УДК 629.7

СТЕММИНГ И ЛЕММАТИЗАЦИЯ В LUCENE.NET

М.В. ЖЕРДЕВА, *асп., ГБОУ ВО МО «Технологический университет»*[(1)],
В.М. АРТЮШЕНКО, *проф., ГБОУ ВО МО «Технологический университет», д-р. техн. наук*[(1)]

masha8908@rambler.ru
[(1)] ГБОУ ВО МО «Технологический университет»
141070 Московская область, г. Королев, ул. Гагарина, д. 42

В данной статье рассмотрены механизмы стемминга и лемматизации. Под стеммингом понимают приближенный эвристический процесс, в ходе которого от слов отбрасываются окончания в расчете на то, что в большинстве случаев это себя оправдает. Стемминг основан на правилах морфологии языка и не требует хранения словаря всех слов. Под лемматизацией понимается преобразование слова в словарный вид или лемму. Данный метод используется в алгоритмах поисковиков при индексировании интернет-страниц. Процесс дает возможность хранения данных страницы набором слов в индексе для удобной схематизации файлов. Это позволяет ускорить индексацию и сформировать более четкий ответ на поисковый запрос, так как сокращенную форму слова поисковик анализирует быстрее. Выделена цель стемминга и лемматизации. Показано применение стемминга и лемматизации в библиотеке полнотекстового поиска Lucene.Net. Lucene.Net – это перенесенный с платформы Java поисковый движок Lucene. Lucene – это высокопроизводительная, масштабируемая библиотека для полнотекстового поиска. Полнотекстовый поиск относится к процессу поиска документов, информации в документах или метаданных о документах. Lucene позволяет добавлять возможности поиска в различные приложения. Главной особенностью библиотеки является то, что требуется малый объем памяти, наличие ранжированного поиска, возможность одновременного поиска и обновления индекса, поиск, основанный на «полях». Lucene в настоящее время и на протяжении уже несколько лет является самой популярной свободной библиотекой полнотекстового поиска. Предложена идея модификации алгоритма полнотекстового поиска Lucene.Net для быстрого и релевантного поиска ключевых слов.

Ключевые слова: полнотекстовый поиск, стемминг, лемматизация, лемма, стоп-слова, токен.

Lucene.Net – это перенесенный с платформы Java поисковый движок Lucene. Lucene – это высокопроизводительная, масштабируемая библиотека для полнотекстового поиска.

Главной задачей Lucene.NET является решение проблемы избытка информации. С данной проблемой сталкивается любой разработчик. Lucene.NET – это высокопроизводительная масштабируемая библиотека информационного поиска. На данный момент она является самой популярной свободной библиотекой информационного поиска. Lucene.NET предоставляет простой, но очень мощный API, который требует минимума понимания механизмов текстовой индексации и поиска.

Одним из полезных средств, используемых в механизмах полнотекстового поиска, является стемминг.

Стеммингом обычно называется приближенный эвристический процесс, в ходе которого от слов отбрасываются окончания в расчете на то, что в большинстве случаев это себя оправдает. Стемминг основан на правилах морфологии языка и не требует хранения словаря всех слов. [4]

Таким образом, стемминг отсекает от слова окончания и суффиксы, чтобы оставшаяся часть была одинаковой для всех грамматических форм слова.

Более сложным подходом к решению проблемы определения основы слова является лемматизация.

Цель стемминга и лемматизации одна – привести словоформы и производные формы слова к общей основной форме.

Лемматизацией называется преобразование слова в словарный вид или лемму. Данный метод используется в алгоритмах поисковиков при индексировании интернет-страниц. Процесс дает возможность хранения данных страницы набором слов в индексе для удобной схематизации файлов. Это позволяет ускорить индексацию и сформировать более четкий ответ на поисковый запрос, так как сокращенную форму слова поисковик анализирует быстрее.

