ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ ПОЛНОТЕКСТОВОГО ПОИСКА ПО АННОТАЦИЯМ БИБЛИОТЕЧНЫХ ИЗДАНИЙ

RESEARCH OF FEATURES OF FULL-TEXT SEARCH BY ANNOTATIONS OF LIBRARY PUBLICATIONS

УДК 004.912

Умряев Денис Тахирович, магистрант, МГТУ им. Н. Э. Баумана, г. Москва

Umryaev D.T. udt15k074@student.bmstu.ru

Аннотация

Информационный поиск – действия, методы и процедуры, позволяющие осуществлять отбор определенной информации из массива данных. Существует два наиболее известных вида информационного поиска – полнотекстовый поиск и поиск по метаданным, но поскольку зачастую поисковые запросы имеют несколько абстрактный характер, то наибольший интерес представляет первый из них. В статье продемонстрированы ключевые особенности полнотекстового поиска на примере библиотеки Lucene. В связи с тем, что полные тексты документов зачастую имеют платный доступ по причине авторских прав издателей, особый интерес представляют аннотации, дающее общее представление о содержании текста. Тем не менее их использование при поиске не всегда осуществляется с должной эффективностью, что зачастую связано с использованием SQL-запросов на основе оператора «LIKE». Поэтому выполнено сравнение полнотекстового поиска и SQL-запросов, в основе которых лежит оператор «LIKE», при выполнении поиска по аннотациям. Полученные результаты показывают, что применение технологии полнотекстового поиска

позволяет увеличить эффективность выполнения поисковых запросов по неструктурированным и слабоструктурированным данным, в частности, по аннотациям.

Annotation

Information retrieval is a set of actions, methods and procedures that allow the selection of certain information from a data array. Two most well-known types of information search are full-text search and metadata search. Since search queries are often somewhat abstract in nature, the first one is of greatest interest. The article demonstrates the key features of full-text search using the Lucene library as an example. Annotations, giving a general idea of the text content, are of particular interest, because full texts of documents are often paid access due to the copyright of publishers. However, their use in search is not always carried out with the required efficiency, which is often associated with the use of SQL queries based on the «LIKE» operator. Therefore, a comparison of search by annotation was made between full-text search and SQL queries based on the «LIKE» operator. The results obtained show that the use of full-text search technology can increase the efficiency of search queries on unstructured and weakly structured data, in particular, on annotations.

Ключевые слова: информационный поиск, полнотекстовый поиск, поиск по аннотациям, Lucene.

Keywords: information retrieval, full-text search, search by annotations, Lucene.

Введение

Информационный поиск (англ. information retrieval) – действия, методы и процедуры, позволяющие осуществлять отбор определенной информации из массива данных [1]. Первые исследования в области информационного поиска были проведены еще в 1950 году американским специалистом в области информатики Кэлвином Мурсом [2].

Изначально информационно-поисковые системы использовались лишь для нахождения научной информации и литературы. В связи с этим данные системы

нашли большое применение у учебных учреждений и библиотек для получения доступа к книгам, учебным пособиям, журналам и другим документам. Сегодня же благодаря Интернету информационно-поисковые системы получили широкое распространение в различных областях. Наиболее известным применением являются поисковые системы по ресурсам Интернета, например, Google или Яндекс. Однако несмотря на обширное распространение информационно-поисковых систем, их использование в учебных учреждениях и библиотеках осталось одним из самых популярных.

Два наиболее известных вида информационного поиска – полнотекстовый поиск и поиск по метаданным. Полнотекстовый поиск (англ. Full-text searching) – автоматизированный поиск по документам, при котором в качестве поискового образа документа используется его полный текст или существенные части его текста [3]. Поиск по метаданным – это поиск по некоторым атрибутам документа, например, название документа, дата создания, размер, автор, различные библиографические сведения и т. д. Но поскольку зачастую поиск имеет несколько абстрактный характер, то есть заранее неизвестны ключевые параметры, которые могли бы существенно увеличить эффективность выполнения поискового запроса (например, название работы или автор), то наибольший интерес представляет полнотекстовый поиск, вследствие чего в данной области проводятся более активные исследования [4,5].

Однако применение полнотекстового поиска для анализа содержимого всего документа не всегда является доступной операцией. Это связано с тем, что полные тексты документов зачастую имеют платный доступ по причине авторских прав издателей. На решение данной проблемы приходят аннотации, которые обычно дают общее представление о содержании текста. В случае научных работ они содержат постановку проблемы или исследуемого вопроса, используемые методы исследования (эксперимент, тематическое исследование, анкетирование и др.), результаты исследования, а также общие выводы и рекомендации. Таким образом,

аннотации являются основой для поиска исследований, научных работ, книг и других документов.

Поиск по аннотациям является важной разновидностью полнотекстового поиска для автоматизированных библиотечных систем. Тем не менее в этих системах часто используются поисковые возможности сервера СУБД. В этом случае эффективность выполнения поисковых запросов может оказаться очень низкой, что зачастую связано с использованием запросов к реляционной базе данных, в основе которых лежит SQL-оператор «LIKE».

Таким образом, цель данного исследования – продемонстрировать ключевые особенности полнотекстового поиска, а также сравнить два способа осуществления поиска по аннотациям – с применением технологии полнотекстового поиска, а также с помощью запросов к реляционной базе данных, в основе которых лежит SQL-оператор «LIKE».

Для проведения исследования необходимо два ключевых компонента: во-первых, система, предоставляющая возможности для полнотекстового поиска, во-вторых, реляционная СУБД, позволяющая строить запросы к базе данных.

