Знания-Онтологии-Теории (ЗОНТ-09)

Предварительная обработка строк при критическом коэффициенте джаккарда для улучшения вычисления схожести веб-документа

Неелова Наталия Валериевна

Тульский государственный университет, пр. Ленина, д. 92, г. Тула, 300600, Россия.

neelova.natalia@gmail.com

Аннотация. *В данной статье рассматривается современная ситуация, связанная с вопросами детектирования дубликатов поисковыми машинами. Выделяются основные преимущества алгоритма Джаккарда, который лучше других подходит для реализации системы online фильтрации дубликатов веб-документов в промышленных масштабах, но неустойчив к орфографическим ошибкам и синонимическим заменам. Предложено усовершенствование метода Джаккарда, которое учитывает недостатки данного алгоритма. Разработана математическая модель вычисления схожести строк при критическом коэффициенте Джаккарда с предварительной обработкой по словарю синонимов. Исследована эффективность разработанной модели. Сделано заключение об улучшении полноты результатов вычисления схожести строк с возможностью сохранения высокой скорости обработки.*

Ключевые слова: поиск в интернете, нечеткие дубли, схожесть Джаккарда, вычисление схожести строк, полнота вычисления

1 Введение

Актуальность проблемы определения дублей в веб-области поисковых систем связано со значительным расширением индексных баз, требующих для своего хранения и обработки значительных временных затрат и затрат дискового пространства, с ухудшением качества выдачи поисковых алгоритмов на запрос пользователя, с необходимостью разработки методов определения первоисточника и борьбы с копирайтингом.

Исследование указанной проблемы активно ведется в настоящий момент. Большинство работ направлены в сторону усовершенствования поисковых алгоритмов определения дублей из-за глобального расширения сети Интернет. Подходы решения проблемы дублирования представлены методами offline (предварительной) и online (на лету) кластеризации. Метод offline кластеризация дублей применяют на этапе индексации сайта. Наличие лишь одной релевантной схожей части документа запросу пользователя при существенном отличии документов друг относительно друга делает невозможным кластеризацию. Данный факт является главным недостатком указанного подхода.

Однако если offline метод позволяет значительно сократить общую индексную базу поисковых систем, то схожесть документов относительно разных запросов будет разная при online подходе. Основной метод работы online кластеризации заключается в анализе текстов - сниппетов, выбранных поисковой системой как релевантные фрагменты поисковому запросу.

Таким образом, вычисление схожести строк часто является подзадачей вычисления схожести объектов и, следовательно, задачи поиска нечетких дублей. Одним из методов нечеткого сопоставления строк является подход, основанный на сопоставлении лексем – схожесть Джаккарда. Согласно исследованиям в [1] функция схожести Джаккарда лучше

других подходит для реализации системы online фильтрации дубликатов веб-документов в промышленных масштабах, т. к. имеет достаточно высокую производительность при незначительно худших результатах полноты. Неустойчивость алгоритма проявляется при наличии орфографических ошибок и синонимических замен.

В проведенном исследовании лексического метода вычисления схожести строк с учетом предварительной обработки [2] была разработана соответствующая математическая модель, которая позволила ограничить действие указанных недостатков. Также была предложена возможность использования разработанного метода при критических коэффициентах Джаккарда, тем самым снизив нагрузку на вычислительные ресурсы, уменьшив время обработки и увеличив полноту детектирования нечетких дублей.

В данном исследовании предлагается изучить эффективность предложенного метода при критических коэффициентах Джаккарда на заданном объеме данных. Оценить временные и качественные показатели предложенного подхода.

2 Теоретические проблемы

2.1 Функция схожести Джаккарда

Лексические методы сопоставления строк оперируют векторной моделью документов и рассматривают тексты как набор слов. Определение схожести на основе функции Джаккарда является лексическим методом сопоставления строк в векторном пространстве.

Формой записи схожести Джаккарда являются выражения:

$$sim(x, y) = \frac{\left|\{x_0,...,x_v\} \cap \{y_0,...,y_w\}\right|}{\left|\{x_0,...,x_v\} \cup \{y_0,...,y_w\}\right|} = \frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sum_i x_i^2 - \sum_i x_i y_i}, \quad (1)$$

где x, y – сравниваемые строки; v, w – количество лексем в сроках x и y соответственно; $\{x_0,...,x_v\}$, $\{y_0,...,y_w\}$ – множество лексем этих строк; i – количество лексем в сроках x и y.