Лемма – это первоначальная, основная форма слова. Для существительных и прилагательных ею является форма единственного числа именительного падежа. Для глаголов лемма является инфинитивом, неопределен-

ной формой слова, отвечающей на вопрос в инфинитиве [1, 2, 3].

Механизм стемминга в Lucene.Net

Механизм стемминга позволяет приводить разные формы одного слова к общей форме. Например, слова «электрическая гитара» и «электрические гитары» – разные формы одного и того же слова. Без применения данного механизма они будут сохраняться в поисковом индексе в качестве разных токенов, а запросы по данным словам будут возвращать разные результаты. Если же привести их к одной форме, то проблема решается, и запросы будут возвращать один результат. Стеммер обрабатывает отдельное слово без знания контекста. Из-за этого он не может дифференцировать слова, которые имеют разные значения, т.к. они отнесены к разным частям речи. [4]

Для поддержки данной возможности в Lucene используется специальный анализатор.

Входными данными для анализатора является поток символов, представленный объектом Reader. С помощью объекта Tokenizer данный поток разбивается на последовательность токенов. После этого последовательность проходит через группу фильтров. Количество фильтров может быть произвольным и может быть нулевым. Фильтры могут добавлять новые токены, модифицировать, а также удалять существующие. Пример такого фильтра – StopFilter, позволяющий исключить из индексации лишние слова, т.е. стоп-слова. [4]

Стоп-слова – это слова, не несущие какой-либо самостоятельной смысловой нагрузки. В целях уменьшения базы данных системы не учитывают стоп-слова при индексировании, заменяя их специальным маркером. К ним относятся союзы и союзные слова, местоимения, предлоги, частицы, междометия, указательные слова, цифры, знаки препинания, вводные слова, ряд некоторых существительных, глаголов, наречий (например всегда, однако и др.).

В связи с постоянным развитием и усовершенствованием существующих алгоритмов поиска, классификации, кластеризации и пр. базы данных стоп-слов обновляются и изменяются.

Еще в качестве примера фильтра могут выступать стемминг-фильтры. Данный фильтр преобразует каждый токен к общей форме. В результате стемминг-анализатор возвращает последовательность токенов, преобразованных к общей форме.

В Lucene стемминг реализован в виде класса RussianAnalyzer. Необходимо задать язык при создании анализатора, для которого будет использоваться стемминг. Поддерживается довольно много языков, но возможности использовать многоязычный стемминг стандартная реализация не предоставляет.

Основная проблема, возникающая при использовании стеммера – это обработка слов, которые при образовании разных грамматических форм меняют не только окончание, но и основу слова.

В Lucene используется стемминг на основе алгоритма Портера. Алгоритм Портера, применяя последовательно ряд правил, отсекает окончания и суффиксы, не используя баз основ слов. Благодаря этому он работает быстро, но не всегда безошибочно.

Алгоритм состоит из пяти шагов. На каждом шаге отсекается словообразующий суффикс и оставшаяся часть проверяется на соответствие правилам (например, для русских слов основа должна содержать не менее одной гласной). Если полученное слово удовлетворяет правилам, происходит переход на следующий шаг. Если нет – алгоритм выбирает другой суффикс для отсечения. На первом шаге отсекается максимальный формообразующий суффикс, на втором – буква «и», на третьем – словообразующий суффикс, на четвертом – суффиксы превосходных форм, «ь» и одна из двух «н».

Минусы данного алгоритма:

– часто обрезает слово больше необходимого, что затрудняет получение правильной основы слова;

– не справляется с изменениями корня слова (например, выпадающие и беглые гласные) [5–10].

В Lucene стемминг реализован в виде класса RussianAnalyzer. Необходимо задать

язык при создании анализатора, для которого будет использоваться стемминг. Поддерживается довольно много языков, но возможности использовать многоязычный стемминг стандартная реализация не предоставляет.

