведений с большим количеством студентов и сотрудников.
- Ограничения по функциональности в бесплатном плане, что может ограничить возможности автоматизации и персонализации рассылки.
- Сложности с настройкой и интеграцией. Некоторые сервисы могут требовать определенных технических навыков для настройки и интеграции с другими системами, такими как CRM или учебная платформа.
- Возможные проблемы с доставкой писем. Иногда письма, отправленные через эти сервисы, попадают в спам, что может снизить эффективность рассылки.
- Ограничения по доступности в определенных странах. Некоторые сервисы могут быть недоступны или иметь ограниченную функциональность в определенных странах, что может создать проблемы для учебных заведений, имеющих международную аудиторию.
- Необходимость предоставлять доступ к персональным данным сторонним лицам.

1.2 Архитектура сервиса для автоматизации рассылки

Архитектура сервиса для автоматизированной рассылки электронных писем может быть различной в зависимости от требований и целей проекта. Однако, обычно такой сервис включает следующие компоненты:

Фронтенд. Интерфейс, через который пользователи могут создавать и управлять рассылками, настраивать шаблоны писем и просматривать статистику. Фронтенд может быть реализован в виде веб-приложения или мобильного приложения.
База данных. Хранит информацию о подписчиках, шаблонах писем, расписаниях отправки и других данных, необходимых для работы сервиса.
Серверная часть. Обрабатывает запросы от фронтенда и выполняет необходимые операции, такие как добавление подписчиков в список, отправка писем, отслеживание статистики и т. д.
Очередь сообщений. Используется для планирования и управления отправкой писем. При добавлении новых писем в очередь они будут отправлены по расписанию или при выполнении определенных условий.
Сервисы доставки писем. Взаимодействует с почтовыми серверами для отправки электронных писем. Может использоваться сторонний сервис доставки писем.
Мониторинг и аналитика. Собирает данные о производительности и использовании сервиса, а также предоставляет статистику о доставке писем, открытии и кликах.

Важно разработать архитектуру, которая будет масштабируемой и гибкой, чтобы обеспечить возможность расширения функциональности и обработки большого количества подписчиков и писем.

2 Заключение

Результатом работы должна стать разработка сервиса для автоматизированной рассылки электронных писем по адресам из базы данных с помощью языка программирования Python. Следующий этап реализации сервиса – анализ полноты отправленных писем и полученных ответов.

Литература

[1] Сервис автоматизации маркетинга для онлайн-бизнеса eSputnik. [Электронный ресурс] URL: https://esputnik.com/

[2] Lapalme, G., Kosseim, L.: Mercure: Towards an automatic e-mail follow-up system. IEEE Computational Intelligence Bulletin 2(1), 14–18 (2003)

[3] Mailchimp [Электронный ресурс]. URL: https://mailchimp.com/marketing-glossary/email-automation/

[4] Brevo [Электронный ресурс]. URL: https://www.brevo.com/products/transactional-email/

[5] GetResponse [Электронный ресурс]. URL: https://www.getresponse.ru/email-marketing/ru-features/email-marketing

[6] MailerLite [Электронный ресурс]. URL: https://www.mailerlite.com/features/

Знания-Онтологии-Теории (ЗОНТ-2023)

Рассуждения на основе прецедентов: обзор методов и перспектив их применения.

Копытков Дмитрий

d.kopytkov@g.nsu.ru

Аннотация. *С использованием прецедентного подхода возможно улучшение качества принятия решений и повышение эффективности их выработки. Прецедентный подход позволяет использовать опыт известных и успешных случаев для принятия решений в конкретных ситуациях. В данной статье, мы сосредоточим внимание на исследовании прецедентов как метода разработки программного обеспечения, а также на их влиянии на качество разработки и результаты.*

Ключевые слова: Прецедентный подход, рассуждения на основе прецедентов (CBR).

1 Введение

В современном мире все большее количество задач требует принятия сложных решений на основе большого объема данных. В таких условиях стандартные методы анализа и обработки данных не всегда могут обеспечить требуемую точность и эффективность решений. В этой связи в последнее время все большее внимание уделяется методам рассуждений на основе прецедентов. Рассуждения на основе прецедентов – это метод, основанный на анализе и классификации предыдущих случаев решения аналогичных задач. В данной статье рассмотрены основные принципы работы с прецедентами, а также методы их классификации и анализа. Рассмотрены перспективы применения рассуждений на основе прецедентов в медицине, праве и технологиях.

2 Обзор литературы

Прецедентный подход был описан в ряде работ, в том числе в работах О. Голдшмидта и Дж. Маккарти, а также в работах А. Фейера и Р. Карр.

Одним из примеров использования прецедентного подхода в программной инженерии является методология разработки Case-Based Reasoning (CBR), разработанная М. Коллинзом в 1986 году. Методология CBR является эффективным инструментом для повышения качества программного обеспечения и ускорения процесса его разработки.

3 Основная часть

Case Based Reasoning (CBR) - это метод искусственного интеллекта, который использует предыдущий опыт (кейсы) для решения новых проблем. Математически, CBR можно описать как процесс нахождения наилучшего соответствия между задачей и кейсом. Формально, процесс можно представить следующим образом:

1.Задача T задается как набор свойств или признаков T = {t1, t2, ..., tn}.

2.Существует множество кейсов C = {C1, C2, ..., Cm}, где каждый кейс Ci состоит из набора свойств или признаков Ci = {ci1, ci2, ..., cik}.

3.Расстояние между задачей T и кейсом Ci определяется функцией расстояния d(T, Ci), которая может быть определена как евклидово расстояние, манхэттенское расстояние или любая другая метрика.

4.Наилучшее соответствие между задачей T и кейсом Ci определяется путем нахождения минимального расстояния между T и Ci, т.е. min d(T, Ci).

5.Задача T решается путем использования решения, которое было применено в кейсе Ci.

Прецедентный подход также может быть использован для улучшения качества программного обеспечения. Программы, созданные с использованием прецедентов, могут быть более точными, надежными и эффективными. Прецеденты могут использоваться для выявления ошибок в программном обеспечении и улучшения его функциональности.

Несмотря на преимущества, есть и недостатки при использовании прецедентного подхода. В частности, описание прецедентов может требовать большого количества времени и усилий. Кроме того, важно иметь качественную базу данных прецедентов, что может быть проблемой.

4 Выводы и дальнейшие перспективы исследования

Системы CBR решают новые проблемы, извлекая соответствующие предыдущие случаи и адаптируя их к новым ситуациям. CBR использовался в различных областях, таких как юридические рассуждения, рассуждения на основе памяти и комбинации CBR с другими методами рассуждений.

В последние годы не было сообщений о переходе от CBR к другой технологии. Вместо этого, CBR используется в сочетании с другими технологиями для улучшения решения проблем и обучения.[6] Кроме того, в статье от 2015 года показано использование CBR в кластеризации документов и поиске корреляций между медицинскими данными с помощью CBR с технологией БД в качестве приложения. [7].

Литература

[1] Goldschmidt, O. and McCarthy, J. (1967). The Primitive Elements of Expert Systems. In Proceedings of the 1967 Spring Joint Computer Conference, pages 181– 189.
[2] Feigenbaum, E.A. and McCorduck, P. (1983). The Fifth Generation: Artificial Intelligence and Japan’s Computer Challenge to the World. AddisonWesley, Reading, MA.
[3] Collins, M. (1986). Case-Based Reasoning. Morgan Kaufmann, Los Altos, CA.
[4] Carr, R. and Feigenbaum, E.A. (1987). The Handbook of Artificial Intelligence, volume 3. Elsevier Science Publishers B.V. (North Holland), New York, NY.
[5] Aamodt, A. and Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications, 7(1):39–59.
[6] https://ieeexplore.ieee.org/document/7384138
[7] https://link.springer.com/chapter/10.1007/978-81-322-2250-7_37

Обучение модели для прогнозирования продовольственных запасов в Российской Федерации

Курдюков Данил Александрович

Новосибирский государственный университет, ул. Пирогова 1, г. Новосибирск, 630090, Россия.
Novosibirsk State University, 1 Pirogova str., Novosibirsk, Russia.

d.kurdyukov@g.nsu.ru

В последние годы, для задач прогнозирования результатов всё чаще и чаще прибегают к помощи искусственного интеллекта [1], т.к. он способен обнаруживать нетривиальные зависимости в передаваемых данных. Однако, для качественного обучения искусственного интеллекта, требуются большие обучающие выборки на тысячи записей, которые иногда сложно или невозможно собрать.

Целью работы является обучение модели для прогнозирования продовольственных запасов в Российской Федерации. Основная задача заключалась в сравнении популярных алгоритмов, используемых для регрессии, с алгоритмами метаобучения. В качестве обучающей выборки была использована открытая база ООН по продовольственным товарам, дополненная из Российских источников. Полученная обучающая выборка включает в себя порядка 20 видов продуктов с информацией по годовому производству, потреблению, импорту, экспорту с 1960 по 2020. Для сравнения использовались следующие алгоритмы, LinearRegression, KNeighborsRegressor, MultiTaskLassoCV, RidgeCV из библиотеки csikit-learn, нейронная сеть с 2 скрытыми слоями, из алгоритмов метаобучения, были реализованы алгоритмы MAML и Reptile, как наиболее популярные и подходящие для решения широкого спектра задач [2] . Сравнение алгоритмов проводилось на 10% тестовых данных и на данных за последние 5 лет.

Наилучшие результаты на 10% тестовых данных показали такие алгоритмы как нейронная сеть с 2 скрытыми слоями и алгоритмы метаобучения, их ошибки составили порядка одного процента, ошибки остальных алгоритмов составили порядка пяти процентов. С данными за последние 5 лет, процентная ошибка на них составила порядка одиннадцати процентов для мета-алгоритмов и двенадцати для остальных. При более детальном изучении данного феномена стало понятно, что большой негативный вклад привносят предсказания импорта и экспорта т.к. в некоторых категориях произошёл большой прирост за последние несколько лет. Например экспорт масла и животных жиров, вырос более чем в 4 раза за последние 10 лет.

Алгоритмы метаобучения показали лучшие результаты по сравнению с популярными алгоритмами. Недостатком является то, что первоначальное “общее” обучение модели потребовало больших временных затрат, что нивелируется сохранением весов.

Благодарности

Работа выполнена при поддержке АО "СберТех" и ПАО "СБЕРБАНК" на базе студенческой учебно-научной лаборатории факультета информационных технологий Новосибирского государственного университета

Литература

[1] Иннополис, Межотраслевой центр трансфера технологий. Применение искусственного интеллекта в приоритетных отраслях экономики
[2] Chelsea Finn, Pieter Abbeel, Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

Анализ согласованности логических аспектов экспертных оценок

Малаева Елена Дмитриевна

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия

e.malaeva@g.nsu.ru

В современном мире мы имеем доступ к огромному объему информации и знаний. За частую, эти знания являются неполными и/или неточными [1, 2]. Не вызывают сомнений все преимущества данного факта, но с этим также возникают трудности в принятии корректных решений: на одну и ту же ситуацию может быть высказано несколько различных точек зрения, которые, хотя и допустимы, могут быть несогласованными друг с другом [3]. Это неизбежно влечет сомнения относительно экспертов, которым и принадлежат данные взгляды.

Для разрешения этой проблемы следует воспользоваться проверкой согласованности экспертных оценок [4, 5]. Она позволяет определить, насколько надежны оценки экспертов, а также степень их консистентности. Разработка системы, которая автоматизирует процесс проверки логической согласованности оценок экспертов, повышает точность расчетов, сокращает затраты времени на них и значительно снижает процент ошибок в ходе работы.

Процесс выглядит так: знания экспертов преобразуются в формализованные выражения логики предикатов и вероятностей, которые отражают степень убежденности эксперта в их корректности. Затем на основе этого набора формул строится система линейных уравнений. Анализируя ее совместность, производится проверка логической согласованности высказываний. С использованием множества СДНФ (совершенных дизъюнктивных нормальных форм) из формул создается графическое представление, которое также позволяет визуализировать места конфликтов значений. После этого предоставляются возможности для разрешения этих конфликтов путем модификации вероятностных значений определенных формул. Корректность процесса проверяется с использованием решения системы линейных уравнений. С ростом темпов автоматизации и объема данных, этот инструмент становится все более необходимым и актуальным в различных сферах деятельности.

Литература

[1] Yakhyaeva G., Logic of fuzzifications. Proceedings of the 4th Indian International Conference on Artificial Intelligence, IICAI 2009, 2009, стр. 222–239

[2] Пальчунов Д.Е., Яхъяева Г.Э. Нечеткие логики и теория нечетких моделей \\ Алгебра и логика, 54, № 1, 2015, с. 109-118.

[3] Palchunov D., Yakgyaeva G. Representation of Knowledge Using Different Structures of Concepts. CEUR Workshop Proceedings, 2020, vol. 2729, pp. 69–74.

[4] Yakhyaeva G., Skokova V. Subjective Expert Evaluations in the Model-Theoretic Representation of Object Domain Knowledge. Lecture Notes in Computer Science. 2021. Т. 12948 LNAI. p. 152-165.

[5] Малаева Е.Д., Яхъяева Г.Э. Программная система визуализации и проверки согласованности оценочных знаний экспертов // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2023. Т. 21. № 1. С. 32-45.

Разработка системы динамической инструментации системных вызовов на основе gVisor

А. М. Мацько

Новосибирский государственный университет, ул. Пирогова, д. 1, г. Новосибирск, 630090, Россия.

a.matsko@g.nsu.ru

Современные требования к высокопроизводительным вычислениям предполагают создание гибких и надежных сред для запуска приложений. В данном контексте, разработка системы динамической инструментации системных вызовов на основе gVisor представляет собой значимый шаг к решению данной проблемы.

Предлагаемый функционал системы включает в себя: возможность подмены аргументов и возвращаемых значений системных вызовов, а также интеграцию пользовательских скриптов с возможностью использования набора предопределенных функций, имеющих доступ к внутренним структурам gVisor; что предоставит разработчикам уникальные инструменты для более тонкой настройки и анализа работы приложений.

Целью работы является разработка системы динамической инструментации системных вызовов на базе gVisor. Для достижения цели были выделены следующие задачи: анализ существующих средств инструментации и используемых ими подходов, разработку прототипа, предоставляющего функционал, описанный выше, и его тестирование в реальных сценариях.

На текущий момент разработан прототип системы, который предоставляет средства для настройки и анализа работы приложений. В частности, предопределённые функции позволяют пользователю с использованием скриптов получить информацию о содержимом буферов, дают возможность изменить это содержимое. Также пользователь может вмешаться в исполнение системного вызова, а именно: изменить возвращаемое значение или вовсе отменить его исполнение.

Возможность динамической инструментации системных вызовов представляет собой значимый шаг в создании более гибких и безопасных сред для запуска приложений в вычислительных окружениях.

Благодарности

Работа выполнена при поддержке *АО "СберТех"* и *ПАО "СБЕРБАНК"* на базе студенческой учебно-научной лаборатории *факультета информационных технологий Новосибирского государственного университета*

Знания-Онтологии-Теории (ЗОНТ-2023)

Извлечение эмоций из текстов естественного языка при помощи LogicText

Миронов В.С.

Новосибирский Государственный Университет 630090, Новосибирская область, г. Новосибирск, ул. Пирогова, 2

v.mironov1@g.nsu.ru

Аннотация. *В данной статье рассматривается вопрос автоматического распознавания эмоций, заложенных в разнообразных текстах. Речь пойдет о механизмах связанных с формальным представлением эмоций, изложенных на естественном языке. Будут рассмотрены методы извлечения и формализации эмоций, изложенных на естественном языке.*

Ключевые слова: Logic Text, извлечение эмоций, формализация эмоций.

1 Введение

Наиболее распространенной формой представления знаний являются тексты на естественном языке. Однако интенсивный рост текстовых массивов за последние несколько десятилетий является причиной трудностей анализа этих данных, когда возникает потребность. Данная проблема дополняется сложностью обработки эмоций, заложенных в тексте. Современные системы, работающие с текстовой информацией не рассчитаны на обработку эмоциональной составляющей текстовой информации, поскольку оперируют лишь словами текста, а не содержанием.

Таким образом возникает актуальность системы, позволяющей извлекать эмоции из текстов естественного языка[1]. В результате извлечения эмоции приобретают явный вид и становятся пригодными для автоматизированной обработки что в свою очередь даст возможность автоматизировать переписку с клиентами, выстраивая диалог согласно эмоциональной составляющей полученного текста.

В данной работе представлен метод извлечения и распознавания эмоций, извлеченных из текстов на русском языке, а также способы их представления и формализации.

2 Теоретические проблемы

2.1 Постановка задачи

В данном исследовании необходимо было произвести усовершенствование системы LogicText. Для существующей системы был создан внешний модуль способный распознавать эмоции, содержащиеся в предложениях на русском языке, а также модуль, интегрирующий знания об эмоциях в атомарную диаграмму.

2.2 Распознавание эмоций

Для распознавания эмоций была написана нейронная сеть[2,3] на языке Python с использованием библиотеки Keras, а также был написан модуль для связи ПО LogicText и нейронной сети. Для оценки качества использовалась функция evaluate которая показала точность модели на тестовых данных 81.69% при определении эмоциональной окраски предложений.