Из формулы [1] видно, что мера принимает значение от 0 до 1 и достигает максимума при полном совпадении двух множеств. Выделяется два существенных недостатка данного подхода. Во-первых, данная мера не учитывает разницу в размере сравниваемых документов, во-вторых, при ее вычислении не используется информация о частоте употребления термов, составляющих документы. Также были указаны проблемы орфографических ошибок, коротких строк и синонимических замен. Однако в работе [2] данные недостатки были сглажены. Но разработанный способ привел к невозможности его использования в промышленных масштабах из-за значительных временных затрат, что по исследованию [1] является главным преимуществом функции Джаккарда среди других online методов.

2.2 Метрики эффективности алгоритмов поиска дубликатов

Большинство оценок текстового поиска основаны на отношении принадлежности документа кластеру. В задачах поиска дубликатов наилучшую характеристику эффективности работы дают точность и полнота.

Точность – способность системы выдавать в списке результатов только документы, действительно являющиеся дубликатами.

$$precision = \frac{a}{a+b},$$

где a – количество найденных пар дубликатов, совпадающих с «релевантными» парами; b – количество найденных пар дубликатов, не совпадающих с «релевантными» парами.

Полнота – способность системы находить дубликаты, но не учитывает количество ошибочно определенных недубликатов.

$$recall = \frac{a}{a+c},$$

где a – количество найденных пар дубликатов, совпадающих с «релевантными» парами; c – количество не найденных пар дубликатов, совпадающих с «релевантными» парами.

F-мера является гармоническим средним и часто используется как единая метрика, объединяющая метрики полноты и точности:

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall},$$

Соотношение полноты и точности, а также значение F-меры, зависит от пороговых значений схожести, при которых два документа считаются дубликатами. Так как в разных системах и задачах цена ошибки первого рода и ошибки второго рода является разной, то наиболее полное представление о характеристиках системы дает график полноты/точности.

2.3 Цель исследования

В работе [2] был сделан вывод, о возможности использования метода определения схожести строк на основе функции Джаккарда с предварительной обработкой строк. Под предварительной обработкой подразумевается отыскание синонимичных лексем, которые используются специально для автоматического рерайтинга статей и увеличения позиций в рейтинге поисковых систем. Данная разработка устранила неустойчивость алгоритма Джаккарда при наличии орфографических ошибок и синонимических замен (в работе вводится допущение о тождественности данных понятий).

Обращая внимание на увеличение популярности схемы искусственного увеличения авторитетности сайтов за счет размещения статей, чаще одинаковых по содержанию с изменением абзацев, использованием синонимов, перестановкой слов, применением рерайтинга, в работе [2] была доказана актуальность подобной разработки. Таким образом метод Джаккарда с предварительной обработкой показал 30 % улучшение детектирования дублей при автоматической генерации синонимичных строк.

Однако разработанный метод значительно увеличил время обработки сравнения строк, что свело к минимуму основное преимущество метода Джаккарда, что явилось препятствием для использования его в промышленных масштабах. Однако в работе были выделены предпосылки уменьшения временных и вычислительных ресурсов при сохранении полученной полноты вычисления: распараллеливание процесса расчета коэффициента, устранение стоп-слов, учет частоты слова.

Таким образом, можно выделить следующие цели данной работы:

– разработка алгоритма нахождения критических интервалов, определяющих принадлежность группы строк к соответствующему кластеру;

– сравнение эффективности алгоритма вычисления схожести строк с применением предварительной обработки при критических коэффициентах Джаккарда с традиционным алгоритмом и алгоритмом работы [2];

– сравнение временных показателей алгоритма вычисления схожести строк с предварительной обработкой при критических коэффициентах Джаккарда с традиционным алгоритмом и алгоритмом работы [2].

2.4 Постановка задачи

Пусть $\overline{x} = (x_1,...,x_d)$ – вектор характеристик объекта предметной области, где размерность d – количество характеристик объекта $\overline{x}$. В данном случае объектом предметной области выступает строка сниппета. $\overline{X} = (\overline{x}_1,...,\overline{x}_n)$ – множество объектов предметной области, i-й объект из $\overline{X}$ определяется как $\overline{x}_i = (x_{i1},...,x_{id})$.

