Знания-Онтологии-Теории (ЗОНТ-09)

Математическая Модель Порождения Правил Синтаксической Сегментации

Манушкин Е.С. [1], Клышинский Э.С.[1]

[1]*Московский государственный институт электроники и математики, Б. Трехсвятительский пер., д. 3/12, г. Москва, 109028, Россия.*

EugeneLebowsky@mail.ru, Klyshinsky@mail.ru

Аннотация. *В работе изложен метод автоматической генерации правил синтаксической сегментации на основе вычисления множеств FIRST, LAST, FIRST2 и LAST2 для формальной грамматики, записанной в виде бэкусовских нормальных форм, предназначенной для синтаксического анализа текстов на естественном языке. При генерации правил учитывается возможность согласования слов. На основе вычисленных множеств решается обратная задача – определяется, какие терминалы порождают цепочки, начинающиеся с данных символов, после чего генерируются правила синтаксической сегментации.*

Ключевые слова: синтаксический анализ, синтаксическая сегментация, автоматическое обучение, согласование слов

1 Введение

При классическом подходе к решению задач автоматической обработки текстов (АОТ) путем полного анализа вычислительные затраты растут экспоненциально с ростом длины анализируемых фрагментов текста. Кроме того, создание систем правил для АОТ сопоставимо со сложностью создания программного обеспечения для таких систем. Количество правил в реальных задачах таково, что в них так же легко запутаться, как в многочисленном и объемном программном коде. В связи с этим начали активно развиваться методы, позволяющие автоматически генерировать правила для АОТ. Среди прочего это произошло благодаря тому, что, с одной стороны, в сети Интернет накопилось достаточное количество одноязыковых и параллельных корпусов текстовС другой стороны, само развитие вычислительной техники позволило перейти к решению подобных задач.

Ярким примером системы, работа которой основана на обработке таких параллельных текстов, может служить система Google Translate (http:\\translate.google.com\). Однако используемая для данной системы генеративная модель, предназначенная для трансфера при машинном переводе, не обеспечивает решения таких задач, как составление рефератов, машинный диалог, анализ содержимого текстов. В связи с этим задача полного анализа текстов на естественном языке сохраняет свою актуальность.

На основе работы с корпусами текстов развиваются статистические методы и для полного анализа. Так, например, ведутся работы по автоматизированному наполнению морфологических словарей за счет анализа несловарной лексики [1]. На основе статистического выделения n-грамм проводится снятие омонимии во входных текстах в ходе предсинтаксического анализа [2]. Однако генерация правил для самого сложного этапа – синтаксического анализа – до сих пор не проводится.

Для ускорения работы синтаксического анализа может использоваться предшествующий ему этап синтаксической сегментации, который, выделяет априорную информацию о структуре предложения

на основе выделения его фрагментов [3] или выделяет составные конструкции [4,5]. Одной из задач синтаксической сегментации является определение границ структурных единиц предложения. Путем анализа входного предложения можно выдвинуть некоторые предположения о том, с какого слова начинаются те или иные синтаксические группы, а также каким словом они заканчиваются. Вместо того чтобы перебирать все возможные варианты разбора предложения, система синтаксического анализа будет проводить разбор исходя из того, что слова фрагмента принадлежат заданной синтаксической группе, не предпринимая неуспешных попыток выйти за заданные границы. За счет этого резко сокращается количество вариантов разбора входного предложения и, как результат, существенно уменьшаются вычислительные затраты.

Если синтаксический анализ предложения ведется с использованием формальной грамматики, записанной в виде бэкусовских нормальных форм (БНФ), то каждой синтаксической группе должно быть сопоставлено некоторое правило, входящее в эту грамматику. Таким образом, задачу синтаксической сегментации можно сформулировать как сопоставление слов входного текста правилам формальной грамматики.

Выделение и написание правил синтаксической сегментации также является трудоемкой и сложной задачей. Автоматическая генерация правил синтаксической сегментации позволит снизить нагрузку на людей, задействованных в создании правил, и повысить качество анализа самих правил.