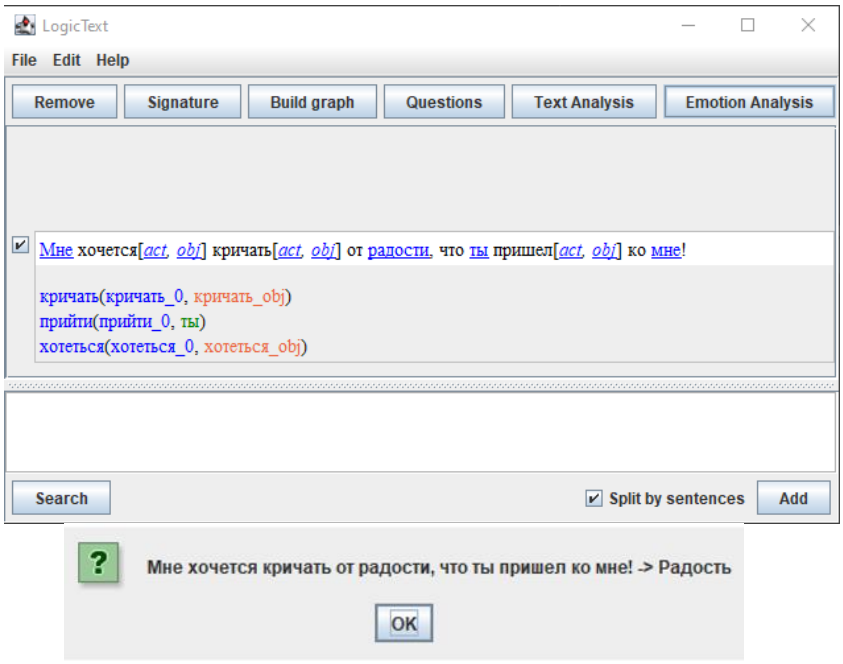


Рис.1. Интерфейс программы LogicText.

2.3 Интеграция эмоций в атомарную диаграмму.

Также в ПО LogicText был добавлен функционал, дополняющий атомарные диаграммы. Теперь на атомарных диаграммах[4], которые предоставляет LogicText можно увидеть причины возникновения тех или иных эмоций. На рисунке 2 и 3 представлен пример такой атомарной диаграммы.

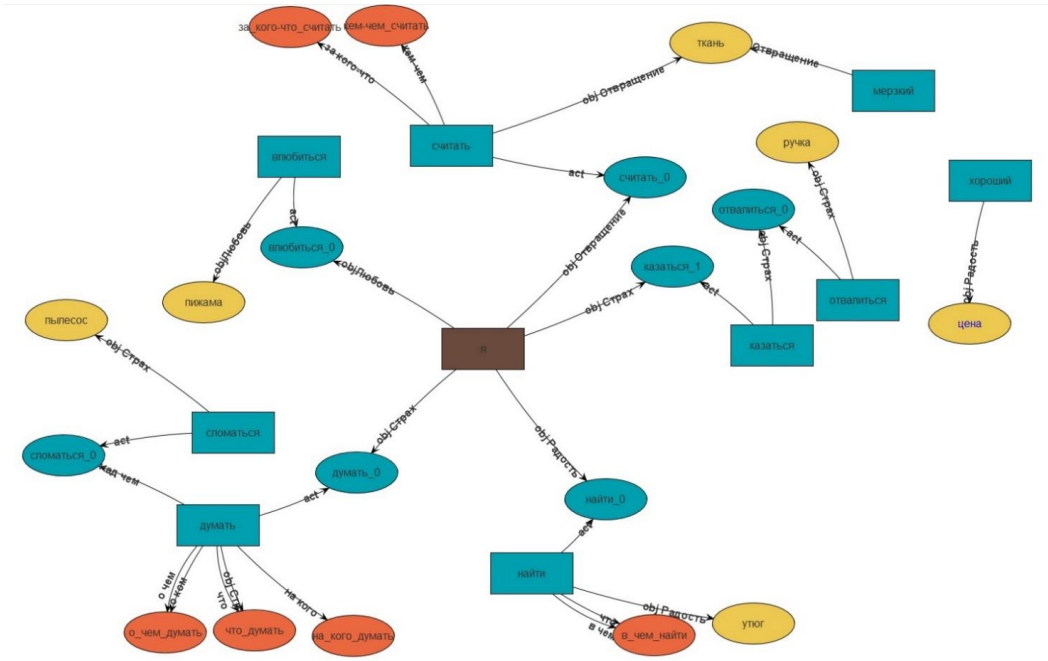


Рис.2. Атомарная диаграмма.

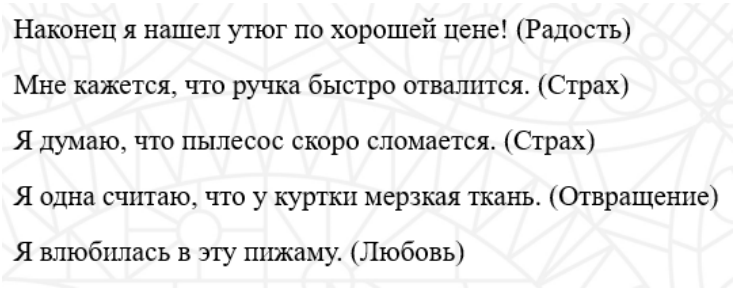


Рис.3. Предложения для построения атомарной диаграммы.

3 Заключение

Описанные выше нововведения значительно расширяют функционал программного обеспечения LogicText, добавляя возможность распознавания и анализа эмоций, содержащихся в текстах на естественном языке.

Литература

[1] Обзор подходов для автоматического распознавания эмоций в текстах [электронный ресурс] // https://cyberleninka.ru/article/n/obzor-podhodovdlya-avtomaticheskogo-raspoznavaniya-emotsiy-v-tekstah

[2] Использование современных алгоритмов машинного обучения для задачи распознавания эмоций [электронный ресурс] // https://cyberleninka.ru/article/n/ispolzovanie-sovremennyh-algoritmovmashinnogo-obucheniya-dlya-zadachi-raspoznavaniya-emotsiy/viewer

[3] Использование нейросетевых технологий в задаче автоматического распознавания эмоций / С. О. Целикова, Я. П. Горожанкин, А. О. Иванов [и др.]. — Текст: непосредственный // Молодой ученый. — 2019. — № 26 (264). — С. 59-61. — URL: https://moluch.ru/archive/264/61173/

[4] Махасоева О. Г., Пальчунов Д.Е. Программная система построения атомарной диаграммы модели по тексту естественного языка. Свидетельство о государственной регистрации программы для ЭВМ № 2014619198, зарегистрировано 10.09.2014.

Разработка веб-приложения для управления регистрационными данными домашних животных

Осипов Александр Александрович, Болотов Кирилл Юрьевич

Новосибирский Государственный Университет, ул. Пирогова, 1., г. Новосибирск, 630090, Россия.
Novosibirsk State University, 1 Pirogova st., Novosibirsk, 630090, Russia.

a.osipov4@g.nsu.ru, k.bolotov@g.nsu.ru

По данным ВЦИОМ, у 68% россиян в семье есть домашние животные, в основном — кошки и собаки [**1**]. Однако каждый день в России теряется от 300 до 1000 собак и кошек, и только 20% из них возвращается к хозяевам, остальные же остаются на улице, попадают в приюты или погибают. Поиск потерявшегося питомца может занимать продолжительное время, а отсутствие единой базы идентификаторов домашних животных не позволяет сразу связаться с владельцем. При этом наиболее частые причины, из-за которых теряются домашние животные, следующие:

- прогулка с питомцем без поводка;
- стресс питомца из-за салютов во время праздничных мероприятий или грозы;
- неправильная перевозка животного в переноске;

Важно также отметить, что часто владельцы питомцев не задумываются о возможной пропаже любимца: они не помещают «адресники» (небольшая подвеска для ошейника, на которой указана информация о кличке животного и о контактных данных хозяина), а также не чипируют своих животных (чип – виртуальный паспорт питомца, выполняющий функцию «адресника»). Однако нужно учесть, что питомец может потерять ошейник с «адресником» в отличие от чипа (потеря которого представляется трудновыполнимой задачей).

Цель нашей работы заключается в разработке сервиса, который будет предоставлять владельцам домашних животных возможность добавлять своих питомцев в нашу базу данных, а пользователям, которые нашли потерянное животное с чипом или клеймом - способ быстро получить контактные данные для связи, оставленные владельцем питомца.

На сегодняшний день реализована основная функциональность веб-приложения. Владельцы домашних животных могут регистрироваться на сайте, оставлять свои контакты и добавлять информацию о питомцах. Пример такой информации: кличка, порода, особенности ухода (это могут быть: аллергии, необходимые лекарства и т. д.). Неавторизованные пользователи, указав чип потерявшегося животного, способны найти важную информацию о нём, а также контактные данные хозяина.

В дальнейшем мы планируем развивать наш сервис: например, имплементировать REST API, реализовать элементы безопасности веб-приложения, поддерживать несколько пользовательских ролей, исследовать статистические данные, полученные с сайта, а также использовать blockchain-арбитраж для оформления сделок по смене владельцев питомца.

Благодарности

Работа выполнена при поддержке АО "СберТех" и ПАО "СБЕРБАНК" на базе студенческой учебно-научной лаборатории факультета информационных технологий Новосибирского Государственного Университета.

Литература

[1] ВЦИОМ, “Россия - страна котов!, 27.11.2019”, https://wciom.ru/analytical-reviews/analiticheskii-obzor/rossiya-strana-kotov

Анализ качества размещения пользователя в кадре с помощью программной системы

Панова Вероника Евгеньевна

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия
Novosibirsk State University, 2 Pirogova st., Novosibirsk, Russia

v.panova1@g.nsu.ru

С развитием технологий видеосвязи, использование этого средства общения стало все более распространенным, особенно в контексте дистанционного обучения и рабочих совещаний [1,2]. Однако, чтобы максимально эффективно использовать видеосвязь, важно не только обеспечивать техническую стабильность, но и создавать условия, максимально приближенные к офлайн встречам, включая визуальную часть общения.

Помочь решить эту проблему может программная система для анализа видеопотока и оценки качества поведения в кадре. Основная цель - создать систему, способную анализировать видеопоток с точки зрения соблюдения ряда критических параметров, включая взаимодействие с камерой, жестикуляцию и положение в кадре. Эта система позволит пользователям получать ценную обратную связь и рекомендации после каждой дистанционной встречи, что, в свою очередь, способствует совершенствованию навыков эффективного общения посредством видеосвязи.

В рамках работы были изучены современные методы и решения, применяемые для анализа видеопотока [3] и оценки качества онлайн-взаимодействия. Разработан алгоритм, способный определить ключевые аспекты ошибок или недочетов при онлайн встречах. На основе этого алгоритма создана программная система, которая может быть внедрена для повышения качества онлайн-коммуникаций и обучения.

Литература

[1] Подкур Т.М., Шестакова Е.А., Яхъяева Г.Э., Модуль обработки контента системы дистанционного обучения ChooseYourCourse. // Материалы Международной конференции «Знания-Онтологии-Теории» (ЗОНТ-2021), Новосибирск, 2021, с. 237-247.

[2] G. Yakhyaeva and T. Podkur, Knowledge base for the system of distance education ChooseYourCourse, // 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Yekaterinburg, Russian Federation, 2022, pp. 860-864.

[3] Шилович О.Б., Гуляй В.Г., Марков А.И., Шаповалов Д.А. К вопросу улучшения качества анализа продукции путём применения алгоритмов компьютерного зрения // ЦИТИСЭ. 2023. № 1. С. 191-201.

Настройка среды для тестирования и развёртывания смарт-контрактов и децентрализованного приложения на основе блокчейна NEO

Петров С.Е.

Новосибирский государственный университет, ул. Пирогова 1, г. Новосибирск, 630090, Россия.
Novosibirsk State University, 1 Pirogova st., Novosibirsk, Russia

s.petrov1@g.nsu.ru

С развитием блокчейн-технологий становится все более важной задачей обеспечить безопасную и эффективную разработку децентрализованных приложений и смарт-контрактов. Однако для этого требуется создание изолированных сред разработки, отладки и тестирования, чтобы минимизировать риски и обеспечить надежное функционирование смарт-контрактов на реальных блокчейн-платформах.

Целью работы является настройка среды для тестирования смарт-контрактов на платформе NEO и автоматизации процесса разработки с использованием непрерывной интеграции и непрерывной доставки (CI/CD).

Для обеспечения изоляции и безопасности смарт-контрактов на платформе NEO создана собственная приватная блокчейн сеть с использованием официального инструмента NEO-CLI. Этот процесс включал в себя настройку узлов сети и установку необходимых зависимостей для обеспечения консенсуса и доступа к RPC API. Такая изолированная среда позволила проводить тестирование и отладку, минимизируя риски и обеспечивая надежное функционирование смарт-контрактов на основе NEO в реальных условиях.

Для автоматизации процессов разработки и тестирования внедрена непрерывная интеграция и непрерывная доставка (CI/CD) с использованием платформы GitLab. Создан пайплайн, включающий в себя несколько этапов, таких как сборка приложения, запуск тестов, развертывание смарт-контрактов и создание релизов. Это решение позволило автоматизировать весь процесс, сократить время развертывания и снизить вероятность человеческих ошибок. Теперь разработка и доставка нашего проекта стали более надежными и эффективными.

В заключении, настройка изолированной среды для тестирования смарт-контрактов и внедрение непрерывной интеграции и доставки оказались ключевыми шагами для обеспечения качественной и безопасной разработки децентрализованных приложений. Эти методы позволяют минимизировать риски, ускорить процесс разработки и обеспечить надежное функционирование проектов на основе блокчейн-технологий.

Благодарности

Работа выполнена при поддержке АО "СберТех" и ПАО "СБЕРБАНК" на базе студенческой учебно-научной лаборатории факультета информационных технологий Новосибирского государственного университета.

Поиск и обработка данных для модели прогнозирования продовольственных запасов Российской Федерации.

П. В. Попова

Новосибирский государственный университет, ул. Пирогова, д. 1, г. Новосибирск, 630090, Россия.

p.popova1@g.nsu.ru

Задача прогнозирования будущих значений величин на основе их предыдущих значений является неотъемлемой частью эффективного финансового планирования и управления производством. Она позволяет определить потенциальные риски и возможности, а также принять обоснованные решения для достижения поставленных целей. Для этого необходим этап обработки информации, включающий поиск, унификацию и дополнительные исправления данных.

Новизна: в настоящее время не существует модели, способной предсказывать изменения показателей, влияющих на баланс продовольственных продуктов в России.

Цель работы: Одним из ключевых этапов использования машинного обучения для прогнозирования, основанного на модели временных рядов, является поиск качественных и точных данных. Основная задача заключалась в составлении такой базы данных.

В данном случае, изначально была использована общедоступная база данных ООН, содержащая данные с 1992 по 2020 годы. Она включала 74 вида продуктов и более 14 категорий, связанных с их товарооборотом. В ходе проведения регрессионного анализа и исследования коллинеарности категорий были выявлены незначимые параметры, которые были исключены. В связи с отсутствием данных во многих категориях возникла потребность в поиске информации в других официальных источниках. В процессе исследования общедоступных источников было принято решение ограничиться 20 основными продуктами питания, которые входят в минимальный продуктовый набор. Благодаря информации с Росстата и официальных статистических сборников по социально-экономическим показателям недостающие данные были обработаны и включены в базу. Для обучения модели необходимы большие наборы данных, поскольку это непосредственно влияет на точность предсказания. Было принято решение исследовать статистику по производству с 1960 по 1991 годы на территории РСФСР и добавить ее в базу данных. Вследствие этого удалось увеличить объем информации в два раза.

Таким образом была собрана достоверно-точная база данных, которая включает информацию о производстве и использовании основных продуктов питания на территории Российской Федерации. В дальнейшем планируется изучение экономической составляющей потребления и на основе полученных знаний расширение набора данных.

Благодарности

Работа выполнена при поддержке *АО "СберТех"* и *ПАО "СБЕРБАНК"* на базе студенческой учебно-научной лаборатории *факультета информационных технологий Новосибирского государственного университета*

Литература

1. Центральное статистическое управление при Совете Министров РСФСР Народное хозяйство РСФСР. - 1-е изд. - М.: Статистика, 1977. - 369 с.

2. Food Balances (-2013, old methodology and population) // FAOSTAT URL: https://www.fao.org/faostat/en/#data/FBSH (дата обращения: 10.07.2023).

3. Промышленное производство // Федеральная служба государственной статистики URL: https://rosstat.gov.ru/enterprise_industrial (дата обращения: 23.07.2023).

Анализ эвристических правил объединения признаков в табличных данных

Радеев Никита Андреевич

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия

n.radeev@g.nsu.ru

Исследование данных путем скрещивания признаков [1] представляет собой одну из основных методик в области построения признаков. При объединении нескольких признаков, которые, возможно, не выглядят важными согласно определенным критериям оценки, возникает потенциал создания более существенного признака, превосходящего любой из составляющих его элементов. Поиск оптимальных комбинаций, то есть, наиболее важных, часто является вычислительно сложной задачей из-за экспоненциального увеличения числа комбинаций при увеличении числа участвующих в этом процессе признаков [2]. Поэтому в практике используются эвристики. Тем не менее, существует интересное противоречие между двумя эвристическими принципами. Первый формулируется как «Вся необходимая информация концентрируется исключительно в важных признаках, поэтому можно просто отбросить неважные». Второй так: «Комбинация неважных признаков иногда может иметь исключительную значимость, и поэтому признаки с низкой важностью не должны быть исключены из рассмотрения.»