Механизм лемматизации в Lucene.Net

Необходимо знать, как создаются различные формы слова, чтобы понимать работу лемматизации. Большинство слов изменяется, когда они используются в различных грамматических формах. Конец слова заменяется на грамматическое окончание, что приводит к новой форме исходного слова. Лемматизация выполняет обратное преобразование – она заменяет грамматическое окончание суффиксом или окончанием начальной формы.

Также лемматизация включает определение части речи слова и применение различных правил нормализации для каждой части речи. Определение части речи происходит до попытки найти основу, поскольку для некоторых языков правила стемминга зависят от части речи данного слова.

В связи с постоянным развитием и усовершенствованием существующих алгоритмов поиска, классификации, кластеризации и пр. базы данных стоп-слов обновляются и изменяются.

Высокая скорость является критерием эффективного индексирования. Она зависит от количества форм слова – чем их меньше, тем раньше закончится схематизация документа.

Словарный лемматизатор, имеющийся в грамматическом словаре, позволяет с намного более высокой точностью находить базовую форму слова из любой грамматической. Если сравнивать результат лемматизации искомого ключевого слова и слов, читаемых из текста, то получается поиск текста с учетом морфологии.

Таким образом, лемматизатор – это оптимизированный и максимально упрощенный морфологический анализатор.

Заключение

Операция лемматизации позволяет увеличить полноту и точность информационного поиска, т.к. во время операции стемминга отсутствует морфологическая обработка, из-за чего часто в выборку попадают документы, не релевантные запросу, но содержащие совпадающие формы, в то время как в релевантных документах данные слова употребляются в другой форме.

Эффективное индексирование позволяет проводить только лемматизация. В этом случае под эффективностью подразумевают скорость индексирования. Скорость индексирования зависит от количества анализируемых слов и их форм. Чем больше слов приходится обрабатывать, тем процесс индексирования идет медленнее. Лемматизатором называют программу или модуль программы, проводящие лемматизацию. Главной его задачей является то, что он улучшает релевантность поиска. Также лемматизатор уменьшает количество анализируемых слов.

Благодаря этому возникает идея использовать лемматизацию вместо стемминга.

Библиографический список

1. Нистратова, М.В. Алгоритмы поиска, используемые в Lucene.Net / М.В. Нистратова // Вестник МГУЛ –Лесной вестник – №6’2016 Том 19.
2. Нистратова, М.В. Алгоритмы поиска релевантной информации в полнотекстовых базах данных / М.В. Нистратова // Естественные и технические науки. – 2015. – № 10.
3. Нистратова, М.В. Оценка эффективности поиска документальной информации в системах единой авторизации / М.В. Нистратова, В.Г. Кулагин // Двойные технологии. – №1. – 2016.
4. Полнотекстовый поиск в веб-проектах: Sphinx, Apache Lucene, Xapian. Режим доступа: http://habrahabr.ru/post/30594/
5. Стемминг и лемматизация. Режим доступа http://delaem-krasivo.ru/programmirovanie/234-stemming-i-lemmatizaciya.html
6. Стемминг. Режим доступа: http://gruzdoff.ru/wiki/ %D0 %A1 %D1 %82 %D0 %B5 %D0 %BC - D0 %BC %D0 %B8 %D0 %BD %D0 %B3
7. Стратегии поиска и выдачи информации. Режим доступа: http://studall.org/all-130662.html
8. Тихонов, В. Поисковые системы в сети Интернет. Режим доступа: http://www.citforum.ru/internet/search/searchsystems.shtml
9. Шарапов, Р.В., Шарапова Е.В., Саратовцева Е.А., Модели информационного поиска. Режим доступа: http://vuz.exponenta.ru/PDF/FOTO/kaz/Articles/sharapov1.pdf
10. Язык запросов Lucene.NET. Режим доступа: https://pavelbelousov.wordpress.com/2011/03/23/язык-запросов-lucene-net/