В данной работе была поставлена цель описать методику автоматизированного порождения правил синтаксической сегментации на основе уже существующей формальной атрибутивной грамматики, записанной в виде БНФ и предназначенной для синтаксического анализа текстов на естественном языке. Построенная математическая модель была проверена на примере имеющейся у авторов грамматики русского языка.

2 Необходимые теоретические сведения

Общепринятым алгоритмам вычислений множеств FIRST, LAST, FIRST2, LAST2 для описания естественных языков не хватает механизмов учета согласования. Для учета этой особенности будем использовать параметры. Определим параметр как пару p=<N, V>, где N — имя параметра (падеж, род и т.д.), а V — его значение (именит., родит., …). Таким образом, каждое слово в тексте будет обладать не только нормальной формой и частью речи, но и набором параметров, указывающих на форму, в которой находится данное вхождение слова. Все эти данные могут быть получены в результате морфологического анализа. Входное слово будет записываться как $\bar{a} =< a, P_w >$, где a – нормальная форма и часть речи входного слова, а P_w={p} – множество параметров данного слова.

В ходе синтаксического анализа параметры используютсядля проверки значений параметра на совпадение или несовпадение с конкретным значением, согласование параметров между собой, добавить новый параметр к слову и т.д.. Для терминальных символов введем типизированный параметр, обладающий также типом: p_t=<N, V, T>, где T – тип параметра. Тип параметра будет определять набор действий, производимых при помощи данного параметра. Каждому символу приписывается некоторое множество типизированных параметров. Не нарушая общности рассуждений, разобьем все параметры в зависимости от приписанного им типа на два множества: множество P_c={p_t} параметров, используемых для проверки согласования конструкций, и множество P_o={p_t} остальных параметров. Множество всех параметров символа обозначим как P_t: P_t=P_c∪P_o.

Таким образом, терминал будет записываться как $\bar{a} =< a, P_t^a >$, нетерминал запишется как $\bar{A} =< A, P_t^A >$. Здесь a – терминал без добавления к нему параметров, A – нетерминал без добавления к нему параметров, P_t^X – множество параметров для символа X.

При сравнении терминального символа с конкретным словом типизированные параметры терминала сравниваются с параметрами этого слова. Если помимо стандартных операций все сравнения типизированных параметров проведены успешно, то считается, что терминал успешно сравнился со словом. Заметим, что сравнивается полное множество параметров терминала P_t. При этом сравнение множеств параметров проводится следующим образом:

$$\bar{b} = \bar{a} : \bar{b} =< b, P_t^{\bar{b}} >, \bar{a} =< a, P_w^{\bar{a}} >, a \sim b, \forall i = \overline{1,n}\ \exists j \in [1,m] : P_{ti}^{\bar{b}} \sim P_{wj}^{\bar{a}}. \quad \text{(I)}$$

Здесь $\overline{b}$ – терминал, $\overline{a}$ – входное слово, n и m – количество параметров у $\overline{b}$ и $\overline{a}$, соответственно, $a \sim b$ означает, что терминал успешно применим ко входному слову, $P_{ti}^{\overline{b}} \sim P_{wj}^{\overline{a}}$ означает, что i-й параметр множества P_t^b успешно сравним с j-м параметром множества P_w^a.

С учетом наличия параметров согласования продукцию можно переписать в виде $\overline{B} \to \overline{\alpha_1}...\overline{\alpha_n}$, где $\overline{B} =< B, P_c^B >, \overline{\alpha_1} =< \alpha_1, P_c^{\alpha_1} >,..., \overline{\alpha_n} =< \alpha_n, P_c^{\alpha_n} >$. При проверке согласования слов в тексте значения параметров берутся для терминалов – от соответствующих слов из текста, а для нетерминалов – из результатов разбора успешно примененных продукций.

Для каждой продукции $\overline{B} \to \overline{\alpha_1}...\overline{\alpha_n}$ можно определить множество имен параметров, используемых для согласования: $C_N = \bigcup_{i=1}^{n} \bigcup_{j=1}^{m_i} N_{ij}$, где m_i – количество параметров у i-го символа продукции, N_{ij} – имя j-го параметра i-го символа продукции. В этом случае можно сказать, что мощность множества, объединяющего значения каждого согласовывающегося параметра $V_{Ni} = \bigcup_{j=1}^{m_i} V_{ij}$ равна 1, либо равна 2 и включает в себя значение параметра, согласовывающееся с любым другим значением (в случае, если такой параметр определен).

$$V_{Ni} = \bigcup_{j=1}^{m_i} V_{ij}, |V_{Ni} - 0| = 1, \quad \text{(II)}$$

где 0 – значение параметра, сравнимое с любым другим значением данного параметра.