С точки зрения здравого смысла очевидно, что оба эти утверждения не верны в общем случае, однако применительно к реальным данным [3] нельзя прийти к однозначному выводу, так как данные реального мира — это небольшое подмножества множества всех возможных данных, а следовательно, они могут обладать некими свойствами, благодаря которым одно из правил будет выполняться чаще другого.

В данном исследовании был проведен анализ корпуса данных реального мира с использованием критериев оценки важности признаков и скрещивания, с целью определить, какое из эвристических правил чаще верно на практике [4, 5].

Литература

[1] «Характерные кресты | Machine Learning», Google Developers. https://developers.google.com/machine-learning/crash-course/feature-crosses/video-lecture?hl=ru (просмотрено 16 февраль 2023 г.).

[2] Luo и др., «AutoCross: Automatic Feature Crossing for Tabular Data in Real-World Applications». arXiv, 15 июль 2019 г. doi: 10.48550/arXiv.1904.12857.

[3] Liu F. and Demosthenes P. Real-world data: a brief review of the methods, applications, challenges and opportunities» \\ BMC Medical Research Methodology, т. 22, вып. 1, с. 287, ноя. 2022, doi: 10.1186/s12874-022-01768-6.

[4] Радеев Н. А. Прозрачное уменьшение размерности с помощью генетического алгоритма \\ Вестник НГУ. Серия: Информационные технологии. Том 21, № 1, 2023, сс. 46-61.

[5] Radeev N. A. Approach to Research Feature Interactions. \\ 2023 IEEE 24th International Conference of Young Professionals in Electron Devices and Materials (EDM), Novosibirsk, Russian Federation, 2023, pp. 1720-1724, doi: 10.1109/EDM58354.2023.10225150.

Мобильное приложение рецептов с голосовым помощником Talky Chef

Садриев В. Д., Иванов А. И., Пальчунова О. Д., Пашкова А. С.

Новосибирский государственный университет, ул. Пирогова, 1, г. Новосибирск, 630090, Россия

v.sadriev@g.nsu.ru, a.ivanov15@g.nsu.ru, o.palchunova@g.nsu.ru, a.pashkova@g.nsu.ru

Аннотация. *В статье описан процесс проектирования и разработки мобильного приложения Talky Chef. Отличительной особенностью данного приложения является наличие модуля рекомендательной системы, позволяющей адаптировать подборку рецептов под каждого конкретного пользователя и модуля голосового управления, позволяющего взаимодействовать с ним при помощи голосовых команд, что особенно удобно при приготовлении еды, когда руки заняты.*

Ключевые слова: мобильное приложение, голосовое управление, рекомендательная система, кулинария, рецепт

1 Введение

На сегодняшний день, в эпоху быстрого развития технологий и автоматизации, голосовые помощники стали неотъемлемой частью повседневной жизни. Они значительно упрощают ряд задач и сделали взаимодействие с техникой более естественным и удобным. Однако, когда речь идет о кулинарии, универсальные голосовые ассистенты оказываются несколько ограниченными и недостаточно адаптированными к специфике этой области.

В такой сфере, как кулинария, где каждый шаг, каждый ингредиент имеет значение, возможность взаимодействовать с устройствами приготовления пищи и получать экспертные рекомендации и рецепты голосом от кулинарного ассистента может радикально улучшить опыт и результаты домашней кулинарии. Это позволяет удовлетворить актуальные потребности пользователей и создать инновационные решения в сфере голосовых технологий.

В ходе приготовления пищи возникает значительное неудобство при взаимодействии с электронными устройствами из-за наличия грязи и ингредиентов на руках. Пользователям часто приходится прерывать процесс кулинарных действий, чтобы воспользоваться сенсорными экранами или клавиатурами на своих устройствах. Эта необходимость в постоянной чистке или вытирании рук приводит не только к неудобству, но и создает потенциальные риски в контексте соблюдения санитарных норм и правил гигиены на кухне.

Пользователи всё чаще обращаются к электронным рецептам, таймерам, заметкам и другим кулинарным приложениям на своих устройствах, но ограничение во взаимодействии из-за грязных рук затрудняет их использование и мешает максимально эффективно и комфортно реализовывать кулинарные идеи.

Анализ рынка показал растущий интерес к голосовым технологиям и специализированным приложениям в сфере кулинарии [1]. Существует ряд конкурентов, предоставляющих голосовые ассистенты для кулинаров: Алиса [2], Hi-Chef [3], Kitchen Stories [4]. Однако они ограничены базовым функционалом и имеют ограниченные ресурсы для рецептов.

В данной статье описывается мобильное приложение рецептов Talky Chef с интегрированным интеллектуальным голосовым ассистентом. Это приложение позволяет взаимодействовать с ним при помощи голосовых команд, что особенно удобно при приготовлении еды, когда руки заняты. Talky Chef выделяется своей глубокой интеграцией в кулинарную среду, обширной базой данных и высокой степенью персонализации для пользователей, что делает его конкурентоспособным на рынке голосовых ассистентов в кулинарии.

Преимущества "Talky Chef" перед универсальными голосовыми ассистентами заключаются в специализации на кулинарной сфере. Помимо стандартного голосового управления, приложение предлагает подборки блюд и имеет обширную базу данных высококачественных рецептов. Это делает "Talky Chef" идеальным компаньоном для кулинаров и тех, кто хочет приготовить что-то особенное.

2 Архитектура приложения

Разработанное мобильное приложение состоит из следующих модулей:

1. База данных
2. Модуль рекомендательной системы
3. Модуль голосового управления

На Рис.1 представлена архитектура приложения.

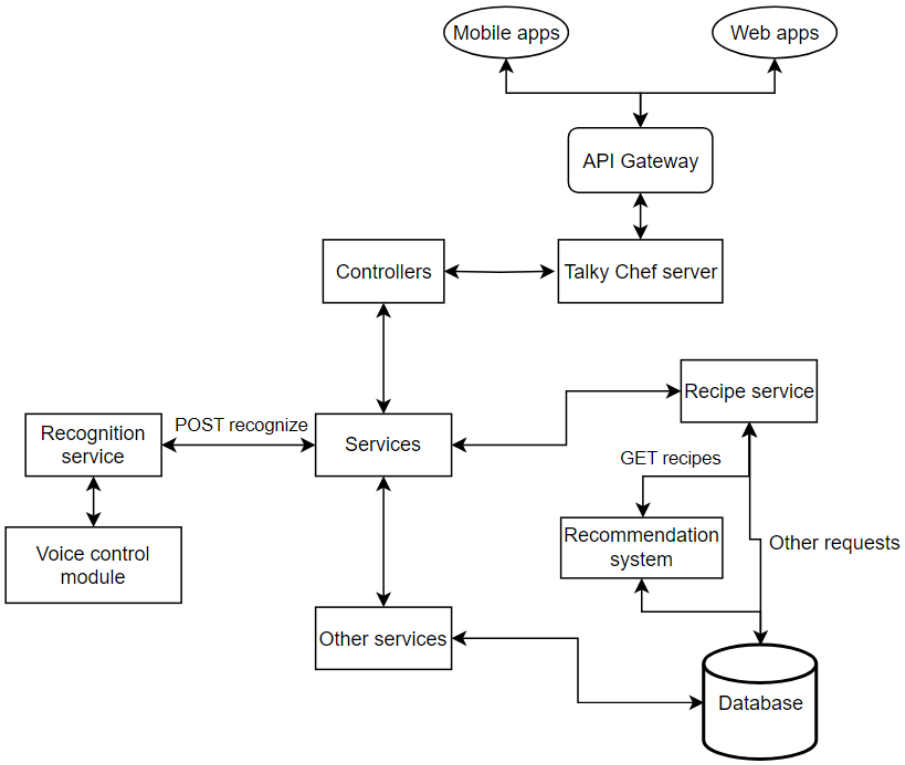


Рис. 1. Архитектура мобильного приложения Talky Chef.

Опишем более подробно каждый из модулей системы Talky Chef.

База данных. В качестве СУБД в проекте используется PostgreSQL — мощная и развивающаяся открытая реляционная система. На данный момент в базе данных существует 18 таблиц, среди которых: рецепты, шаги рецептов, коллекции, комментарии, оценки, пользователи и прочее.

Основной тут является таблица рецептов, которая хранит название рецепта, id изображения, которое будет показано на главном экране, id автора, КБЖУ и количество времени на приготовление блюда. На данный момент в базе данных содержится около 1000 рецептов. Каждый рецепт имеет определенное количество шагов, каждый шаг имеет описание и необязательное изображение. Ингредиенты не фиксированы, если в таблице нет данного ингредиента, то при добавлении рецепта добавляется и новый кортеж с ингредиентом.

Немаловажной является таблица пользователей, где хранятся пароль и логин каждого зарегистрированного пользователя. Кроме этого, пользователь может оставлять комментарии и оценивать рецепты.

На Рис.2 показана схема базы данных.

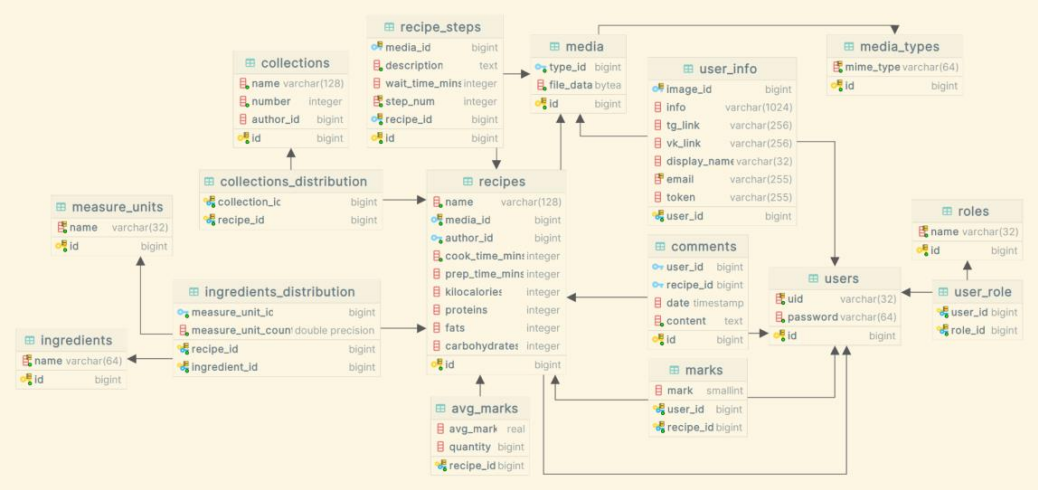


Рис. 2. База данных мобильного приложения Talky Chef.

Модуль рекомендательной системы. Рекомендательная система реализована на основе алгоритма коллаборативной фильтрации Slope One, который прогнозирует оценки пользователей на основе их предыдущих оценок и оценок других пользователей с похожими предпочтениями[5, 6]. По данным об оценках строятся 2 матрицы для всех пар рецептов: матрица различий и частот. Матрица различий содержит информацию о том, насколько отличаются оценки пользователей для конкретной пары рецептов, а матрица частот — сколько пользователей оценили данную пару рецептов. В рамках рекомендательной системы Slope One, для прогнозирования оценок рецептов, которые пользователь еще не оценил, используется матрица частот freq в сочетании с матрицей различий diff. При оценке пользователем какого-либо рецепта система прибегает к данным из матрицы freq для оценки достоверности этих оценок. Для каждого пользователя вычисляется прогнозируемая оценка неоцененного рецепта, основываясь на оценках других рецептов.

Модуль голосового управления. В приложении предусмотрено 9 голосовых команд: вперёд, назад, таймер, стоп, выйти, сброс, читай, не читай, начало, а также их вариации. Для распознавания была написана нейронная сеть. Рассмотрим данный процесс более детально.

Рассматривать весь акустический сигнал целиком сразу — некорректно. Очевидно, что слово “вперёд” можно произнести разными способами, с различным произношением, оттенками и т.п. Поэтому изначальный сигнал будет биться на небольшие промежутки по 25 мс. Каждый сигнал может являться какой-то фонемой, в то же время, одна фонема может заключаться сразу в нескольких сигналах. По звуковой дорожке длиной 25 мс нужно определить, есть ли фонема или нет, и если есть, то какая.

С помощью дискретного преобразования Фурье можно разложить сложный входной сигнал в сумму простых синусоидальных гармоник с заданной частотой и амплитудой. Т.к. для наших целей нужно выделять конечное число гармоник, частота которых фиксирована, а амплитуда зависит от входного сигнала, то можно получить амплитудно-частотную характеристику для данного сигнала длиной в 25 мс. Амплитудно-временная характеристика также известна.

Теперь можно построить черно-белое изображение, где координата по горизонтали отвечает за время, а по вертикали — за частоту. Рассматриваемые частоты фиксированы. Интенсивность пикселя на позиции (*x*, *y*) от 0 до 255 будет определяться значением амплитуды гармоники с частотой *y* в интервал времени *x*.

По сути, получили спектрограмму, но она излишне выразительна. Воспользуемся MFCC(Mel Frequency Cepstral Coefficients) для уменьшения размерности данных и устранения шума [7]. В итоге владеем, по существу, изображением звука. Чтобы преобразовать это в конкретную фонему, воспользуемся Convolutional блоком нейронной сети. В качестве конкретной архитектуры для данного сверточного блока имеет смысл воспользоваться VGG-16 CNN Architecture.

Стоит отметить, что рассматривать фонемы в отрыве друг от друга будет упущением, т.к. для фонем, произносимых в текущий момент есть некая зависимость от фонем, произнесенных ранее. Чтобы отслеживать данные зависимости, нужны слои нейронной сети, способные запоминать контекст. Для этой задачи подходит LSTM(Long short-term memory) нейронная сеть, способная запоминать контекст на 30-50 элементов последовательности.

Результатом работы данного модуля является вектор вероятностей — уверенности сети в той или иной команде. Для получения такого вектора достаточно в конце разместить Softmax Layer.

Архитектура данного модуля показана на Рис. 3

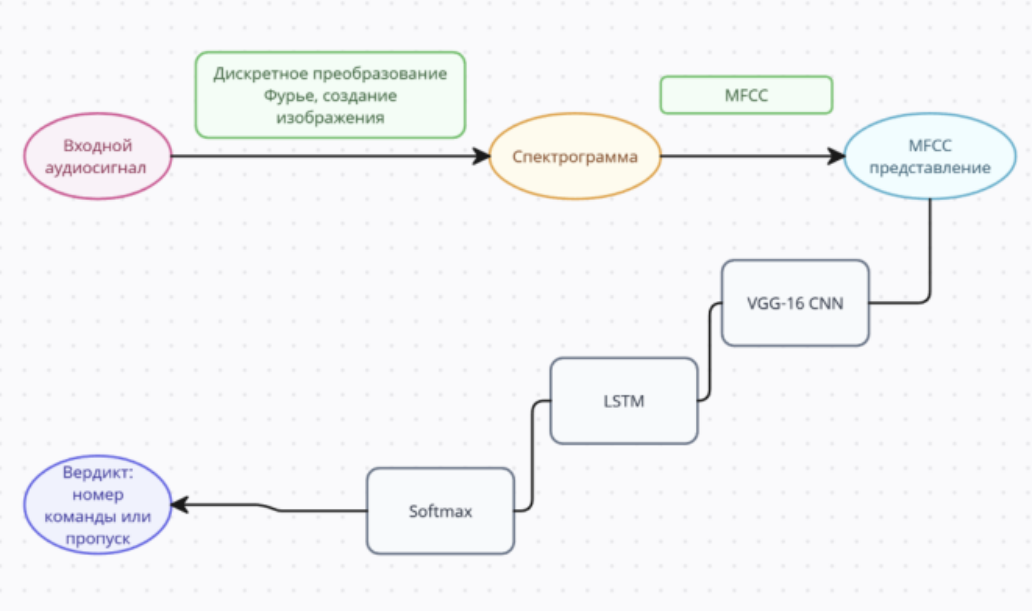


Рис. 3. Архитектура голосового модуля.

3 Интерфейс

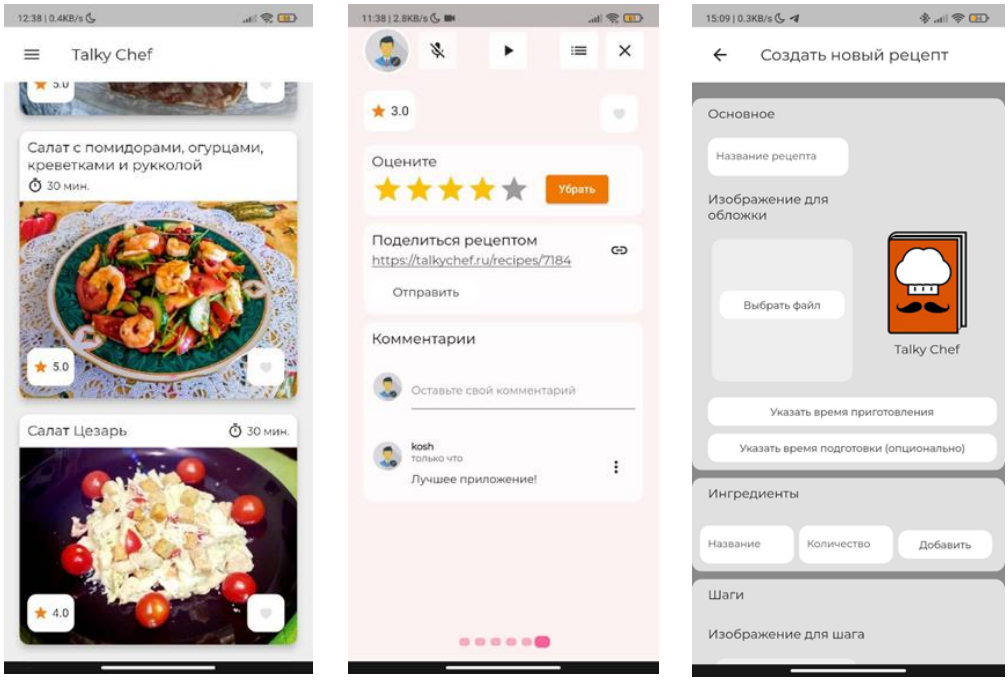
Открыв приложение, пользователь оказывается на главной странице, которая показана на Рис.3. Тут он видит рецепты, которые ему предоставила рекомендательная система. Если пользователь авторизован, то у него виден свой пакет рецептов. В ином случае рекомендации для всех одинаковы. Также, вне зависимости от того, зашёл ли человек в свой аккаунт или нет, он сможет найти любое интересующее его блюдо, воспользовавшись поиском. Если же человек не знает, какую именно еду он хочет приготовить, он может выбрать из конкретных подборок.