Каждой *q*-ой характеристики объекта $\bar{x}$ можно поставить в соответствие набор $\bar{y} = (y_1,...,y_e)$ – вектор расширения характеристики объекта предметной области, где *e* – количество расширений характеристики объекта $\bar{x}_i$. В данном случае, под расширением понимается синонимичный ряд для каждого слова сниппета. При этом вектор $\bar{y}$ может быть нулевым, а для характеристики x_{iq} расширение определяется как $\bar{y}_{iq} = (y_{iq1},..., y_{iqe})$.

$\bar{S} = (\bar{s}_{11},..., \bar{s}_{nn})$ – множество матриц схожести двух объектов $\bar{X}$, где *n* – количество объектов $\bar{X}$. Каждая матрица доступа представляет набор соотношения всех лексем двух сниппетов: $\bar{s}_{i,j} = (s_{11},..., s_{1dj},..., s_{di1},..., s_{didj})$ – вектор схожести объектов $\bar{x}_i$ и $\bar{x}_j$ размерности *di* и *dj* соответственно, а $s_{ij} = \{0,1\}$.

В случае, если $y_{jgm} \equiv x_{ih}$, где $g \in [1, dj]$, а $m \in [1, e]$, тогда $x_{ih} \in (y_{jg1},..., y_{jge})$, где $h \in [1, di],$ а следовательно, $s_{ih,jg} = 1,$ то есть если расширенный ряд одой из характеристик соответствует сравниваемой характеристики, то это говорит о схожести данных характеристик и отражается в матрице схожести двух объектов $\bar{x}_i$ и $\bar{x}_j$. Применив формулу [1] к матрице $\bar{s}_{i,j}$ можно получить коэффициент Джаккарда.

Таким образом, реализация предварительной обработки строк при критическом коэффициенте Джаккарда заключается в алгоритме (рис.1.):

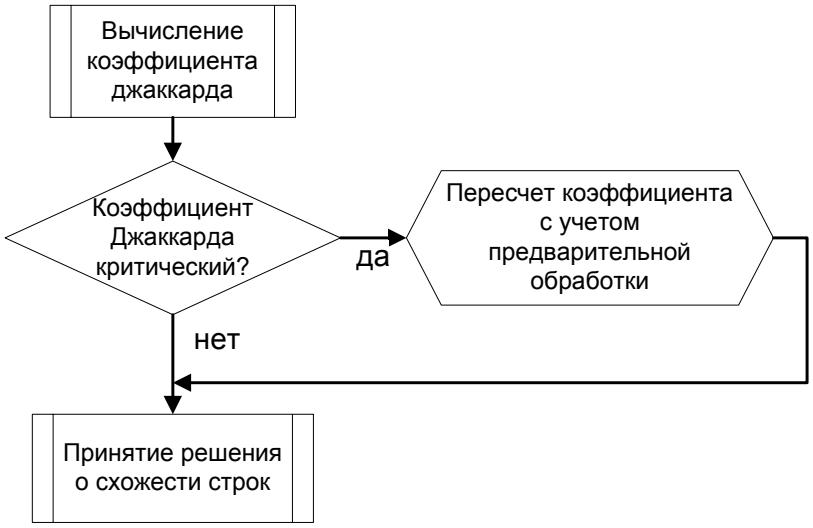

Рис.1. Схема процесса определения схожести строк на основе коэффициента Джаккарда с предварительной обработкой

Расчет коэффициента Джаккарда: $sim(x_i, x_j) = \frac{|\bar{x}_i \cap \bar{x}_j|}{|\bar{x}_i \cup \bar{x}_j|} = \frac{\sum \{s_{ih,jg} \mid s_{ih,jg} = 1\}}{d}$, где $\bar{x}_i$ и $\bar{x}_j$ - векторы предметной области, $s_{ih,jg}$ - элемент матрицы схожести $\bar{s}_{i,j}$, $d = |\bar{x}_i \cup \bar{x}_j|$ - мощность пересечения двух объектов.