Таким образом, продукция может считаться успешно сравнившейся, если для каждого терминального символа в ней выполняется условие (I), разбор каждого нетерминала завершен успешно и для продукции выполняется условие (II). Значения параметров для нетерминалов берутся из описанных выше объединенных множеств значений параметров.

Дадим теперь определения множеств FIRST2' или LAST2' для терминалов, обладающих параметрами. Для этого определим множество FIRST', состоящее из терминалов, входящих в FIRST($\overline{A}$), с приписанными к ним множествами параметров согласования.

Пусть $\overline{A} =< A, P_c^{\overline{A}} >, \overline{a} =< a, P_c^{\overline{a}} >$, причем $A \in VN, \overline{a} \in VT$. Тогда $\overline{a} \in FIRST'(\overline{A})$ при условии, что $\exists \overline{b} =< a, P_c^{\overline{b}} >: \overline{A} \to \overline{b}\overline{\alpha}$, причем $a \in FIRST(A)$ и $P_c^{\overline{a}} = P_c^{\overline{A}} \cap P_c^{\overline{b}}$. То есть множеству $FIRST'(\overline{A})$ принадлежит не сам символ $\overline{b}$, с которого могут начинаться выводимые из $\overline{A}$ цепочки, а его модификация, в которой множество параметров определяется как пересечение параметров согласования нетерминала $\overline{A}$ и терминала $\overline{b}$, а остальная информация берется от $\overline{b}$.

Множество LAST' будет определяться аналогично для цепочек вида $\overline{A} \to \overline{\alpha}\overline{b}$.

Введем новое множество ONLY'(α), которое будет содержать множество цепочек, выводимых из α и состоящих из единственного терминального символа. Оно будет определяться аналогично для цепочек вида $\overline{A} \to \overline{b}$.

Определим теперь множество DirectFIRST2' как множество пар согласуемых терминалов с приписанными к ним типизированными параметрами. Пусть дана продукция вида $\overline{B} \to \overline{\alpha_1}\overline{\alpha_2}\beta$, где β может являться произвольной цепочкой символов и $\overline{B} =< B, P_c^{\overline{B}} >, \overline{a} =< a, P_c^{\overline{a}} >, \overline{b} =< b, P_c^{\overline{b}} >$, причем $\overline{a}, \overline{b} \in VT$. Тогда $\overline{a}\overline{b} \in$ DirectFIRST2'($\overline{B}$), если $\overline{a} \in$ ONLY'($\overline{\alpha_1}$), $\overline{b} \in$ FIRST'($\overline{\alpha_2}$) и $P_c^{\overline{ab}} = P_c^{\overline{a}} \cap P_c^{\overline{b}}$. То есть, из множеств параметров каждого из символов мы можем отбросить те

параметры, которые участвуют в согласовании, но присутствуют лишь в одном из терминалов. Аналогичным образом переопределим множество DirectLAST2' на множестве пар терминалов, содержащих типизированные параметры.

Введем множество MIDDLE2'(α), показывающее какие пары терминалов образовываются в середине цепочек, выводимых из α. Рассчитаем множества MIDDLE2' по следующему алгоритму. Пусть рассматривается продукция вида $\overline{\mathrm{B}} \to \alpha\overline{\mathrm{A}_1}\,\overline{\mathrm{A}_2}\beta$, причем α и β могут являться произвольными цепочками символов. Пусть $\overline{a},\overline{b} \in VT$. Тогда для α и β возможны следующие варианты.