Когда пользователь определился с рецептом, он может его открыть. На первой странице будут предоставлены все ингредиенты и их количество, время подготовки и приготовления. Дальше последуют страницы с шагами. На них будет изображение действия, а также его описание. Как говорилось ранее, наше приложение удобно тем, что им можно пользоваться с мокрыми и грязными руками, не касаясь экрана. Следует только сказать команду, и система сама

начнет читать информацию со страницы. Если пользователь что-то пропустил, ему предоставляется возможность повторить текст или вовсе остановить прочтение.

Но не только с этой проблемой сталкиваются большинство хозяек. Некоторые этапы приготовления требуют времени. Например, необходимо время, чтобы закипела вода. И тут опять на помощь придет наше приложение. Пользователю не придется искать таймер и грязными руками его настраивать. Он уже имеется на каждом шаге, где необходим. Остается только сказать, чтобы таймер включился.

На последнем слайде, показанном на Рис.4 предлагается оценить рецепт и оставить комментарий. Это удобно для других пользователей, а также поможет выбрать правильные рекомендации для каждого. Но данная функция доступна только для тех, кто уже зашел в систему.



а) **б)** **в)**

Рис 4. Интерфейс мобильного приложения Talky Chef: а) главная страница; б) страница обратной связи; в) страница создания рецептов.

После авторизации открываются новые удобные функции. Для этого можно зарегистрироваться или войти в уже существующий аккаунт. Теперь у пользователя есть своя страница, где высвечиваются его персональные данные, а также созданные рецепты и коллекции. По желанию можно добавить ссылки на социальные сети, информацию о себе или изменить пароль.

На Рис.5 показан реализованный функционал, позволяющий пользователям делиться своими рецептами. В нем есть возможность написать всю нужную информацию, описать любое количество шагов, прикрепить таймеры, добавить картинки. Таким образом, на странице пользователя появятся новые рецепты. Но можно не только выставлять свои, но и смотреть блюда на чужих страницах, а понравившиеся рецепты сохранять к себе.

Чтобы было легче найти правильный рецепт, предлагается создать категории. Например, сладкое, праздничное, закуски и т.д. Так будет проще не запутаться в большом количестве сохраненных рецептов. И, как говорилось ранее, появится возможность оценить и оставить комментарии.

4 Реализация приложения

Аутентификация и авторизация пользователей реализована с использованием JWT(JSON Web Token). При регистрации пользователя сервер шифрует пароль и сохраняет всю информацию, а дальше генерирует уникальную пару токенов и отдаёт клиенту. Если пользователь забыл свой пароль, то предусмотрена функция смены пароля по введенному при регистрации адресу электронной почты.

Мобильное приложение написано на языке Dart с использованием Flutter. В качестве языка программирования серверной части был выбран язык Java. В качестве системы контроля версий выбран Git, для тестирования конечных точек — Postman.

Основным элементом проекта является Spring — универсальный фреймворк с открытым исходным кодом для Java-платформы, включающий в себя различные компоненты: Spring Boot, Spring Data, Spring MVC и т.д.

Для сборки проекта был использован Gradle, позволяющий с легкостью добавлять зависимости, собирать и запускать приложение. С помощью него добавлены нужные компоненты Spring, а также упрощающие написание кода библиотеки: ModelMapper и Lombok.

5 Заключение

Таким образом, приложение сделано очень удобно как для тех, кто хочет научиться готовить, так и для тех, кто уже овладел этим навыком и желает поделиться с другими. В дальнейшем планируется увеличить возможности голосовых команд, реализовать сайт, дублирующий функционал приложения, а также добавить функцию распознавания жестов, чтобы управление могло осуществляться даже без голоса.

6 Благодарности

Работа выполнена при поддержке АО "СберТех" и ПАО "СБЕРБАНК" на базе студенческой учебно-научной лаборатории факультета информационных технологий Новосибирского государственного университета.

Литература

[1] 50 лучших идей приложений для стартапа в 2023 году. URL: https://appmaster.io/ru/blog/50-luchshih-idej-prilozhenij-dlya-startapa-v-2022 ?

[2] Навык Алисы. URL: https://dialogs.yandex.ru/store/skills/bc631b01-umnye-recepty

[3] Hi-chef. URL: https://hi-chef.ru/

[4] Kitchen Stories. URL: https://www.kitchenstories.com/en

[5] Pu Wang, HongWu Ye, A Personalized Recommendation Algorithm Combining Slope One Scheme and User Based Collaborative Filtering, IIS '09, 2009.

[6] DeJia Zhang, An Item-based Collaborative Filtering Recommendation Algorithm Using Slope One Scheme Smoothing, ISECS '09, 2009.

[7] Xuechen Liu, Md Sahidullah, Tomi Kinnunen, Learnable MFCCs for Speaker Verification, ISCAS '09, 2021

Выборочное подключение процессов к VPN посредством перехвата системных вызовов в ОС Linux

Смоляков П.Е.[1]

[1]*Новосибирский государственный университет*

p.smolyakov@g.nsu.ru

Ключевые слова: системные вызовы, VPN (виртуальная частная сеть), Wireguard

Идея выборочного подключения процессов к VPN может быть использована во многих прикладных задачах. В частности:

- многим в сфере IT нужно подключаться к корпоративным VPN сетям, при этом нельзя либо не хочется передавать через них весь трафик
- у пользователя может не быть прав суперпользователя, нужна поддержка работы из пользовательского пространства
- необходимо использование двух и более VPN-подключений параллельно

Отсутствие возможности выборочно подключить ограниченное число процессов к VPN-туннелю в определенных ситуациях создаёт сложности для пользователей. В рамках работы предлагается использовать механизм перехвата системных вызовов для решения поставленной проблемы.

Вдохновителем работы является достаточно известный в области информационной безопасности инструмент, называющийся Proxychains и также позволяющий запускать дочерние процессы, осуществляя TCP-подключения через прокси (socks, http(s)) без ведома запускаемого процесса.

Цель: разработка инструмента, позволяющего выборочно подключать процессы к VPN посредством перехвата системных вызовов.

Для выполнения работы необходимо было выполнить следующие задачи:

- Обеспечить перехват системных вызовов с помощью LD_PRELOAD (для программ с динамическим связыванием).
- Написать подменяющую библиотеку (shared object) на языке C. Она отправляет сообщения в формате BSON с аргументами перехваченных вызовов IPC серверу.
- Написать IPC (inter process communication) сервер на языке Go, принимающий аргументы перехваченных вызовов и осуществляющий общение с удалённым сервером.
- Реализовать создание и общение через VPN-туннель в рамках процесса посредством gVisor netstack - userspace реализации сетевого стека, предоставляющей возможность создавать VPN-туннель в рамках процесса без создания псевдоустройств.

Исходный код утилиты доступен для использования и представлен в репозитории github.com/vpnchains/vpnchains-wireguard. В ветке master размещена полностью функционирующая версия для TCP с фильтрами для внешних UDP датаграмм. Ветка development-udp содержит версию для TCP+UDP с некоторыми недоработками.

Работа выполнена при поддержке АО "СберТех" и ПАО "СБЕРБАНК" на базе студенческой учебно-научной лаборатории факультета информационных технологий Новосибирского государственного университета.

«Аналитик - фасилитатор командной работы»

Сухих Александр Олегович

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия
Novosibirsk State University, 2 Pirogova st., Novosibirsk, Russia

a.sukhikh1@g.nsu.ru

Сегодня IT – сфера входит в одну из самых быстроразвивающихся отраслей российской экономики. Данный тезис можно обосновать различными аспектами: поддержка сотрудников со стороны государства, формирование государственных практик, позволяющих получить it-специальность через различные формы переподготовки, интеграция it-решений в повседневные задачи граждан. [1] К таким примерам можно отнести: постановку диагноза заболевания с помощью различных приложение, оформление кредитных обязательств, заказ продуктов и многое другое.

Для создания it-услуги или it-продукта требуется идея, а также команда, которая осуществит процесс доставки финального решения до пользователя. Процесс создания продукта всегда связан с огромной эмоциональной нагрузкой, которая ложится на команду и постепенно разрушает здоровый рабочий процесс. Негативные изменения рабочего процесса влияют на показатели эффективности команды, на их вовлеченность и желание достигать определенных целей, влияющих на успех продукта. Все это заставляет менеджеров, а также неравнодушных членов команды искать решение данной проблемы, так как взаимодействие членов команды является основополагающим фактором пропорционально влияющем на положительный результат рабочего процесса.

Цель работы – рассмотреть роль аналитика, как фасилитатора, сформировать определение фалиситатора, определить задачи и предложить решения проблем, связанных с взаимодействием в команде.

В ходе изучения материалов, пользовательского опыта и скрининг-интервью обнаружилось, что в современных проектных командах роль: «Фасилитатор» не применяется или вовсе не знакома. Ответы, полученные с помощью скрининга, показали, что участники команды не готовы делиться проблемами, связанными с внутренним взаимодействием. Из этого можно сделать вывод, что решения, принимаемые владельцем продукта / менеджером, могут быть ошибочны. На основе изученных материалов, успешных пользовательских практиках было сформировано определение: «Фасилитатор» и даны точные обязанности роли.

Фасилитатор – вовлеченный член команды, который взаимодействуют со всеми участниками и руководит групповым мероприятиям, управляет дискуссией так, чтобы группа эффективнее достигла поставленных целей.

К задачам фасилитатора можно отнести свод обязанностей:

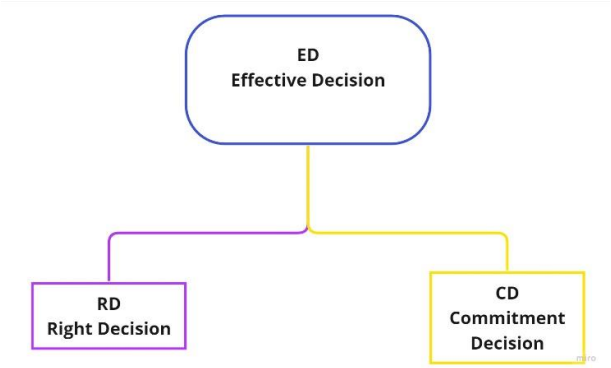
- Управление эмоциями, эмоциональное выгорание;
- Урегулирование конфликтов в команде проекта;
- Командообразование;
- Преодоление основных барьеров в коммуникативном процессе;
- Методы психологического воздействия;
- Манипуляции и способы защиты от них;
- Сохранение нейтральной позиции в спорах.

Для роли фасилитатора аналитик подходит лучше других членов команды, если нет выделенной роли – scrum-мастер. Аналитик плотно взаимодействую с каждым членом команда, заинтересован в правильной технической реализации проекта, а также не обременен задачами управления (роль менеджера).

Также стоит отменить формулу Майкла Вилкинсона, которая говорит о том, что фасилитация – это возможность выращивать в компании культуру вовлеченности и ответственности, представленную в виде формулы: [2]

Эффективное решение = Правильное управление x Уровень приверженности (схема №1)

Схема №1



По мнению Вилкинсона, много решений и изменений в компаниях проваливаются не потому, что они были не верны. Вполне может быть, что первый компонент этой формулы (правильность решения) присутствовал. Но люди, которые его реализовывали - скорее всего, не участвовали в принятии решения - и тем самым не были ему привержены. [3]

Их уровень приверженности может быть очень низким, потому что им непонятно, почему так решили: они не принимали участия ни в сборе вариантов, ни в расчете, что будет лучше, они не знали предысторию и не обдумывали риски. Поэтому данные члены команды не готовы двигать идею вперед и сдаются при первых препятствиях. Часто у таких членов команды могут быть свои решения, в результате чего они могут саботировать присланные.

Литература

[1] Министерство науки и высшего образования Российской Федерации. Режим доступа: https://minobrnauki.gov.ru/press-center/news/nauka-i-obrazovanie/39914/

[2] Michael Wilkinson. The secrets of facilitation: The s.m.a.r.t. guide to getting results with groups, 2019, p. 125-270.

[3] Jon Bronson. The Executive Guide to Facilitating Strategy, 2011, p. 149.

Знания-Онтологии-Теории (ЗОНТ-2023)

Оптимизация рациона питания на основе индивидуальных параметров человека

Тимофеев Артем Тимофеевич

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия
Novosibirsk State University, 2 Pirogova st., Novosibirsk, Russia

a.timofeev@g.nsu.ru

В наше время, с ростом осознания важности здорового питания, а также учетом уникальных потребностей и характеристик каждого индивида, обращение к индивидуализированным диетическим подходам становится настоятельной необходимостью. Оптимизация рациона питания в соответствии с потребностями и целями каждого человека является ключевой задачей в поддержании общего благополучия и предотвращении различных заболеваний.

Существующие сервисы по составлению питания учитывают индивидуальные потребности и параметры каждого человека, такие как пол, возраст, физическая активность и другие факторы, но не все их них содержат такие параметры, как количество приемов пищи, бюджет и кулинарные способности человека. Это может приводить к составлению меню питания, которое не удовлетворяет параметрам и ресурсам конкретного человека. Семантический подход к моделированию предметных областей позволяет разрабатывать сервисы, заточенные под индивидуальные параметры пользователей [1, 2].

Генетический алгоритм представляет собой метод оптимизации, который находит приближенное оптимальное решение в условиях сложных и многопараметричных задач [3]. Этот метод вдохновлен природными механизмами эволюции и способен адаптироваться к различным вариантам рациона питания, учитывая множество факторов, влияющих на здоровье и благополучие человека.

Применение генетических алгоритмов в оптимизации рациона питания предоставляет возможность создания индивидуализированных диетических планов, которые стремятся к наиболее точному соответствию множеству параметров человека. Эти параметры включают в себя такие факторы, как рост, вес, пол, возраст, уровень физической активности, поставленные цели питания, количество приемов пищи в день, предпочтения в пище, уровень кулинарных навыков, а также бюджет. Генетический алгоритм может оптимизировать соотношение белков, углеводов, жиров и калорий в рационе питания, чтобы достичь конкретных целей, таких как снижение веса, улучшение физической формы или поддержание здоровья

Литература

[1] Yakhyaeva G.E. Application of Boolean Valued and Fuzzy Model Theory for Knowledge Base Development \\ SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings, 2019, pp. 868–871.

[2] Yakhyaeva, G., Application of the Case Models Restriction for Modeling Argumentation Reasoning \\ 2021 International Symposium on Knowledge, Ontology, and Theory, KNOTH 2021, 2021, pp. 40–44.

[3] Kramer O. Genetic Algorithm Essentials \\ Springer, Studies in Computational Intelligence, Volume 679, 2017.

Разработка интеллектуального голосового помощника для заведений общественного питания на базе SberPortal

Тихомолов Ярослав, Михалёв Макар
Новосибирский государственный университет, ул. Пирогова 1, г. Новосибирск, 630090, Россия. Novosibirsk State University, 1 Pirogova st., Novosibirsk, Russia
y.tikhomolov.@g.nsu.ru, m.mikhalev@g.nsu.ru

Почти каждый ресторанный бизнес сталкивается в ходе своего развития с многими проблемами. Первая из них, это долгое время обслуживания - из-за ограниченного количества персонала бывает сложно успеть обслужить всех на высшем уровне Вторая - текучка официантов. Это может стать угрозой лояльности для существующих клиентов. Следующая проблема - низкий средний чек. Если уменьшается чек, то это приводит к уменьшению выручки и прибыли ресторана, а получение прибыли это одна из основных целей бизнеса Четвертая проблема - контроль качества обслуживания. Конфликтные ситуации это не редкость, а за вежливостью общения бывает сложно уследить. Наш проект позволит нанимать на работу официантов с ОВЗ. Актуальна и проблема привлечения новых клиентов, ведь это также способствует увеличению прибыли.

Целью работы является разработка электронного ассистента для предоставления посетителям кафе и ресторанов возможности самостоятельного формирования заказа с помощью визуально-голосового интерфейса или telegram-бота. Создание базовой структуры проекта, включая настройку среды разработки, выбор стека технологий и инструментов. Портирование созданного приложения на sber portal и добавление голосового управления. Разработка Telegram-бота для осуществления заказов и бронирования в кафе или ресторане с удобным и интуитивно понятным пользовательским интерфейсом.

На данный момент разработано веб-приложение с возможность запуска на sber portale. Реализованы функции необходимые для составления и оплаты заказа. Также добавлено голосовое управление для имитации официанта. Реализован телеграмм бот с аналогичным функционалом.