Определение критичности рассчитанного коэффициента:
$$\begin{cases} sim(x_i, x_j) \in [sim_{kmin}; sim_{kmax}] \rightarrow n.3 \\ sim(x_i, x_j) \notin [sim_{kmin}; sim_{kmax}] \rightarrow n.4 \end{cases}$$

Вычисление схожести с учетом предварительной обработки: $sim(\bar{x}_i, \bar{x}_j) = \frac{|\bar{x}_i \cap \bar{x}_j|}{|\bar{x}_i \cup \bar{x}_j|} = \frac{\sum \{s_{ih,jg} \mid s_{ih,jg} = 1\}}{d - \sum \{s_{ih,jg} \mid s_{ih,jg} = 1 \wedge x_{ih} \in (y_{jg1},..., y_{jge})\}}$.

Принятие решения о схожести объектов на основе коэффициента Джаккарда.

Таким образом, можно поставить следующие задачи для построения метода определения схожести строк с предварительной обработкой при критическом коэффициенте Джаккарда:

1. Найти эффективный метод представления каждого из объектов предметной области $\bar{X}$ в виде вектора $\bar{x} = (x_1,...,x_d)$.

2. Разработка алгоритма нахождения критических интервалов коэффициента Джаккарда $[sim_{kmin}; sim_{kmax}]$.

3. Оптимальная реализация поиска $x_{ih} \in (y_{jg1},..., y_{jge})$ и учет найденных пар синонимов в пересечение множеств характеристик объектов.

3 Практическая реализация

3.1 Методика исследования

В исследовании принимали участия несколько десятков отобранных сниппетов, выдаваемых поисковой системой Яндекс. Каждое описание страницы представляло собой набор из 2-4 предложений. Далее каждая такая строка была размножена на 5 вариантов при помощи программы синонимайзера (программа, размножающая статьи при помощи синонимичных рядов), к которому подключался словарь синонимов. Этот словарь в дальнейшем участвовал в предварительной обработки матрицы схожести. В результате каждая строка имела 5 копий.

Для удовлетворения задачи представления каждого из объектов предметной области $\bar{X}$ в виде вектора был применен парсер mystem, предоставляемый компанией Яндекс [10]. Программа mystem производит морфологический анализ текста на русском языке и приводит все слова к нормальной форме. Для слов, отсутствующих в словаре, порождаются гипотезы. Таким образом, вектор объекта $\bar{x}$ – набор слов в нормальном виде рассматриваемой строки.

Критический интервал оценивался, исходя из матрицы традиционного алгоритма Джаккарда.

Для расчета коэффициента Джаккарда была реализована программа, которой задавался файл с обрабатываемыми строками, сравнивающимися между собой и файл с синонимичным словарем. Также была реализована возможность указания критического интервала вручную. Для оценки временных затрат, была реализована функция расчета затрачиваемого времени на обработку n-ого количества строк. На выходе программы имелось 6 файлов: общая матрица схожести Джаккарда n-строк между собой без предварительной обработки, общая матрица схожести Джаккарда n-строк между собой с предварительной обработки, общая матрица схожести Джаккарда n-строк между собой с предварительной обработки при критическом коэффициенте Джаккарда и 3 файла, соответствующих временных затрат на подсчет указанных матриц.

С помощью матриц рассчитывались показатели точности и полноты всех матриц. При этом по условию количество нечетких дублей было известно заранее. В результате были построены графики, оценивающие метрики разработанного алгоритма и графики эффективности алгоритма поиска нечетких дублей при критическом коэффициенте Джаккарда.

Таким образом, были решены поставленные выше задачи.

3.2 Результаты исследований

Используя статистические подходы, основанные на методах построения доверительных интервалов нормального распределения (характер распределения был установлен отдельно) и

полученный график распределения точек, характеризующих коэффициент Джаккарда (рис.2), был установлен доверительный интервал для определения нечетких дублей, соответствующий [35 %; 60 %]. Также был найден, нижний критический интервал соответствующий [25 % - 35 %]. Именно эти значения использовались при анализе эффективности предложенного метода расчета схожести строк.