1. Пусть α=e и β≠e. Тогда $\overline{a}\,\overline{b}$ ∈MIDDLE2'($\overline{\mathrm{B}}$), $\overline{a}$ ∈LAST'($\overline{\mathrm{A}_1}$), $\overline{a}$ ∉ONLY'($\overline{\mathrm{A}_1}$), $\overline{b}$ ∈FIRST'($\overline{\mathrm{A}_2}$).
2. Пусть α≠e и β=e. Тогда $\overline{a}\,\overline{b}$ ∈MIDDLE2'($\overline{\mathrm{B}}$), $\overline{a}$ ∈LAST'($\overline{\mathrm{A}_1}$), $\overline{b}$ ∈FIRST'($\overline{\mathrm{A}_2}$), $\overline{b}$ ∉ONLY'($\overline{\mathrm{A}_2}$).
3. Пусть α=e и β=e. Тогда $\overline{a}\,\overline{b}$ ∈MIDDLE2'($\overline{\mathrm{B}}$), $\overline{a}$ ∈LAST'($\overline{\mathrm{A}_1}$), $\overline{a}$ ∉ONLY'(A_1), $\overline{b}$ ∈FIRST'($\overline{\mathrm{A}_2}$), $\overline{b}$ ∉ONLY'($\overline{\mathrm{A}_2}$).
4. Пусть α≠e и β≠e. Тогда $\overline{a}\,\overline{b}$ ∈MIDDLE2'($\overline{\mathrm{B}}$), $\overline{a}$ ∈LAST'($\overline{\mathrm{A}_1}$), $\overline{b}$ ∈FIRST'($\overline{\mathrm{A}_2}$).
5. MIDDLE2'(α)∪MIDDLE2'($\overline{\mathrm{A}_1}$)∪MIDDLE2'($\overline{\mathrm{A}_2}$)∪MIDDLE2'(β)⊂MIDDLE2'($\overline{\mathrm{B}}$).

В этом случае возможен вариант, когда пара терминалов будет входить в множества несколько раз с различными наборами согласовываемых параметров. Это будет происходить в связи с тем, что единичный терминал может порождаться из данного нетерминала более чем одним способом. При этом при порождении различными способами будет формироваться различный набор параметров.

Также определим множества backFIRST2' и backLAST2'.

backFIRST2($\overline{a}\,\overline{b}$)={ $\overline{A}$ }: $\overline{a}\,\overline{b}$ ∈DirectFIRST2'($\overline{A}$), ∀ $\overline{B}$: $\overline{a}\,\overline{b}$ ∉MIDDLE2' ($\overline{B}$)

backLAST2($\overline{a}\,\overline{b}$)={ $\overline{A}$ }: $\overline{a}\,\overline{b}$ ∈DirectLAST2'($\overline{A}$), ∀ $\overline{B}$: $\overline{a}\,\overline{b}$ ∉MIDDLE2' ($\overline{B}$)

Данные множества определяют, какие нетерминалы могут начинаться или заканчиваться данной парой согласуемых терминалов. При этом считаем, что пара $\overline{a}\,\overline{b}$ не может образовываться в середине выводимых цепочек за счет сцепления хвоста и головы двух цепочек (для этого в определение функций были введена проверка отсутсвия в множестве MIDDLE).

3 Метод генерации правил

На основании вычисления backFIRST2 и backLAST2 можно определить, какие правила начинаются или заканчиваются данным терминалом или парой терминалов. Будем говорить, что пара терминалов $\overline{a}\,\overline{b}$ является характеристической, если мощность множества backFIRST2($\overline{a}\,\overline{b}$) или backLAST2($\overline{a}\,\overline{b}$) равна единице. Будем говорить, что пара терминалов $\overline{a}\,\overline{b}$ является характеристической по порогу R, если мощность множества backFIRST2($\overline{a}\,\overline{b}$) или backLAST2($\overline{a}\,\overline{b}$) не превышает R. Аналогично можно определить и отдельные характеристические терминалы.

Если пара терминалов является характеристической, то встретив ее во входном потоке можно предположить, что с данной пары начинается (или заканчивается) единственное правило. Таким образом, весь дальнейший разбор должен вестись так, чтобы к моменту прохождения данной пары мы начали разбор с данного правила (или завершили этой парой разбор данного правила). За счет этого может быть получено существенное ускорение работы синтаксического анализа.