У нас есть следующие идеи, которые мы хотим реализовать. Персональные рекомендации. Способность давать советы на основе анализа пользовательского опыта и личных предпочтений в еде. Интеграция с существующими системами. Цифровой официант будет интегрирован с используемыми во многих заведениях системами автоматизации ресторанов r_keeper и iiko.

Благодарности

Работа выполнена выполнена при поддержке АО "СберТех" и ПАО "СБЕРБАНК" на базе студенческой учебно-научной лаборатории факультета информационных технологий Новосибирского государственного университета.

Литература

[1] https://salute-developers.github.io/native-sdk/index.html

Знания-Онтологии-Теории (ЗОНТ-2023)

Выявление аномалий в работе программных систем на основе распределенной трассировки логов

Худяков Даниил Александрович

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия
Novosibirsk State University, 2 Pirogova st., Novosibirsk, Russia

d.khudyakov@g.nsu.ru

Микросервисная архитектура крупных современных приложений способствует решению множества задач, стоящих перед разработчиками. Ключевой особенностью микросервисов является то, что они могут разрабатываться и публиковаться независимо, что позволяет ускорить доставку новой функциональности до пользователей. Сокращение time to market зачастую является ключевой целью бизнеса, поэтому микросервисная архитектура заслуженно пользуется популярностью у разработчиков.

Быстрая доставка обновлений является крайне важным, но не единственным условием для успешного развития программного продукта. Чтобы обеспечить высокое качество его работы, крайне важно анализировать поведение разработанной системы для оперативного реагирования на различные сбои или же аномалии, возникающие при её эксплуатации. Аномалии неизбежно происходят и сопровождаются репутационными и финансовыми потерями. Поэтому разработчикам требуются инструменты для их обнаружения. При этом микросервисная архитектура, будучи распределенной, сильно усложняет анализ работы систем спроектированных на её основе [1].

Главным источником информации, помогающим искать аномалии, являются логи. В случае несложных и малонагруженных систем разработчики и специалисты технической поддержки могут самостоятельно проанализировать логи, чтобы найти в них отклонения от ожидаемого поведения. Однако с масштабированием системы и ростом нагрузки, это становится крайне затруднительным. По этой причине актуальной задачей является внедрение инструмента для автоматического обнаружения аномалий на основе логов, который мог бы работать с системами, имеющими микросервисную архитектуру.

Особенностью сбора логов таких систем является то, что к ним применяется распределенная трассировка. Она присваивает всем логам, относящимся к одному запросу в систему, уникальный идентификатор [2]. Получившиеся таким образом цепочки можно группировать и анализировать в зависимости от типа запроса в систему [3]. Данная работа посвящена разработке нового подхода к автоматическому выявлению аномалий в работе систем, имеющих микросервисную архитектуру, опирающегося на распределенную трассировку логов.

Литература

[1] Esposito C., Castiglione A., Choo K.R. Challenges in Delivering Software in the Cloud as Microservices. IEEE Cloud Computing 3.5, 2016, pp. 10–14.

[2] Beschastnikh I., Wang P., Brun Y., Ernst M. D. Debugging distributed systems: Challenges and options for validation and debugging. Communications of the ACM, Aug. 2016, vol. 59, no. 8, pp. 32-37.

[3] Худяков Д.А. Разработка системы выявления аномалий на основе распределенной трассировки логов. Вестник НГУ. Серия: Информационные технологии. 2023;21(1):62-72.

Разработка цифрового ассистента для подбора финансовых продуктов

Чеболтасова Софья Константиновна

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия
Novosibirsk State University, 2 Pirogova st., Novosibirsk, Russia

s.cheboltasova@g.nsu.ru

За последние несколько лет количество людей, использующих цифровых ассистентов, увеличилось в разы. Сфера разработки таких приложений не стоит на месте - помощников обучают и совершенствуют каждый день, чтобы удовлетворить запросы пользователей, но до сих пор полное взаимопонимание между пользователем и ассистентом не было достигнуто, так как цифровой помощник пока не может запоминать и анализировать сказанную пользователем информацию, например, чтобы предоставлять персональные рекомендации.

В процессе рассмотрения различных существующих на рынке решений голосовых помощников было изучено то, как самые популярные голосовые помощники оперируют контекстом диалога и работают с персонализированным контентом.

Голосовой помощник “Алиса” от компании Яндекс не запоминает определенное число фраз при разговоре с пользователем. Состояние диалога Алисы - это текущий диалоговый «сценарий», состоящий из слотов, которые должны быть заполнены для выполнения запроса. Пока Алиса может связать то, что говорит пользователь, с текущим состоянием, она всегда будет поддерживать контекст.
Система персональных рекомендаций у Алисы основана на других сервисах Яндекса, которыми пользуется авторизованный пользователь. Основываясь на истории сервисов “Кинопоиск”, “Яндекс Музыка” и д.р., Алиса подбирает похожие фильмы и музыку для конкретного пользователя. На данный момент у Алисы отсутствует возможность сохранения и анализа пользовательских предпочтений, которые не связаны с сервисами Яндекса.

Голосовой помощник “Alexa” от компании Amazon отслеживает ссылки через несколько раундов разговора. Alexa анализирует семантическое содержание высказываний по категориям домена, намерения и слота. Разные сценарии (погода, поиск, маршрут) могут назначать одни и те же данные (Сан-Франциско) разным слотам (WeatherLocation, City, Town).
Система персональных рекомендаций частично основана на истории покупок в сервисе Amazon, если пользователь в нем авторизован, - Alexa будет предлагать товары, похожие на ранее купленные. Также Alexa умеет отслеживать повторяющиеся пользовательские действия и предлагать на их основе свои рекомендации.

Голосовой помощник “Google Assistant” использует технологию, которая позволяет обрабатывать контекст путем перефразирования. Технология перефразирует

следующий запрос пользователя, с учетом контекста предыдущих запросов, так что отсутствующая контекстная информация включается в переформулировку в конце.
Система персональных рекомендаций Google Assistant основана на истории поиска в Google, предыдущих запросах Google Assistant и местоположении. Система может предложить рецепт, ресторан, музыку, опираясь на то, что пользователь искал ранее.

В предыдущих абзацах были рассмотрены виртуальные ассистенты, способные предоставить общую информацию по практически любым вопросам из разных сфер, но в данной работе планируется создать цифрового ассистента, специализирующегося в финансовой области.

Сервис Банки.ру представляет собой одно из существующих решений в финансовой сфере, ориентированное на российский рынок. На этой платформе можно найти различные финансовые инструменты и калькуляторы, которые помогают пользователям выбирать и оформлять финансовые продукты в соответствии с их индивидуальными параметрами.
Однако, несмотря на все преимущества платформы, она не предоставляет пользователям рекомендации, основанные на их финансовом положении и целях. Перед тем как выбрать на платформе конкретный финансовый продукт, пользователю необходимо самостоятельно разобраться, для каких целей он нужен, как с ним работать, а также самостоятельно оценить все существующие риски таким образом, чтобы не потерять свои деньги.

Рассмотрев существующие решения, предлагается разработать цифрового ассистента, который будет предоставлять пользователям рекомендации по управлению своим бюджетом на основе введенных данных, таких как доходы, расходы, финансовые цели и другие параметры. Ассистент будет предоставлять актуальную информацию о финансовых инструментах, доступных в различных банках, и предлагать финансовые стратегии, наилучшим образом соответствующие конкретным потребностям пользователя.

Основное преимущество приложения заключается в том, что на российском рынке пока не было представлено решения, которое предоставляло бы рекомендации по нескольким подходящим финансовым продуктам, которые, используясь вместе, могли бы помочь улучшить пользователю его финансовое положение. Пользователи смогут получить всю необходимую информацию о страховых компаниях, пенсионных фондах, налоговой службе, банках и брокерах в одном месте. Кроме того, рекомендательная система поможет пользователям начать изучение и использование новых финансовых инструментов с целью сохранения и увеличения своих сбережений.

Планируется, что платформа будет содержать информацию о следующих финансовых продуктах:

1. Информация от страховых компаний:
- о добровольном медицинском страховании
- о страховании жизни, здоровья
- наилучшие условия страхования ОСАГО согласно введённых параметров (марка авто, год выпуска и др.)

2. Информация от пенсионного фонда:
- о стаже
- предварительный расчет размера выплаты за счет средств пенсионных накоплений

3. Информация от ФНС:
- о налоговых вычетах
- о налоговых льготах
- о налоговых обязательствах

4. Информация от банков:
- о лучшей процентной ставке депозита согласно внесенным критериям (сумма, срок)
- предварительный расчет размера выплаты за кредит (ипотечный, потребительский)
- информирование о программах лояльности

5. Информация от брокеров:
- о доступных инвестиционных продуктах согласно риск-профиля лица (госбумаги, корпоративные облигации, акции и др.)
- о налоговых льготах и вычетах для физических лиц (долгосрочное владение акциями, ИИС и др.)

Литература

[1] Клюев, Леонид Как устроена Алиса. Лекция Яндекса / Леонид Клюев. — Текст : электронный//Habr: [сайт]. — URL: https://habr.com/ru/company/yandex/blog/349372/

[2] Алиса. Как Яндекс учит искусственный интеллект разговаривать с людьми. — Текст:электронный//Habr:[сайт].—URL: https://habr.com/ru/company/yandex/blog/339638/

[3] Help & Customer Service. — Текст: электронный//Amazon : [сайт]. — URL: https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230

[4] Как Яндекс применил генеративные нейросети для поиска ответов. — Текст : электронный//Habr: [сайт]. — URL: https://habr.com/ru/company/yandex/blog/561924/

[5] Learn how Google improves speech models. — Текст : электронный // Google : [сайт]. —URL: https://support.google.com/assistant/answer/11140942?hl=en&ref_topic=7110546#zippy=%2Cfederated-learning

[6] Ивтушок, Елизавета Как устроена голосовая помощница Яндекса и чему ей еще предстоит научиться / Елизавета Ивтушок. — Текст : электронный // N + 1 : [сайт]. — URL: https://nplus1.ru/material/2018/02/27/yandex-alice

[7] Sarikaya, Ruhi The role of context in redefining human-computer interaction / Ruhi Sarikaya. — Текст: электронный//Amazon Science : [сайт]. — URL:

https://www.amazon.science/blog/the-role-of-context-in-redefining-human-computer-interaction

[8] Prashan, Wanigasekara Multimodal context carryover / Wanigasekara Prashan. — Текст :электронный//Amazon Science : [сайт]. — URL: https://www.amazon.science/publications/multimodal-context-carryover

[9] Arpit, Gupta How Alexa is learning to converse more naturally / Gupta Arpit. — Текст :электронный//Amazon Science: [сайт]. — URL: https://www.amazon.science/blog/how-alexa-is-learning-to-converse-more-naturally

[10] Aurelien, Boffy Contextual Rephrasing in Google Assistant / Boffy Aurelien. — Текст : электронный//Google : [сайт]. — URL: https://ai.googleblog.com/2022/05/contextual-rephrasing-in-google.html?m=1

[11] Хранение состояния. — Текст : электронный//Яндекс : [сайт]. — URL: https://yandex.ru/dev/dialogs/alice/doc/session-persistence.html

[12] Request Body Parameters. — Текст : электронный // Developer Amazon : [сайт]. — URL: https://developer.amazon.com/en-US/docs/alexa/custom-skills/request-and-response-json-reference.html#request-body-parameters

[13] Крылов А. В. Проблема извлечения знаний с использованием рассуждений на основе прецедентов // Изв. вузов. Приборостроение. 2018. Т. 61, № 11. С. 956—962

Разработка архитектуры децентрализованного приложения на основе блокчейна NEO

Черновская Я.Т.

y.chernovskaya@g.nsu.ru

Новосибирский государственный университет, ул. Пирогова, 1, г. Новосибирск, 630090, Россия

Ключевые слова: блокчейн, децентрализация, метавселенная, смарт-контракт, десктопное приложение.

Современное информационное общество переживает уникальное развитие в сфере цифровых технологий. Одним из самых важных и актуальных направлений этой эволюции являются метавселенные – виртуальные миры и экосистемы, предоставляющие пользователям возможность взаимодействовать, творить и развиваться в цифровом пространстве. Однако, несмотря на впечатляющие достижения в этой области, существует одно существенное ограничение: большинство метавселенных принадлежат отдельным корпорациям и организациям, что ограничивает доступность и независимость пользователей.

Целью работы является разработка проекта, представляющего децентрализованное приложение на основе технологии блокчейн. Приложение не только дает возможность пользователям создавать и управлять своими виртуальными мирами и персонажами, но и устанавливает новый стандарт прозрачности, независимости и безопасности в сфере метавселенных.

Для решения задачи поддержания актуальных данных в самой блокчейн сети и синхронизации данных с игрой был написан модуль с обертками над смарт-контрактами. Каждое изменение карты игроком фиксируется приложением на языке программировании Си и отправляется в очередь, находящейся в модуле, написанном на языке Java. Этот модуль отслеживает состояние очереди, и при необходимости делает запрос модулю со смарт контрактами для внесений всех изменений в блокчейн сеть. Также одной из задач было написание графической составляющей лаунчера - загрузчика игровой части, который также соединяет пользователя с сетью. Поскольку реализация модуля взаимодействия с блокчейн сетью и самой игрой была написана на языке программирования Java, то оптимальным решением стало использование платформы JavaFX (набор инструментов для создания кроссплатформенных приложений на основе Java), отличающейся простым написанием кода и легковесностью графической составляющей.

Благодарности

Работа выполнена при поддержке АО "СберТех" и ПАО "СБЕРБАНК" на базе студенческой учебно-научной лаборатории факультета информационных технологий Новосибирского государственного университета.

Литература

[1] Tharaka Hewa, Mika Ylianttila, Madhusanka Liyanage, Survey on blockchain based smart contracts: Applications, opportunities and challenges, Journal of Network and Computer Applications, Volume 177, 2021.

Построение онтологии продукта для реализации поисковой системы

Р.С. Чурилова

Институт математики им. С.Л. Соболева СО РАН, г. Новосибирск

renata.tchurilowa@yandex.ru

Аннотация. *Поисковые системы играют важную роль в определении успеха электронной коммерции (e-commerce). Несмотря на то, что было предпринято много усилий для улучшения методов поиска (Search Methods (SM)), они до сих в основном ограничиваются семантическим подбором ключевых слов. Это означает что SM не в состоянии поддерживать исследования продуктов, которые лучше всего удовлетворяют клиентов, в соответствии с их пользовательскими характеристиками. Чтобы решить эту проблему, в статье представлен подход способный определить новую онтологическую модель, которая формализует знания, необходимые для реализации поисковой системы, способной направлять клиента к поиску желаемого продукта или услуги в соответствии с его характеристиками и потребностями запроса. система интеллектуального поиска для интернет-магазина, основанная на формировании онтологии товаров. Система предполагает обработку и расширение пользовательского запроса. В статье описывается структура онтологии в формате OWL.*

Ключевые слова: базы знаний, интеллектуальный поиск, онтология продукта, электронная коммерция

1 Введение

Термин «информационный поиск» впервые введён в научный оборот американским математиком Кельвином Муэрсом в 1951 году. В ГОСТе 7.73-96 информационный поиск – это действия, методы и процедуры, позволяющие осуществлять отбор определенной информации из массива данных. Проблема информационного поиска может быть выделена в работе интернет-магазина. Она заключается в том, что пользователи не могут быстро найти необходимый товар и покидают интернет-магазин. Большинство систем сегодня базируется на булевой модели поиска. Логические переменные получают значение «истина» или «ложь» в зависимости от вхождения или не вхождения терминов запроса в соответствующий документ. Рост всемирной паутины выявил недостаточность существующих в настоящее время методов поиска и индексации данных. Информационный поиск может быт основан на семантический структуре. Семантическая сеть позволяет создавать между терминами связи, придавая информации значение и помогая интегрировать новые скрытые знания. Если с помощью онтологии можно формализовать любую область знаний, то есть возможность применения онтологии для электронной коммерции.

Онтология играет ключевую роль в необходимости структурирования большого объема информации. С ее помощью клиенты могут находить нужные товары и принимать решения о покупке. Существует возможность применения в системе поиска рекомендаций, взаимодействия с пользователем. Пользователи чаще всего знают, что ищут, но не знают, как сформулировать поисковый запрос. Самый простой способ – описать ключевые характеристики или метаданные искомого объекта. В связи с этим возникает потребность в создании интеллектуальной системы поиска. Если представить товары, категории, характеристики и связи между ними в виде семантической сети и использовать машину логического вывода для формирования результата поискового запроса, то качественные показатели поисковой системы изменятся.

Однако успех электронной коммерции тесно связан с опытом клиентов, связанным с процессом покупки продукта, начиная с поиска продукта (услуги). Соответственно, поисковые системы играют важную роль в этом контексте, поскольку они представляют собой мощный и простой инструмент, облегчающий покупку продуктов и помогающий пользователям найти то, что они хотят. Было предприняты попытки по улучшению методов и процессов поиска, основанных на связывании веб-сайтов с ключевыми словами, так что поисковая система действительно достигла хорошего уровня эффективности. Однако, поскольку они работают посредством семантической обработки ключевых слов, введенных пользователем, или в соответствии с определенными критериями поиска, заданными пользователем (например, поисковый фильтр), они не позволяют находить продукты, выходящие за пределы сферы знаний пользователя.