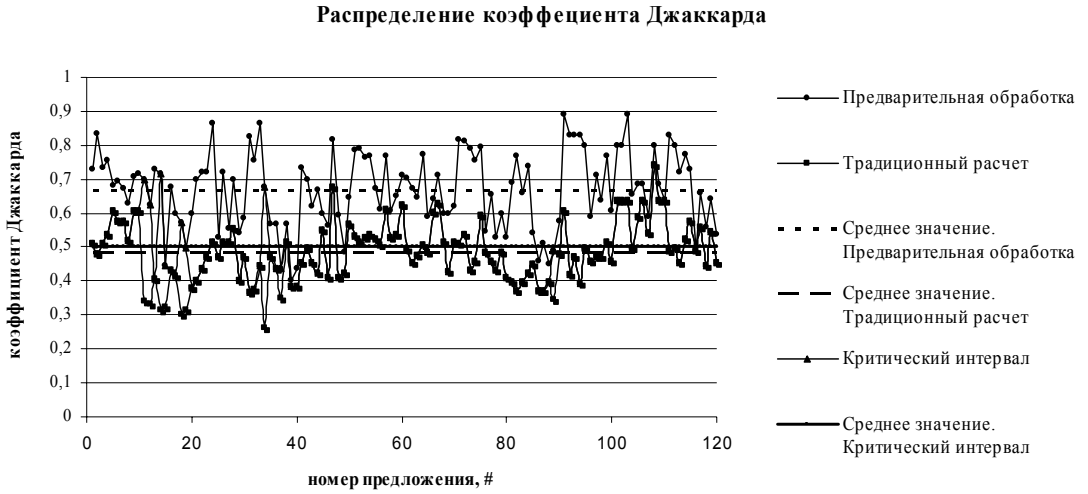

Рис.2. Графики распределения коэффициентов Джаккарда при разных режимах работы: традиционной, с предварительной обработкой, с предварительной обработкой при критических значениях коэффициента

После определения нижнего критического интервала к исходным данным была применена разработанная программа. В результате по полученным данным был построен график (рис.2), из которого видно, что при использовании функции Джаккарда с предварительной обработкой при критических коэффициентах значения функции в данных точках увеличиваются, что в свою очередь увеличивает полноту определения нечетких дублей, что видно из рис.3. Также об улучшении эффективности говорит увеличение среднего значения коэффициента Джаккарда.

На рис.3. видно, что как раз на критическом интервале [25 % - 35 %] значение полноты увеличивается, а, следовательно, все дубли попадут в область кластеризации. Значительного же увеличения полноты вычисления не произошло, для уточнения необходимо увеличить количество обрабатываемых строк для соответственного увеличения вероятности попадания в критический интервал.

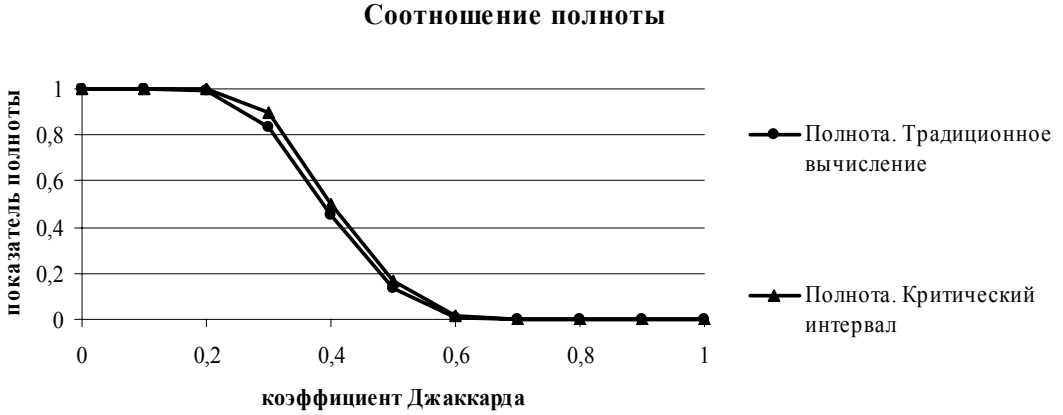

Рис.3. Соотношение метрики полнота при традиционном вычислении схожести и с предварительной обработкой при критических значениях Джаккарда

Для уменьшения временных показателей был усовершенствован метод поиска синонимов, процесс расчета коэффициентов был распараллелен, при начальной обработки строк устранялись стоп-слова. Результат проделанной работы представлен в таблице.

Таблица 1. Время расчета матрицы схожести Джаккарда.