На основе этой информации можно построить правила синтаксической сегментации. Для множеств backFIRST2($\overline{a}\,\overline{b}$)={ $\overline{A}$ } с мощностью, равной единице, можно сформировать правила синтаксической сегментации вида «Если встретилось $\overline{a}\,\overline{b}$, то со слова, соответствующего терминалу $\overline{a}$, начинать разбор по правилу $\overline{A}$». Аналогично для множеств backLAST2(A) с мощностью, равной единице, можно сформировать правила вида «Если встретилось $\overline{a}\,\overline{b}$, то со слова, соответствующего терминалу $\overline{b}$, заканчивается разбор по правилу $\overline{A}$». Заметим, что при мощности множеств, превышающей единицу, также можно составить соответствующее количество правил, однако их применение будет носить вероятностный характер.

4 Результаты эксперимента

Для проверки изложенных положений нами был проведен вычислительный эксперимент. Была разработана программа, которая вычисляла значения множеств backFIRST2' и backLAST2'. На вход программы были поданы реально действующие грамматики русского и английского языков, разработанные для системы машинного перевода «Crosslator 2.0» [6]. По результатам вычислений были выделены наборы правил, соответствующие терминалам или парам терминалов.

Грамматика английского языка в эксперименте содержала около 450 правил. В результате генерации множеств FIRST2 и LAST2 было обнаружено, что более 900 пар терминалов однозначно идентифицируют начало правила и порядка 100 пар терминалов однозначно идентифицируют окончание правила. Для русского языка была взята грамматика из примерно 130 правил. Для нее было получено более 100 пар в множестве backFIRST2 и всего 13 в множестве backLAST2. Полученные результаты в основном являлись декартовым произведением небольшого количества конкретных слов, встречающихся рядом, что существенно снижает вероятность применения правил.

5 Обсуждение

В нашей нотации в зависимости от параметров терминал будет искать произвольный глагол или в мужским роде, может искаться как конкретный, так и произвольный предлог. В связи с этим один и тот же терминал может применяться к целому классу слов входного текста. Также справедливо и обратное: несколько различных терминалов могут применяться к одному и тому же слову.

В связи с этим возникает дополнительная неоднозначность: к одному и тому же месту в тексте может быть применено несколько конкурирующих правил. Однако подобная ситуация не так страшна. Дело в том, что мы в любом случае существенно сокращаем количество вариантов разбора, достигая тем самым поставленной цели. Вместо проверки всех возможных гипотез мы строим лишь несколько. Кроме того, вариативность результатов может быть изначально присуща этапу синтаксической сегментации (так как входной текст сам по себе может быть синтаксически неоднозначен), и нам не придется менять ход синтаксического анализа.

Также следует заметить, что порождаемые правила носят достаточно простой и частный характер. Однако метод гарантирует точность и однозначность их применения в ходе синтаксической сегментации, тогда как ручное создание подобных правил требует тщательной проверки.

Литература

[1] Ляшевская О.Н., Кобрицов Б.П., Сичинава Д.В. Автоматизация построения словаря на материале массива несловарных словоформ // Интернет-математика 2007

[2] Сокирко А.В., Толдова С.Ю. Сравнение эффективности двух методик снятия лексической и морфологической неоднозначности для русского языка // Международная конференция «Корпусная лингвистика 2004». С.-Пб., 2004

[3] Кобзарева Т.Ю., Лахути Д.Г., Ножов И.М. Модель сегментации русского предложения // Труды Международного семинара Диалог'2001, том 2, Аксаково, 2001

[4] Большакова Е.И., Васильева Н.Э. Терминологическая вариантность и ее учет при автоматической обработке текстов // Труды одиннадцатой национальной конференции по искусственному интеллекту КИИ-2008, том 2, Дубна, 2008

[5] Добров Б.В., Лукашевич Н.В., Сыромятников С.В. Формирование базы терминологических словосочетаний по текстам предметной области // Труды пятой всероссийской научной конференции "Электронные библиотеки: Перспективные методы и технологии, электронные коллекции". - 2003, с. 201-210.

[6] Модуль фрагментарного анализа в составе системы машинного перевода. Crosslator 2.0 // Вестник ВИНИТИ, 2005 г. НТИ. Серия 2. № 8 сс. 31-33