Для повышения эффективности поисковой системы используются другие методы отслеживания действий пользователя вовремя просмотра с целью сбора информации для идентификации пользователя. Среди них наиболее распространены профилирующие файлы («profiling cookies»), основанные на использовании постоянных файлов для отслеживания активности пользователя в Интернете. Информация, собранная о предпочтениях пользователей, используется для улучшения поисковой системы, предлагая другое содержимое (например, продукты, темы и т.д.) Однако массовое использование этих методов может считаться наносящим ущерб конфиденциальности и может негативно сказаться на опыте пользователей в Интернете. Более того, они не позволяют пользователю найти продукты или услуги, о которых он еще не знает или которые он никогда не искал. Кроме того, потребитель часто просто имеет потребность и не знает какой товар или услуга могут его удовлетворить. Возможным решением этой проблемы могло бы стать внедрение поисковой системы, способной направлять клиента в поиске желаемого товара или услуги в соответствии с его характеристиками и потребностями. Однако для достижения этой цели необходима стратегия управления несколькими областями знаний.

Таким образом, можно заметить, что эти проблемы могут быть решены с использованием семантических поисковых систем, которые облегчают работу пользователя, эффективны в поиске, поскольку они находят результаты на основе контента, предоставляя более точную информацию о том, что ищется, предлагая более релевантное количество результатов, облегчающее работу по фильтрации результатов пользователем. Поисковые системы интерпретируют поисковые запросы пользователей, используя алгоритмы, которые символизируют понимание, быстро предлагая точные результаты и в конечном итоге распознавая правильный контекст для искомых слов или предложений. Это не что иное, как семантическая поисковая система, которая выполняет поиск, рассматривая значение группы написанных слов.

2 Теоретические проблемы

2.1 Постановка задачи

Семантическая паутина заключается в том, чтобы данные в Сети были определены и связаны таким образом, чтобы их можно было более эффективно использовать для обнаружения, автоматизации, интеграции и повторного использования между различными приложениями. Задача семантической сети состоит в том, чтобы предложить язык, который выражает данные и правила для рассуждений о многих данных, а также позволяет экспортировать правила любой системы представления знаний в Сеть, обеспечивая степень гибкости и «свежести» традиционным централизованным системам представления знаний, которые становятся чрезвычайно громоздкими, и его увеличение в размерах неуправляемо. Разные веб-системы могут использовать разные идентификаторы для одного и того же понятия. Таким образом программа, которая хочет сравнить или объединить информацию между такими системами должна знать, какие термины означают одно и то же. идеале у программы должен быть способ обнаружения общих значений любой базы данных, с которой она сталкивается. Одним из решений этой проблемы является добавление онтологий в семантическую сеть.

Разработка онтологий имеет отношение к поисковой оптимизации (SEO), поскольку она может помочь улучшить способ, которым поисковые системы понимают и индексируют контент на веб-сайте. Когда поисковая система сканирует веб-сайт, она пытается понять содержание и контекст каждой страницы, чтобы определить насколько она релевантна заданному поисковому запросу. Благодаря использованию онтологий для обеспечения четкого и структурированного представления концепций и взаимосвязей в предметной области поисковым системам становится легче понять содержимое веб-сайта и то, как оно соотносится с заданным поисковым запросом.

По своей сути SEO представляет собой набор методов и стратегий, направленных на повышение релевантности и точности результатов поисковых систем. Это реализуется с

помощью семантической разметки, такой как встраивание слов и обработка естественного языка, чтобы лучше понять значение и содержание на данном веб-сайте. Таким образом, поисковые система могут возвращать более релевантные результаты для данного запроса, что может помочь улучшить взаимодействие с пользователем и привлечь больше клиентов на веб-сайты, использующие семантические методы SEO.

Одним из ключевых инструментов, используемых в SEO, является микроразметка. Это тип кода, который добавляет HTML-код веб-сайта для предоставления дополнительной информации о содержимом сайта. Например, разметку схемы можно использовать для указания типа контента на странице (например, статьи), а также его релевантности определенным ключевым словам или темам.

Использование методов семантического SEO имеет несколько ключевых преимуществ:

1. Улучшение результата поиска. Используя семантическую разметку и обработку естественного языка для лучшего понимания смысла и содержания на данном веб-сайте, поисковые системы могут возвращать более релевантные и точные результаты для данного запроса.
2. Улучшение взаимодействия с пользователем. Поскольку семантическое SEO помогает поисковым системам выдавать более релевантные результаты, оно также может улучшить взаимодействие с пользователем. Например, если кто-то ищет конкретный продукт, семантическое SEO может помочь убедиться, что ему будут показаны наиболее релевантные результаты, а не список несвязанных продуктов, которые содержат ключевое слово, которое они искали.
3. Повышение видимости веб-сайта. Если веб-сайт использует семантическую разметку, чтобы указать, что он охватывает определенную тему или продукт, он с большей вероятностью будет отображаться в результатах поиска по этой теме или продукту.
4. Лучшее ранжирование в поисковых системах. Повышение видимости веб-сайта может помочь ему занять более высокое место в результатах поиска и улучшить его общую видимость.
5. Усовершенствование оптимизация контента. Семантическая поисковая оптимизация может помочь улучшить организацию и структуру содержимого веб-сайта, сделав его более удобным для пользователя и понятным для поисковых систем.

Хотя семантическое SEO предлагает множество преимуществ, оно также сопряжено с рядом проблем и ограничений:

1. Сложность. Семантическое SEO включает в себя использование ряда методов и технологий, включая семантическую разметку, обработку естественного языка и алгоритмы для анализа и понимания контента на веб-сайте.
2. Ограниченное внедрение. Не все поисковые системы способны полностью понимать и использовать методы семантической разметки и обработки естественного языка, используемые в семантической поисковой оптимизации, что может ограничивать ее эффективность.
3. Ограниченный охват. Семантическое SEO не единственный фактор, определяющий успех веб сайта. Например, дизайн веб-сайта, взаимодействие с пользователем и маркетинговые усилия также могут оказать значительное влияние на его видимость и посещаемость. Поэтому следует рассматривать SEO как часть более широкой стратегии поисковой оптимизации, а не как отдельное решение.
4. Изменение алгоритмов. Поисковые системы постоянно обновляются и совершенствуют свои алгоритмы, это означает, что методы и стратегии, используемые в семантическом SEO, могут со временем устареть. Владельцы веб-сайтов должны знать об этих изменениях и соответствующим образом адаптировать свои семантические стратегии SEO.

Несмотря на проблемы и ограничения SEO, это быстрорастущая область, которая продолжит развиваться в ближайшие годы.

2.2 Система интеллектуального поиска

Онтологии можно использовать для поддержки более сложных функций поиска, таких как семантический поиск и обработка естественного языка. Семантический поиск включает в себя понимания значения и контекста поискового запроса для получения более релевантных результатов. Используя онтологию для понимания значения и контекста поискового запроса, поисковая система может более точно сопоставить запрос с соответствующими документами. Точно так же обработка естественного языка включает в себя понимание значения и контекста текста на естественном языке для излечения соответствующей информации, часто называемой

распознаванием именованных объектов. А также поисковая система может более точно извлекать релевантную информацию из документов.

Онтология – это формальное представление набора понятий в предметной области и отношений между этими понятиями. Онтологии обычно используются в искусственном интеллекте для поисковой оптимизации обработки естественного языка и представления знаний, чтобы компьютеры могли понимать и рассуждать о данных.

Семантическая тройка, также известная как тройка субъект-предикат-объект или тройка, представляет собой представление части данных в виде тройки, состоящей из субъекта, сказуемого и объекта. Субъект – это описываемая сущность или понятие, предикат – это свойство или отношение, утверждаемое о субъекте, а объект – это значение свойства или сущности, связанной с субъектом посредством предиката. Семантические тройки используются для представления данных таким образом, чтобы они были значимы как для людей, так и для компьютеров, позволяя им понимать и рассуждать о данных. Онтологии и семантические тройки тесно связаны меду собой и часто используются вместе для структурированного представления и организации данных. Онтологии обеспечивают основу для организации понятий и отношений внутри предметной области, а семантические тройки обеспечивают способ представления отдельных фрагментов данных в этой структуре.

2.3 Создание онтологии предметной области в Protege

Мобильные устройства обладают свойством быстрого обновления и каждый день создается все больше мобильных устройств, поэтому объем знаний о мобильных устройствах постоянно расширяется, что доставляет в первую очередь большие проблемы при поиске товаров и услуг (например, ремонтных) обычным покупателям. Современные электронные каталоги (ЭК) смарт-товаров переполнены различными параметрами, с которыми пользователю не всегда легко разобраться. Классификация некоторых мобильных функций в ЭК индивидуальна, одним и тем же атрибутам присваиваются разные названия. База мобильной онтологии поможет систематизировать и формализовать знания о мобильных устройствах, чтобы сделать знания доступными для совместного использования, как пользователя, так и компьютера. Потребить мобильных устройств сможет легко найти свои предпочтения по характеристикам поиска, основанном на базе онтологии, а веб-сайты обеспечиваются мощной поддержкой семантического поиска.

Перед началом работы с системой Protégé необходимо составить концептуальную модель предметной области. И для будущей ее формализации необходимо определить какими сущностями можно оперировать. В данном случае это сущности модели RDF и языка OWL, т.к. многие системы и библиотеки, в том числе Protégé поддерживает эти стандарты, и они являются наиболее распространенными.

Следует отметить, что модель RDF является абстрактной и существует множество языков с разными синтаксисами, реализующих ее. Онтология в рамках модели RDF предполагает определение любых понятий предметной области в виде троек: субъект – предикат – объект. Множество таких троек образует граф понятий. Субъекты и объекты интерпретируются как вершины ориентированного графа, а предикаты как его дуги (и определяют отношения между сущностями).

Каждый класс является подклассом owl:Thing и типом rdf:Class. Классы могут быть равны или не пересекаться друг с другом, а также образовывать иерархическую структуру, используя свойство объекта subClassOf. Каждый класс может иметь дополнительное описание.

Каждое свойство объекта является вспомогательным свойством owl:topObjectProperty и имеет тип rdf:ObjectProperty. Свойства объекта могут иметь характеристики, которые определяют больше правил. Эти характеристики показаны в таблице 1. Экземпляры A и B, связанные свойством, обозначаются как $A \rightarrow_p B$. Свойства объекта также могут иметь определение области определения и значения, обозначаемые в программе domain и range соответственно.

Таблица 1. Характеристики свойств объекта.

Наименование	Описание
Functional	Для индивидуального ввода есть только один вывод
Inverse functional	Индивидуальный выходной параметр может быть связан только с одним транзитивным свойством
Transitive	Транзитивное свойство – $A \rightarrow_p B \cap B \rightarrow_p C \Longrightarrow A \rightarrow_p C$
Symmetric	Свойство и его инверсия совпадают
Asymmetric	Если $A \rightarrow_p B$, то $B \rightarrow_p A$ не может существовать
Reflexive	Каждый элемент может быть связан сам с собой
Irreflexive	Ни один экземпляр не может быть связан сам с собой

Каждое свойство данных является подствойством owl:topDataProperty и типа rdf:DatatypeProperty, для которых также могут быть определены domain и range. Свойства данных являются буквальным значением, область определения берется из набора классов, а область значения может быть значением строки определенного типа xsd:string или xsd:double.

Онтология продукта началась с анализа характеристик смарт-объекта (SO). Нами была собрана информация, относящаяся к общим стандартным данным для различных электронных товаров (например, производительность, цена, бренд, модель и т.д.) Впоследствии произведен анализ для выявления специфики смарт-продуктов. База мобильной онтологии построена на существующих технологических свойствах мобильного телефона. Онтология SO разделена на классы: модель, беспроводные интерфейсы, бренд, операционная система (ОС), версия ОС, диагональ экрана, количество SIM, объемы встроенной и оперативной памяти, продавец, процессор, разрешение экрана, скидка, спутниковая навигация и т.д. В основном эти понятия формируют атрибуты, например, hasbrand означает взаимосвязь между смартфоном и брендом, hascolor взаимосвязь между смартфоном и цветом, hasmodel означает взаимосвязь между моделью и брендом.

Кроме того, были учтены все характеристики основных компонентов, такие как расширение экрана, объем памяти, частота процессора, встроенные датчики (акселерометр, гироскоп, уровень и др.). Также взяты во внимание технические характеристики, относящиеся к измеряемым параметрам: разрешающая способность, единица измерения, точность и т.д. Класс единицы измерения собирает стандарт для измерения величины, которая может быть связана с параметрами, а также со спецификациями. Кроме того, был определен набор сервисов в соответствии с основной функцией существующих SO, чтобы учитывать потребности пользователей (например, возможность контролировать, улучшать, мониторить или управлять некоторыми аспектами повседневной жизни в офисе, дома, на улице т.п.). На рисунке 1 показана частичное представление онтологии продукта. Некоторые классы не были расширены, чтобы не влиять на удобочитаемость.

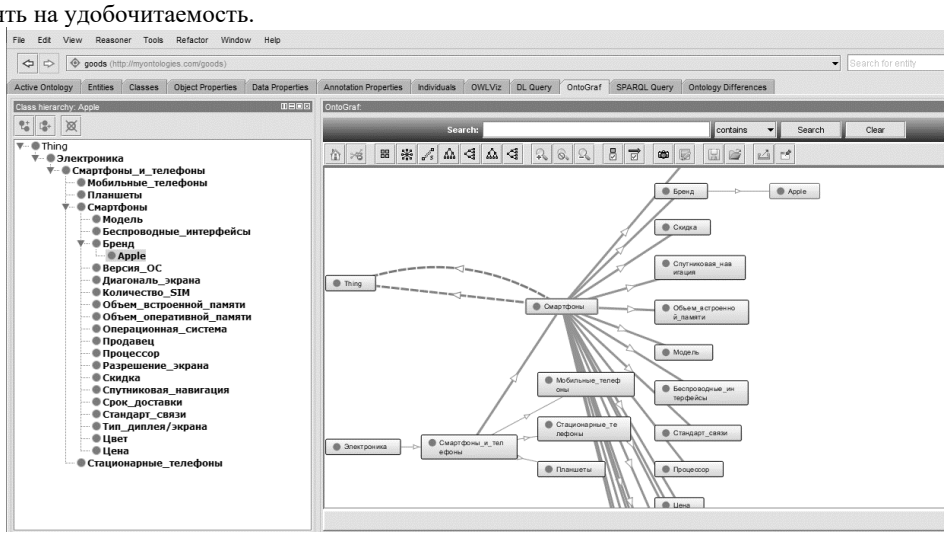


Рис. 1 Онтология мобильного продукта в Protégé.

3 Заключение и будущие работы

Использование позволяет значительно улучшить понимание контекста поисковых запросов и адаптировать SEO-стратегии под изменяющиеся потребности пользователей. Это открывает новые возможности для продвижения сайтов и улучшения их видимости в поисковых системах.

Эта работа представляет собой описание разработки онтологической модели в системе Protégé, которая может быть использована для создания ориентированной на пользователя поисковой системы, способной предлагать клиентам результаты, которые наилучшим образом соответствуют их потребностям и целям, улучшая общее впечатление от поисковой активности. Будущие работы будут включать доказательство осуществимости и превосходства модели онтологии предметной области методом эксперимента, глубокую интеграцию с поисковой системой и проверкой пользователя в веб-приложении.

Чтобы удовлетворить требования достаточно большого объема запросов, поисковые системы должны быть персонализированы к доступу информации. Это связано с тем, что в реальной жизни у разных пользователей поисковых систем могут быть разные предпочтения в

поиске. Отсюда возникает потребность в создании онтологии пользовательского контекста и возникает вопрос какие характеристики пользователя являются важными для составления персонализированного поиска. Кроме того, следует понять связь между пользовательскими онтологиями и онтологиями продуктов, чтобы повысить удовлетворенность потребителей поиском, сравнительным анализом и покупками при просмотрах веб страниц. Будущая исследовательская работа направлена на решение этих проблем.

Литература

[1] Цуканова Н. И. Онтологическая модель представления и организации знаний: учебное пособие для вузов. − М.: Изд-во Горячая линия – Телеком, 2015.– С. 272.

[2] Басипов А. А., Демич О. В. Семантический поиск: проблемы и технологии // Вестн. Астрах. гос. техн. ун-та. − 2012. − №1. − С. 8.

[3] Пальчунов Д. Е. Решение задачи поиска на основе: бизнес информатика // Архитектура предприятий. − 2008. − №1. − С. 11.

[4] Денисов М.П. Лабораторные работы по описанию предметных областей с использованием системы Protégé / М.П. Денисов. – Казань: Казан. ун-т, 2020. – 53 с.

[5] Sukanta Sinha, Rana Dattagupta, Debajyoti Mukhopadhyay “Designing an Ontology based Domain Specific Web Search Engine for Commonly used Products using RDF”, CUBE 2012 International IT Conference, CUBE 2012 Proceedings, Pune, India; ACM Digital Library, USA; September 3-5, 2012; pp.612-617; ISBN 978-1-4503-1185-4.