Количество строк	Время обработки, *с*		
	Традиционная обработка	Предварительная обработка	Критические интервалы
2	47	564063	184
20	547	7177688	1084
40	1032	13081500	5216
60	1157	14444766	9617

Из таблицы видно, что использование предварительной обработки лишь при критических коэффициентах Джаккарда значительно снижает время расчета схожести между набором строк примерно на порядок 10^4. Однако относительно традиционной обработки данный метод все равно достаточно долог (примерно в 5-10 раз). Однако при уменьшении сравниваемых строк с критическим коэффициентом Джаккарда время значительно сокращается. Часто количество автоматического рерайтинга в выдачи поисковых систем зависит от тематики запроса.

4 Заключение

Предложенный метод расчета схожести двух строк с предварительной обработкой в виде проверки набора слов сравниваемых строк на синонимическую зависимость лишь при критическом коэффициенте Джаккарда показал хорошие результаты на заданном кластере автоматически созданных копий при использовании синонимайзера. Чем подключаемый словарь при создании копий ближе к словарю при детектировании нечетких дублей, тем процент эффективности метода больше.

Предложенное ограничение использования предварительной обработки значительно сократило время расчета матрицы схожести и увеличило полноту нахождения нечетких дублей в критическом интервале коэффициента Джаккарда.

При доработке программы работы [2] не была применена возможность усовершенствования алгоритма детектирования дублей за счет учета частоты встречающихся слов в строках. Также предсказана временная зависимость от тематики запроса пользователя. Согласно источнику [3] наибольшему автоматическому рерайтингу подвергаются страницы коммерческих запросов, а, следовательно, это отражается на сниппетах выдаваемых поисковой системой. В данном исследовании указанный факт не был рассмотрен.

Таким образом, при дальнейшем усовершенствовании временных показателей разработанный подход с предварительной обработкой в критических интервалах возможен для применения в промышленных масштабах.

Литература

[1] Цыганов Н.Л., Циканин М.А. Исследование методов поиска дубликатов веб-документов с учетом запроса пользователя // Интернет-математика 2007: Сб. работ участников конкурса / Екатеренбург: Изд-во Урал. ун-та, 2007. С. 211-222.

[2] Неелова Н.В. Исследование лексического метода вычисления схожести строк веб-документа с учетом предварительной обработки // XXXV Гагаринские чтения: Научные труды Международной молодежной научной конференции в 8 томах / М.: МАТИ, 2009. 4 том, с. 31-32.

[3] Ашманов И. Оптимизация и продвижение сайтов в поисковых системах. / И. Ашманов, А. Иванов – СПб.: Питер, 2008. – 400 с.: ил.

[4] Кузнецов С.О. Порождение кластеров документов-дубликатов: подход, основанный на

поиске частых замкнутых множеств признаков / С.О. Кузнецов, С.А. Объедков, Д.И. Игнатов, М.В. Самохин

[5] Зеленков Ю.Г., Сегалович И.В. Сравнительный анализ методов определения нечетких дубликатов для WEB-документов // Труды 9ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL'2007: Сб. работ участников конкурса / Переславль-Залесский, Россия, 2007.

[6] Ильинский С. Эффективный способ обнаружения дубликатов web доку-ментов с использованием инвертированного индекса / С. Ильинский, М.Кузьмин, А.Мелков, И.Сегалович: пер. с англ. Шкредова Л., 2004 [Электронный ресурс]. – Электрон. дан. (1 файл). – http://company.yandex.ru/articles/article7.html

[7] Ukkonen E. Finding approximate patterns in strings / E.Ukkonen // Journal of Algorithms – V.6, 1985 – P. 132-137

[8] Никконен А.Ю. Устранение избыточности и дублирования сюжетов новостных сообщений // Сборник работ участников конкурса «Интернет математика 2007». – С. 153-163.

[9] Рузанов Д. Проблема дублирования контента: что есть «дубликат» для поисковых систем, 2007 [Электронный ресурс]. – Электрон. дан. (1 файл). – http://www.seonews.ru/masterclass/85/

[10] Segalovich I. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. – MLMTA, 2003 [Электронный ресурс]. – Электрон. дан. (1 файл). – http://yandex.ru/articles/iseg-las-vegas.xml

N. V. Neelova

Preliminary processing strings at critical measure Jaccard for improvement computation of web-documents similarity

The mathematical model of strings similarity computation with preliminary processing under the dictionary of synonyms at critical measure Jaccard is developed. Efficiency of the developed model is investigated. The conclusion about recall improvement of results of strings similarity calculation with an opportunity of preservation of high speed of processing is made.