STEMMING AND LEMMATIZATION IN LUCENE.NET

Zherdeva M.V., pg. «University of Technology»[(1)]; **Artyushenko V.M.,** Prof. «University of Technology», Dr Sci. (Tech.)[(1)]

masha8908@rambler.ru,
[(1)] «University of Technology», 141070 Moscow region, Korolev, ul. Gagarin, 42

In this article mechanisms stemming and lemmatization are considered. Under Stemming understand approximate heuristic process, in which the words are dropped from the end, based on the fact that in most cases it is very rewarding. Stemming is based on the morphology of the language rules, and does not require storage of a dictionary of all the words. By lemmatization understood transform words into the dictionary views or lemma. This method is used in the algorithms of search engines when indexing web pages. The process makes it possible to store the data page a set of words in the index for easy schematic files. This allows you to speed up indexing and form a clear answer to a search query as a shortened form of the word the search engine analyzes faster. The purpose of stemming and lemmatization is allocated. Application of a stemming and lemmatization in library of full text search Lucene.Net is shown. Lucene.Net is the search cursor of Lucene postponed from the Java platform. Lucene is a high-performance, scalable library for full text search. Full text search belongs to process of search of documents, information in documents or metadata on documents. Lucene allows to add features for search in various applications. The main feature of library is, the fact that the small memory size, existence of the ranged search, a possibility of simultaneous search and updating of an index, the search founded on «fields» is required. Lucene now and throughout already several years is the most popular free library of full text search. Proposed the idea of modification of algorithm of full-text search Lucene.Net for quick and relevant search engine keywords.

Key words: full text search, stemming, lemmatization, Lemma, stopwords, token.

References

1. Zherdeva, M.V. Algoritmy poiska, ispol'zuemye v Lucene.Net [Search algorithms used in lucene.net] – ZHurnal VAK «Lesnoj vestnik» [Journal VAC «Forest Bulletin»] – №6'2016 Tom 19
2. Nistratova, M.V. Algoritmy poiska relevantnoj informacii v polnotekstovyh bazah dannyh [Search algorithm relevant information in the full-text databases] – ZHurnal VAK «Estestvennye i tekhnicheskie nauki» (oktyabr' 2015, zhurnal № 10) [Journal VAC « Natural and technical science « (october, 2015, Journal No. 10)]
3. Nistratova, M. V., Kulagin V. G. Ocenka ehffektivnosti poiska dokumental'noj informacii v sistemah edinoj avtorizacii [Оценка эффективности поиска документальной информации в системах единой авторизации] – ZHurnal VAK «Dvojnye tekhnologii» [Journal VAC « Двойные технологии «]. – №1. – 2016
4. Polnotekstovyy poisk v veb-proyektakh: Sfinks, Apache Lucene, Xapian. [Full-text search in web projects: Sphinx, Apache Lucene, Xapian.] Available at: http://habrahabr.ru/post/30594/
5. Stemming i lemmatizaciya [Stemming and lemmatization]. Available at: http://delaem-krasivo.ru/programmirovanie/234-stemming-i-lemmatizaciya.html
6. Stemming [Stemming]. Available at: http://gruzdoff.ru/wiki/ %D0 %A1 %D1 %82 %D0 %B5 %D0 %BC %D0 %BC %D0 %-8 %D0 %BD %D0 %B3
7. Strategii poiska i vydachi informacii [Strategy of search and issue of information]. Available at: http://studall.org/all-130662.html
8. Tihonov, V. Poiskovye sistemy v seti Internet [Search engines on the Internet]. Available at: http://www.citforum.ru/internet/search/searchsystems.shtml
9. SHarapov, R.V., SHarapova E.V., Saratovceva E.A. Modeli informacionnogo poiska [Models of information search]. Available at: http://vuz.exponenta.ru/PDF/FOTO/kaz/Articles/sharapov1.pdf
10. Yazyk zaprosov Lucene.NET [Language of inquiries Lucene.NET]. Available at: https://pavelbelousov.wordpress.com/2011/03/23/yazyk-zaprosov-lucene-net/