[6] Gulla F, Cavalieri L, Ceccacci S, Papetti A, Germani M. The user-product ontology: A new approach to define an ontological model to manage product searching based on user needs. In: International Conference on Human Interface and the Management of Information (HIMI 2017). 9-14 July 2017; Vancouver. Cham: Springer; 2017. pp. 333-346. DOI: 10.1007/978-3-319-58521-5_27

Разработка алгоритма замены упражнений в йога-тренировках

Шатрова Анна Игоревна

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия
Novosibirsk State University, 2 Pirogova st., Novosibirsk, Russia

a.shatrova@g.nsu.ru

В современном мире все больше все больше людей осознанно подходят к своему здоровью: правильно питаются, следят за режимом сна и занимаются физической активностью. Раньше люди стремились в тренажёрные залы, где занимались довольно изнурительными тренировками. С распространением Интернета и различных технических средств у большего количества людей появилась возможность заниматься спортом в домашних условиях. Наибольшую распространенность это приобрело во время пандемии, но и сейчас многие не хотят отказываться от возможности тренироваться в любом удобном месте. Занятия йогой же с каждым днем становятся все более популярными из-за осознанного подхода к физическим упражнениям, к своему телу. Однако все тренировке по йоге представлены в готовом виде и простому обывателя трудно адаптировать их без участия тренера. Для решения этой проблемы предлагается разработать алгоритм замены упражнений в йога-тренировках.

Чтобы разработать такой алгоритм, необходимо определить способ, с помощью которого будет осуществляться замена упражнений. Было принято решение реализовать классификацию упражнений на основе ключевых слов [1]. Для этого необходима база данных йога-упражнений, в которой содержатся названия и описания упражнений. Все описания упражнений приведены в начальную форму, после этого выполняется подсчет частоты встречаемости каждого слова без учета стоп-слов, которыми являются предлоги, союзы и слова, которые часто употребляются, но не имеют смыслового значения [2]. С помощью метода кумулятивной суммы составляется словарь ключевых слов, на основе которых будет осуществляться классификация упражнений. После этого составляется вектор для каждого ключевого слова. У каждого упражнения есть набор ключевых слов и по соответствию их векторов будет выполняться замена упражнений в йога-тренировках.

Литература

[1] Yakhyaeva G.E. Application of Boolean Valued and Fuzzy Model Theory for Knowledge Base Development \\ SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings, 2019, стр. 868–871.
[2] Ignatow G., Mihalcea, R. \\ An introduction to text mining. SAGE Publications, 2018.

Знания-Онтологии-Теории (ЗОНТ-2023)

Использование психографики для создания рекомендательной системы компьютерных игр

Шелдяков Иван Павлович

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия
Novosibirsk State University, 2 Pirogova st., Novosibirsk, Russia

i.sheldyakov@g.nsu.ru

Компьютерные игры набирают популярность с каждым годом, и их ассортимент расширяется. Учитывая, что время для современного пользователя является драгоценным ресурсом, возникает необходимость в быстром и эффективном выборе игры, соответствующей его вкусам, чтобы избежать ненужных затрат. Один из подходов к решению этой проблемы – разработка рекомендательных систем на основе семантического анализа предметной области [1, 2].

Данное исследование направлено на создание рекомендательной системы, которая бы учитывала психографические особенности пользователя. Психографика предоставляет инсайты о внутренних предпочтениях игрока, позволяя системе предлагать игры в соответствии с желаемым типом взаимодействия и удовольствия [3].

Для определения психографического профиля игрока применяется специализированный опросник. А для определения типа игры используется программа, анализирующая текстовое описание игры и отзывы критиков, выделяя ключевые слова, которые указывают на конкретный тип удовольствия. Эти данные преобразуются в вектор, представляющий каждую игру. Аналогичный вектор создается для игрока на основе его ответов в опроснике. Сравнив вектора, система формирует рекомендации для пользователя.

Первоначально ключевые слова, связанные с типами удовольствия, формируются вручную. Но со временем система автоматически обновляет этот список, основываясь на частоте их упоминания в описаниях игр. Если слово появляется около определенного предельного значения, оно добавляется в список ключевых слов. В качестве ключевых слов выбираются прилагательные и существительные.

Литература

[1] Yakhyaeva G., Skokova V. Subjective Expert Evaluations in the Model-Theoretic Representation of Object Domain Knowledge. Lecture Notes in Computer Science, 2021, 12948 LNAI, стр. 152–165.

[2] Yakhyaeva G.E. Application of Boolean Valued and Fuzzy Model Theory for Knowledge Base Development \\ SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings, 2019, стр. 868–871.

[3] Шелл Джесси Геймдизайн. Как создать игру, в которую будут играть все // “Альпина Паблишер”, Москва, 2020.

Исследование влияния методов прунинга нейронных сетей на их интерпретируемость

Щербин Андрей Сергеевич

Новосибирский государственный университет, ул. Пирогова 2, Новосибирск, Россия

a.shsherbin@g.nsu.ru

Искусственные нейронные сети используются в критически важных областях, таких как медицина и автономные автомобили. В этих областях важно понимать какие факторы повлияли на принятие нейронной сетью того или иного решения. Понимание причин принятия решений позволяет улучшать качество работы нейронных сетей, а также находить скрытые закономерности в предметной области [1, 2]. Благодаря методам интерпретируемости повышается доверие пользователей к алгоритмам, основанным на нейронных сетях, и расширяется граница их применимости.

Не менее важным направлением исследований является оптимизация нейронных сетей. В индустрии часто возникают задачи запуска нейросетевых алгоритмов на маломощных устройствах: видеокамерах, умных колонках, смартфонах. Для решения таких задач применяется структурированный прунинг - удаление из нейронной сети весов, не важных для решения конкретной задачи. В этом случае архитектура нейронной сети становится более специализированной для решения конкретной задачи, а вычислительная сложность сети уменьшается.

Целью данной работы является исследование влияния алгоритма прунинга нейронной сети на ее интерпретируемость. В рамках исследования был проведен вычислительный эксперимент над нейронной сетью архитектуры ResNet50 и набором данных ImageNet2012. В качестве алгоритмов использовались прунинг по Тейлору и прунинг по L2 норме весов. Оптимизация производилась в 2 раза по количеству сложений и умножений (MAC). В качестве метрики интерпретируемости использовалась Integrated gradients [3]. Важными преимуществами данной метрики является универсальность реализации и чувствительность. Эксперимент показал, что просадка интерпретируемости для модели, оптимизированной по Тейлору меньше, чем для модели, оптимизированной по L2.

В дальнейшем планируется проверить полученный результат для других моделей, других метрик интерпретируемости, а также других методов оптимизации, в частности квантования.

Литература

[1] Ibrahim M. A Global Explanations of Neural Network: Mapping the Landscape of Predictions / M. Ibrahim, M. Louie, C. Modarres, J. Paisley //AIES '19: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 279–287.

[2] Shcherbin A., Yakhyaeva G. Gaussian Based Active Learning Algorithm for Image Classification Problem. International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices, EDMthis link is disabled, 2021, 2021-June, pp. 542–546.

[3] Sundararajan M. Axiomatic Attribution for Deep Networks / M. Sundararajan, A. Taly, Q. Yan // Proceedings of the 34 th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017, pp. 3319–3328.

Знания-Онтологии-Теории (ЗОНТ-2023)

Автоматизированные методы ведения аргументативного диалога с пользователем

Якобсон Александр Алексеевич

Новосибирский государственный университет, г. Новосибирск, 630090, Россия.

a.yakobson@g.nsu.ru

Аннотация. *В работе представлено описание архитектуры интеллектуального помощника, использующего прецедентный подход для ведения аргументативного диалога с пользователем. Архитектура системы представляет собой набор модулей для формализации текстового ввода пользователя, трехуровневой семантической модели пользователя, модуль метрик сходства, определяющие схожесть прецедентов, а также базу прецедентов, организованную по принципу семантического сходства прецедентов.*

Ключевые слова: интеллектуальный помощник, аргументативный диалог, прецедент, автоматизация ведения диалога

В настоящее время в сфере оказания консультационных услуг, таких как банки и интернет-магазины, используются виртуальные ассистенты, предназначенные для помощи пользователю в поиске подходящего решения или товара. Однако, зачастую, эти системы работают по заранее определенному скрипту и при возникновении новой или не предусмотренной скриптом операции переводят клиента на звонок консультанта-человека. Возникает ситуация, в которой виртуальный ассистент не способен найти решение или выдать нужную рекомендацию, что приводит к снижению мотивации пользователя использовать его в дальнейшем. Кроме того, подобные системы практически всегда ведут диалог «С нуля», не запоминая пользователя и контекст диалога.

Разработанная система автоматизирует момент поиска решения за счет использования накопленных ранее случаев (прецедентов). За счет сравнения цели текущего пользователя и данных из предыдущих диалогов возможно не только найти похожее решение в прошлом, но и дополнительно аргументировать предложенное решение на основе совпадения намерений. Это дает системе преимущество перед существующими аналогами, которые не способны к накоплению прецедентов и аргументации своих решений.

Разработанная система является развитием системы, описанной в работе [1]. В текущем состоянии, состоит из следующих компонентов.

1. Блок ведения диалога с пользователем на естественном языке в текстовом формате. Сообщения пользователя преобразуются в формальный вид – конструкции из речевых действий и терминов предметной области [2].
2. Хранилище данных о пользователе системы в виде трёхуровневой модели: общие знания, представление о текущей ситуации, намерения пользователя. Модель интегрирована с предметной областью и адаптирована под нее.
3. Модель предметной области представлена в виде набора понятий и семантических связей между ними. Структура базы знаний поддерживает настройку под конкретную предметную область, содержит в себе систему самонастройки, в качестве входных данных для настройки конфигуратор принимает набор понятий и связей между ними. Онтология предметной области использует структуру, описанную в работе, Онтология предметной области использует структуру, описанную в работе [3].

4. База прецедентов хранит в себе объекты вида «Прецедент»: намерения пользователя, список шагов системы (вариации решения), ответ пользователя, прецедент, использовавшийся при построении решения или пометка о генерации прецедента с нуля.

5. Для определения схожести прецедентов используется универсальная метрика на основе оценки и сравнения параметров объектов. Метрика представляет собой алгоритм оценки, основанный на нейросетевом и/или логическом подходе [4].

6. Блок анализа новых/неточно определенных понятий. В задачи блока входит анализ понятий, не обнаруженных в предметной области и дополнение/обновление предметной области для более корректной работы с этим понятием в дальнейшем.

Таким образом, разработанная система способна к ведению аргументативного диалога с пользователем, анализу и накоплению прецедентов для улучшения работы системы. Применяемый подход позволяет генерировать аргументы для подкрепления полученного решения перед пользователем. Явная аргументация со ссылкой на прецедент позволяет повысить прозрачность работы системы.

Литература

[1] Palchunov D., Yakobson A. Automated methods for conducting an argumented dialogue with the user //2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). – IEEE, 2022. – С. 880-885.

[2] Махина Е.Д., Пальчунов Д.Е. Программная система для определения речевых действий в текстах естественного языка // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2018. Т. 16. № 4. С. 95–106. DOI 10.25205/1818-7900-2018-16-4-95-106

[3] Найданов Ч. А. Разработка ядра онтологической модели, настраиваемой под предметную область //Вестник Новосибирского государственного университета. Серия: Информационные технологии. – 2019. – Т. 17. – №. 1. – С. 72-81.

[4] Abro W. A. et al. Natural language understanding for argumentative dialogue systems in the opinion building domain //Knowledge-Based Systems. – 2022. – Т. 242. – С. 108318.

Компиляция объектно-ориентированного языка в WebAssembly

Янин А. С.[1]

[1]*Новосибирский государственный университет, ул. Пирогова, д. 1, г. Новосибирск, 630090, Россия.*

a.yanin@g.nsu.ru

Аннотация. *Рассматривается вопрос компиляции объектно-ориентированного языка для современной виртуальной машины WebAssembly. WebAssembly разрабатывается как универсальный байт-код, высокопроизводительный за счёт близкости к машинным инструкциям, безопасный для исполнения, а также удобный для трансляции в него существующих языков. Однако первые две цели — эффективность и безопасность — накладывают ряд ограничений и особенностей, которые усложняют достижение третьей. Для исследования вопроса был разработан компилятор объектно-ориентированного языка Cool в WebAssembly с применением расширения Wasm GC, добавляющий ссылочные структуры и массивы. Большая часть семантики однозначно соответствует каким-то средствам WebAssembly, однако отдельные элементы: асинхронность, обработка ошибок — представляют трудности. На примере компилятора показывается, каким образом транслируются различные конструкции языка, и обсуждаются возникшие сложности.*

Ключевые слова: компиляторы, WebAssembly, байт-код, языки программирования.

1 Введение

WebAssembly — новый стандарт для виртуальной машины, предназначаемой для веб-браузеров[1]. Он позиционируется как универсальный набор инструкций: близкий, с одной стороны, к машинным, но также пригодный, с другой стороны, для трансляции в него программ на уже существующих языках программирования (C, C++, Rust).

Рассматривается разработка компилятора программ на языке Cool в WebAssembly. Cool является минималистичным объектно-ориентированным языком[2]: он имеет статические классы, объекты с полями и методами, единственное наследование с переопределением методов, статическая типизация, статические и виртуальные вызовы. В языке также присутствует особый тип SELF_TYPE, который имеет ссылка на текущий объект self. Таким образом, язык охватывает основную функциональность объектно-ориентированных языков.

2 Компиляция

2.1 Постановка задачи

Характерной чертой байт-кода WebAssembly является структурированность потока исполнения. В отличие от JVM, в WebAssembly отсутствует инструкция перехода по адресу. Вместо этого тело функций представляет собой последовательность вложенных друг в друга блоков конструкций потока управления: if, loop, block; переходы допускаются только изнутри в начало (для loop) или конец (для остальных) охватывающего блока.

Версия 1.0 стандарта допускает только числовые и вещественные значения — в обоих случаях 32- и 64-битные. Так как основные реализации виртуальной машины — браузерные JavaScript-движки, которые также поддерживают работу с объектами, было предложено расширение стандарта, добавляющее ссылочные объекты, управляемые сборщиком мусора[3].

В свете описанных выше особенностей интерес представляет сложность трансляции в WebAssembly программ на современных языках, в частности объектно-ориентированных, как наиболее распространённых. Для изучения этого вопроса был разработан компилятор для языка Cool в WebAssembly.

2.2 Трансляция объектов

Классы в Cool задают соответствующие им типы объектов. Объекты представляются в виде структур (из расширения Wasm GC, добавляющего ссылочные объекты) с объявленными в классе полями. Кроме того, все объекты имеют дополнительное поле со смещением в таблице методов, используемое для виртуальных вызовов.

Наследование реализовано на основе аннотаций подтипа, также предоставляемых Wasm GC. В силу особенности системы типов, расширяемой Wasm GC, объекты двух разных классов имеют различные типы, даже если содержат одинаковые поля (таким образом реализуется номинативность системы типов Cool).

Синтаксически SELF_TYPE может присутствовать только как тип возвращаемого значения метода, поля, локальной переменной, либо в операторе new. Во всех случаях он обозначает динамический тип ссылки на объект self. Так как WebAssembly не имеет такой тип, при трансляции методов и полей SELF_TYPE заменяется на тип самого базового класса, в котором объявлен этот член класса. При объявлении локальной переменной типа SELF_TYPE используется тип класса, в котором пристствует это объявление. Наконец, конструкция new SELF_TYPE транслируется в виртуальный вызов копирующего конструктора.

2.3 Трансляция методов и выражений

Ссылка self передаётся явно первым аргументом методам. WebAssembly не задаёт отношение подтипизации между функциональными типами, поэтому при переопределении метода в подклассе статический тип метода не изменяется. Статический тип добавляемого первого параметра определяется так же, как для поля типа SELF_TYPE.

Все методы группируются в таблицы по WebAssembly-типу. Таблицы хранят ссылки на соответствующие функции; при этом тип их статически известен, поэтому при виртуальных вызовах не требуются динамические проверки. Также составляется сводная таблица методов: в пределах классов методы нумеруются (в порядке наследования), затем для каждого метода (WebAssembly-типа M) каждого класса последовательно записывается позиция соответствующей ссылки на функцию в таблице функций типа M. Отступ в сводной таблице, с которого перечисляются методы класса C, записывается при инициализации объектов класса C.

Виртуальный вызов метода под номером n типа M (оба параметра известны во время компиляции) объекта класса C происходит следующим образом: из поля объекта извлекается отступ p для класса C в сводной таблице методов, запрашивается (p + n)-ый элемент сводной таблицы — индекс k в таблице функций типа M. По этому индексу получается ссылка на функцию, которая затем вызывается. Таблицы функций предоставляются непосредственно WebAssembly; сводная таблица записывается в память программы перед её запуском.

Строковый тип данных реализован как ссылочный массив 8-битных знаковых чисел. Этот тип данных также предоставляется Wasm GC. Функции для работы с этими массивами встраиваются компилятором в программу во время трансляции. Для удобства был создан DSL, использующий макросы языка Rust (на нём написан компилятор), с помощью которого достигается близость к синтаксису текстового формата WebAssembly; код встроенных функций реализован при помощи этого DSL.

Ошибки во время исполнения (деление на 0, обращение за пределы строки) особым образом не обрабатываются: обе операции порождают исключительные ситуации виртуальной машины WebAssembly. Это уменьшает количество кода в получаемой программе, но теряется информация об ошибке. Встроенный в среду исполнения метод Object.abort, аварийно прекращающий работу программы, реализован средствами встраивания в JavaScript-окружение с помощью JavaScript-исключения.

2.4 Внешнее окружение

WebAssembly реализован как встраиваемый язык: стандартом предусмотрены средства внешнего взаимодействия с модулем WebAssembly. С помощью этих средств были реализованы на JavaScript процедуры для ввода-вывода (IO.read_int, IO.read_string, IO.out_int, IO.out_string).

Браузер не предоставляет блокирующего интерфейса для запроса ввода, а WebAssembly не имеет встроенного способа приостановки исполнения. Существуют решения (например, Asyncify в Binaryen[4]), позволяющие переписать программу на WebAssembly с целью добавить возможность приостановить исполнение вызовом специальной функции и возобновить позднее, однако они снижают производительность программ; кроме того, во время реализации компилятора никакое из этих решений не поддерживало используемые расширения WebAssembly. Поэтому подаваемые на вход программе данные должны указываться заранее, до её запуска.

3 Заключение

Был реализован компилятор из минималистичного объектно-ориентированного языка Cool в WebAssembly, позволивший проанализировать сложность трансляции различных конструкций языка. Выяснилось, что WebAssembly в совокупности с расширениями в целом позволяет выразить семантику объектно-ориентированного языка. Большая часть характерной для него функциональности имеет непосредственное отражение в WebAssembly: объекты, подтипизация и т. д. Однако имеются отдельные сложности.

Обработка исключительных ситуаций в текущей версии WebAssembly рудиментарна: стандартом не определяется интерфейс для перехвата и получения информации об ошибке. С целью добавить механизм исключений (и конструкций try/catch) ведётся работа над соответствующим предложением в стандарт.

В WebAssembly отсутствует механизм приостановления исполнения программы по её инициативе. Это делает трудным взаимодействие с асинхронным окружением браузера для реализации I/O-примитивов. Проблему частично решает алгоритм инструментации кода Asyncify из Binaryen, однако на текущий момент он не поддерживает локальные переменные ссылочных типов.

Таким образом, в сочетании с расширениями WebAssembly может служить целью компиляции современных объектно-ориентированных языков. В частности, уже ведётся разработка компиляторов для Kotlin[5], Dart[6], использующих расширение Wasm GC.

Литература

[1] A. Haas *et al.* Bringing the Web up to Speed with WebAssembly. *SIGPLAN Not.*, vol. 52, no. 6, pp. 185–200, Jun. 2017, doi: 10.1145/3140587.3062363.

[2] A. Aiken. The Cool Reference Manual. 2000. https://theory.stanford.edu/~aiken/software/cool/cool-manual.pdf

[3] A. Rossberg. GC extension. 2019. https://github.com/WebAssembly/gc/blob/master/proposals/gc/Overview.md

[4] Binaryen contributors. Binaryen. https://github.com/WebAssembly/binaryen

[5] JetBrains. Kotlin Wasm. https://kotlinlang.org/docs/wasm-overview.html

[6] The Dart project authors. Compiling Dart to WebAssembly. Jul. 21, 2023. https://github.com/dart-lang/sdk/blob/b6e3c5cffd859e33a5393da1244f575c320935cc/pkg/dart2wasm/README.md

Modern seismic vizualization tools

Lyutikova Veronika Sergeevna[1], Litovchenko Irina Nikolaevna[2]

[1,2] *LLP "Institute of Seismology" of the Ministry of Emergency Situations of the Republic of Kazakhstan, al-Faraby 75a, Almaty, Kazakhstan*

Nikki.vallo16@ gmail.com, *litovira@rambler.ru*

Annotation. *The article provides information about modern tools and the main elements of the visual presentation of data, as well as a tool for visualizing seismic data. The field of application is seismology and means of visualization of seismic data. The paper presents the main theoretical and practical results of applying the pattern recognition algorithm based on the method of graphical clustering (MGCl) of data - using the example of earthquake swarms. An attempt was made to assess the prospects for development and its application.*

Key words: visualization, seismic data, modern software

1 Introduction

In the modern world, information visualization tools are becoming very important. Information is becoming more and more open. The question arises about the search for forms and modern means of its presentation. From a large amount of heterogeneous information that comes to us today from various sources, it is sometimes difficult to single out the main thing, and it is easy to get lost in the data stream [4,5]. Graphic information is perceived several times faster than text information. In addition, the human perception of graphic images is associative. With the development of modern visualization tools and software, infographics are gaining popularity - a special type of presentation of information, data and knowledge in a graphic format, the main advantages of which are the speed of understanding, visibility and accessibility [4]. Graphs, as one of the simplest elements, are used to show the relationship of one set of data to another. Graphs are of several types. Linear are sets of points connected by a line, corresponding to the values along the axes. Scatterplots show the distribution of a set of points corresponding to values along the axes. Graphs are a subtype of charts. In addition, charts can be columnar (histograms), circular, ring, petal, etc. [4]. Cycle diagrams show the key steps of a process that contains a set of repetitive actions. Separately, it is worth noting such types of charts as a heat map and a flat tree. A heat map compares values within a dataset by coloring them in one of the colors of the spectrum. A flat tree representing a dataset hierarchy in which elements are parent or child of each other is displayed as a set of nested rectangles, each of which is a [4,5] branch. Trees and link diagrams are somewhat different from the representations mentioned above. They are designed to show the structure (hierarchy) of a data set, to reflect the relationship of individual components. These representations are widely used to visualize mathematical structures such as graphs and networks. The visual display of the graph is a collection of nodes connected by lines - edges. Trees and mental maps are special cases of graphs [4,5] . Maps are another popular visualization tool. Maps can visualize the unemployment rate by region, identify areas affected by floods, the seismicity of the region, etc. In other words, maps and cartograms allow you to quickly identify geographic and temporal patterns, which cannot be achieved using only pivot tables with data. In addition, using thematic markers on the map, you can show the location of any objects, for example, strong earthquakes. Thus, based on the whole variety of means of representing, reflecting and visualizing information, you can choose the most convenient methods. Let's move on to the features of modern seismic data visualization tools and the analysis of the Earth's seismicity [1,2, 6-9].

2 Theoretical problems

2.1 Statement of the problem

Based on the world catalog of earthquakes [3], the authors performed a visual representation of seismic data based on modern software tools - a 3- D representation of earthquake epicenters on a virtual

globe in the Origin software environment [10]. The main modern visualization tools can be found in more detail in [4]. Origin is an OriginLab Corporation software package for numerical data analysis and scientific graphics that runs on a computer running the Microsoft Windows operating system . To perform operations, you can both use the GUI tool user (dialogs/menus) and call them in programs. Origin includes its own compiler C / C++ with support and optimization of vector and matrix calculations. Origin is designed to create 2D, 3D scientific graphics that are created using ready-made templates available for user editing. Once an image has been created, it can be edited using menus and dialogs invoked by double-clicking on its elements. You can export the resulting graphs and tables in a number of formats [10]. With the help of Origin, it is possible to carry out numerical analysis of data, including various statistical operations, signal processing, etc. The authors performed an analysis of seismic data and performed a 3D visualization of the global seismicity of the Earth in the Origin environment . Figure 1 shows the visualization of the global seismicity of the Earth on a 3D virtual globe

2.2 Drawings

The result of visualization of seismic data by modern means demonstrates the global seismicity of the Earth on a virtual globe. The practical value of such a representation lies in the fact that it is directly visible where and how all earthquake epicenters are distributed on a virtual globe. In the Origin software environment, the virtual globe can be rotated, which even better demonstrates the visualization of earthquakes on it. With the change of seismic data, it is possible to track their visualization on a virtual globe in time and space (see picture 1). Figure 1 shows seismic data visualized in the Origin environment.

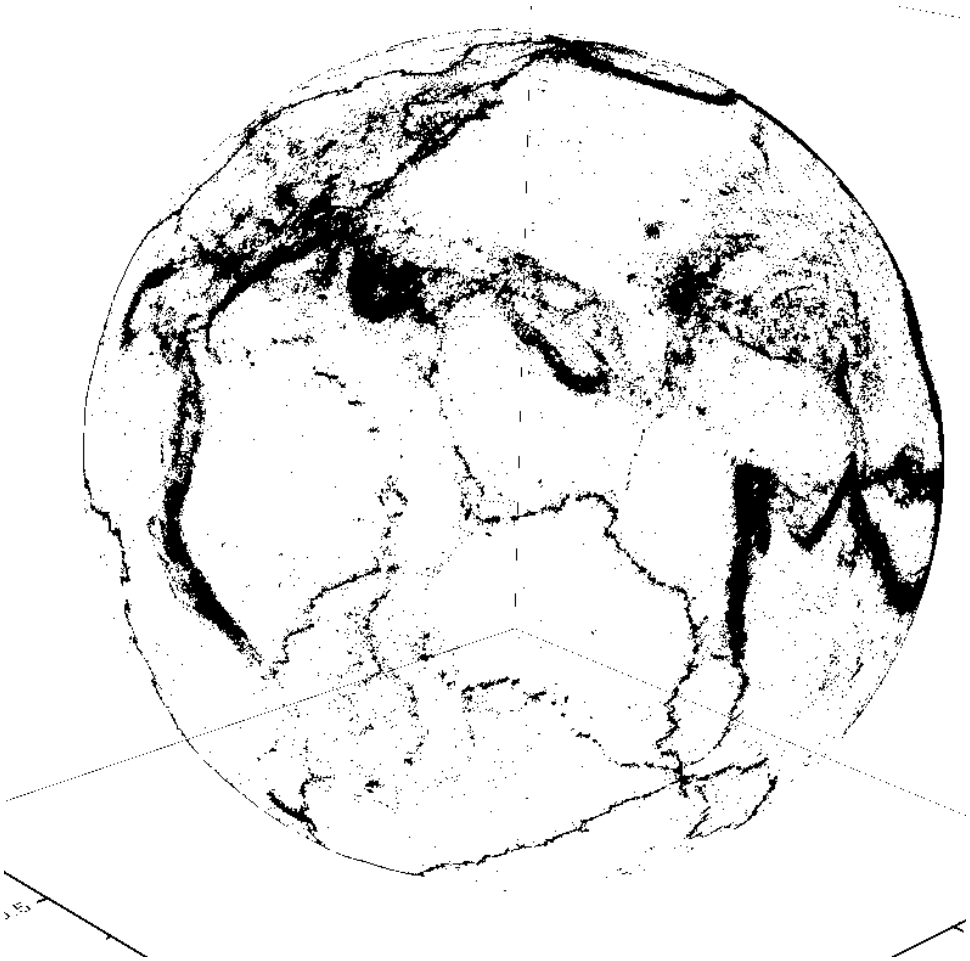


Fig.1. Global seismic data visualized in the Origin environment [3,10]

The work was carried out in the laboratory of physics of geodynamic and seismic processes within the framework of the PCF "Evaluation of the seismic hazard of the territories of regions and cities of Kazakhstan on a modern scientific and methodological basis", program code F.0980. Funding source - Ministry of Education and Science of the Republic of Kazakhstan

3 Conclusion

In conclusion, it can be noted that a lot of scientific and practical work has been done on the visualization of seismic data using modern computing and software tools. The results obtained are applied in practice in the study of the spatio-temporal distributions of earthquakes on a virtual globe.

4 Acknowledgments

The authors are grateful to Academician of the National Academy of Sciences of the Republic of Kazakhstan Kurskeyev A.K. for scientific leadership.

Literature

[1] Kurskeyev A.K. Gravitational interaction of the planets of the solar system and the seismicity of the Earth " Evero ". Almaty. Kazakhstan. 2021.-p.547.

[2] Kurskeyev A.K., Kolumbetova K.K., Litorvchenko I.N., Amirov N.B., Lyutikova V.S. On the physical nature of the magnitude of earthquakes / / Int. scient. - pract.conf . "Modern methods for assessing seismic hazard and earthquake prediction " - Almaty, 2022.- 141-148.

[3] World catalog of earthquakes/ /[Electronic resource]: access mode: http :// www . earthquake . usgs . gov / earthquakes / search

[4] For structured data in client web applications . scientific conf . (Moscow, July 2014). - T. 0. - Moscow: Buki-Vedi, 2014. - S. 14-19. — URL: https://moluch.ru/conf/tech/archive/90/5910/ (date of access: 01/23/2023).

[5] Vetrov Yu. A. Data visualization. Visual and compact way to display information // URL: http://www.jvetrau.com/visualization/

[6] Lyutikova V.S., Ismailova R.T. Modern means of pattern recognition (on the example of swarms of earthquakes) // "Bulletin of Turan " .- Almaty, 2022.-5p.

[7] Lyutikova V.S., Litovchenko I.N., Swarm activity in the seismicity of the Northern Tien Shan and adjacent territories / / Int. scient. - pract.conf . "Modern methods for seismic hazard assessment and earthquake prediction.-Almaty, 2022.- CC . 281-287.

[8] Lyutikova VS, Litovchenko IN, Amirov NB Activation of Weak Seismicity as an indicator of the forvation of source zone of strong earthquekes in the earth's crust Northern Tien Shan// Materials Int . scientific method . _ journal "Global Science and innovations 2022: Central Asia". - Nur - Sultan , 2022. - SS .3-7.

[9] Litovchenko I.N., Amirov N.B., Lyutikova V.S. Earthquake Swarm Pattern Recognition and Their Numerical Characteristics// Innovative Technologies and Geospatial Digital Engineering. Intern.scient.-pract. Conf., Almaty, 2022.-SS.549-555.

[10] https://en.wikipedia.org/wiki/Origin_2022)

Указатель авторов

Аникин М.А. с. 10
Апанович З.В. с. 169
Артемьева И.Л. с. 19
Архипова В.И. с. 369
Ахмедов Э.Ю. с. 24
Балашова С.А. с. 371
Балчиков В.И. с. 31
Барахнин В.Б. с. 39
Бахмутова И.В. с. 212
Бериков В.Б. с. 45, 154, 295
Болотов К.Ю. с. 401
Вартазарян Э.А. с. 375
Вопилова Е.В. с. 49, 54
Галиева А.Г. с. 60
Гатиатуллин А.Р. с. 66
Глушкова В.Н. с. 73
Гринберг Э.Я. с. 83
Губанов Д.А с. 89
Гуртуева О.А. с. 376
Гусев В.Д. с. 212
Давыдов А.В. с. 95
Демидов Д.В. с. 105
Демчук М.Е. с. 378
Долгов Н.Ю. с. 114
Доманов О.А. с. 120
Дробышевский И.А. с. 379
Дымонт Н.А. с. 382
Дьяченко М.А. с. 383
Зайцев А.О. с. 129
Зыкова А.А. с. 384
Иванов А.И. с. 407
Иванус А.И. с. 137
Ильин В.П. с. 145
Каблуков И.В. с. 299
Кладов Д.Е. с. 154
Климонтов В.В. с. 154
Ковалевская О.А. с. 390
Козицын А.С. с. 162
Колганова А.В. с. 169
Копылова О.А. с. 188
Копытков Д.А. с. 393
Коровина К.С. с. 73
Королева М.Н. с. 10
Крючкова Е.Н. с. 54
Кубельский М.В. с. 83
Кузнецов О.П. с. 89
Курдюков Д.А. с. 395
Кутненко О.А. с. 45
Ланин В.В. с. 31
Ларионов А.А. с. 95
Левичев А.В. с. 195, 337
Лобовиков В.О. с. 202
Мадиримов Ш. с. 39
Малаева Е.Д. с. 396
Мардер Л.М. с. 105
Мацько А.М. с. 397
Менглиев Д.Б. с. 39
Миронов В.С. с. 398
Мирошниченко Л.А. с. 212
Михалёв М.А. с. 417
Нагул Н.В. с. 95
Нещадим М.В. с. 195
Осипов А.А. с. 401
Осипов А.Л. с. 222
Павлова А.И. с. 222
Пальчунов Д.Е. с. 60, 129
Пальчунова О.Д. с. 407
Пальянов А.Ю. с. 195, 337
Панова В.Е. с. 402
Пахомов И.В. с. 229
Пашкова А.С. с. 407
Петров С.Е. с. 403
Попова П.В. с. 404
Привезенцев А.И. с. 235
Прокопьев Н.А. с. 66
Радеев Н.А. с. 406
Рева Г.В. с. 313
Резников В.М. с. 246
Садриев В.Д. с. 407
Смоляков П.Е. с. 413
Сулейманов Д.Ш. с. 66
Суров И.А. с. 255
Сухих А.О. с. 414
Суховеров В.С. с. 89
Тимофеев А.Т. с. 416
Тихомолов Я.А. с. 417
Трегубов А.С. с. 129, 307
Филимонов В.А. с. 266

Хайретдинов М.С. с. 188
Хорошавин А.К. с. 272
Худяков Д.А. с. 418
Чеболтасова С.К. с. 419
Черновская Я.Т. с. 423
Чернявцева С.И. с. 279
Черпаков К.М. с. 285
Чудаков Д.С. с. 295
Чурилова Р.С. с. 424
Чусова А.Е. с. 19
Чхартишвили А.Г. с. 89
Шатрова А.И. с. 429
Шачнев Д.А. с. 162
Шелдяков И.П. с. 431
Шелехов В.И. с. 299
Шестаков В.К. с. 178
Шишкин А.А. с. 307
Щеглов Б.О. с. 313
Щеглова С.Н. с. 313
Щербин А.С. с. 432
Якобсон А.С. с. 433
Янбулатов Д.Р. с. 299
Янин А.С. с. 435
Akhmedova X.I. с. 348
Davronbekov D.A. с. 356
Khakimova S.S. с. 321
Kolonin A.G. с. 329
Litovchenko I.N. с. 439
Lyutikova V.S. с. 439
Masharipov S.A. с. 348
Matqurbanov T.A. с. 356
Matyokubov U.K. с. 356
Rakhimova L.S. с. 348
Sadikov M.A. с. 365
Sobirov B.I. с. 365

ЗНАНИЯ - ОНТОЛОГИИ - ТЕОРИИ

Материалы IX Международной конференции
“Знания - Онтологии - Теории”

Научный редактор Д.Е. Пальчунов.

ISSN 0568-661X