Несмотря на то, что сегодня СУБД обладают функционалом для полнотекстового поиска (например, MS SQL Server, MySQL, PostgreSQL [6]), всё же в данном исследовании рассмотрим отдельную систему в лице Lucene, зарекомендовавшую себя как одну из самых продвинутых и эффективных. А в качестве реляционной СУБД возьмем Microsoft SQL Server, разработанную корпорацией Microsoft. Основным языком запросов в ней используется Transact-SQL, созданный Microsoft совместно с Sybase.

Lucene

Lucene – это высокопроизводительная библиотека для полнотекстового поиска, полностью написанная на Java. Типичное приложение с использованием Lucene состоит из двух элементов: *indexer* и *searcher*.

Indexer необходим для создания индекса. Здесь указывается анализатор, который будет выполнять предобработку текста (токенизация, удаление стоп-слов и

пунктуации, перевод в нижний регистр, стемминг/лемматизация [7]), а также определяется, где будет хранится индекс, в памяти или в файловой системе. Единицей данных в индексе является документ – именованный набор полей с текстовым значением. В Lucene используется инвертированный индекс для организации поиска. Если в классическом индексе для каждого документа указывается полный список слов, содержащихся в нем, то в инвертированном индексе для каждого слова из коллекции документов перечисляются все те документы, в которых данное слово встретилось.

Searcher, в свою очередь, необходим для построения запроса, и осуществления поиска по данному запросу с последующей выдачей ранжированных результатов. Lucene предоставляет довольно богатые возможности для построения запроса. Так, например, возможен поиск с использованием булевых операторов (AND, OR и NOT), поиск по фразе (в кавычках), поиск нечетких соответствий (в случае наличия в документе грамматических ошибок), поиск с использованием подстановочных знаков (например, поиск слов отличающихся в одной букве в середине), поиск с использованием регулярных выражений, поиск с диапазоном, а также поиск с учетом расположения (позволяет найти слова, расположенные рядом в документе).

Ранжирование результатов поиска в Lucene по умолчанию проводится по релевантности. Для этого в Lucene используется специальная формула, в основе которой лежит популярная TF-IDF мера [8]. TF-IDF (от англ. TF – term frequency, IDF – inverse document frequency) – это числовая статистика, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса [9]. Вес слова прямо пропорционален частоте использования этого слова в документе и обратно пропорционален частоте использования этого слова во всех документах коллекции.

Формула TF-IDF состоит из двух отдельных статистик – TF и IDF. Существуют различные способы определения значений каждой из этих статистик,

отличающиеся коэффициентами, нормировками и использованием логарифмированных шкал.

TF (term frequency – частота слова) – отношение количества использований некоторого слова в документе к общему количеству слов в этом документе. Таким образом, оценивается важность слова t в отдельном документе d. TF в базовом варианте имеет следующий вид (Формула 1).

$$TF(t,d) = \frac{n_t}{\sum_k n_k} \quad (1)$$

где n_t – количество использований слова t в документе d;

$\sum_k n_k$ – количество слов в документе d.

IDF (inverse document frequency – обратная частота документа) – инверсная частота, с которой некоторое слово встречается в документах коллекции. IDF позволяет уменьшить вес часто используемых слов. Базовый вариант IDF имеет вид (Формула 2).

$$IDF(t,D) = log\frac{|D|}{|\{d_i \in D | t \in d_i\}|} \quad (2)$$

где $|D|$ – количество документов в коллекции;

$|\{d_i \in D | t \in d_i\}|$ – количество документов в коллекции D, в которых хотя бы раз встречается слово t (т.е. $n_t \neq 0$).

В итоге, TF-IDF есть произведение двух сомножителей (Формула 3).

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (3)$$

В данной статистике большой вес имеют слова, которые часто встречаются в пределах конкретного документа и при этом редко встречаются в других документах.

Результаты

Для проведения исследования был подготовлен датасет с 20000 аннотациями на русском языке, но с использованием слов на английском языке (в основном непереводимые названия чего-либо, например, языков программирования или программ). В качестве анализатора Lucene был выбран морфологический

анализатор библиотеки RussianMorphology [10], количество выдаваемых результатов при поиске неограниченно. Выполнение запросов к базе данных осуществлялось из Java приложения через JDBC драйвер, в основе запросов лежал SQL-оператор «LIKE». Проведение тестов осуществлялось на компьютере со следующими характеристиками: Intel Core i5, 4 ядра, 3200 МГц; 16 Гб 2133 МГц DDR4; 240 Гб SSD.

Для сравнения результатов поиска был подготовлен набор из 60 запросов, разделенный на 4 группы: простые запросы с одним словом, запросы с аббревиатурой или непереводимым названием чего-либо на латинице (1 слово), короткие запросы (2 слова), а также длинные запросы (3 и более слов).

Оценивались две главные характеристики: время поиска и количество выданных результатов. Для более достоверной оценки времени поиска было проведено 10 итераций выполнения каждого запроса, а затем на основе этого высчитывалось среднее значение в группе. Количество выданных результатов также оценивалось, как среднее значение в каждой группе несмотря на то, что значения выдаваемых результатов среди одной группы могли заметно отличаться, в данном случае большой разброс существенного влияния не имеет. Результаты тестов приведены в табл. 1.

Таблица 1 – Результаты тестов

	Lucene	MS SQL Server
Простые запросы с одним словом		
Время (мс)	4	774
Кол-во результатов	291	141
Запросы с аббревиатурой или названием на латинице (1 слово)		
Время (мс)	2	776
Кол-во результатов	46	52
Короткие запросы (2 слова)		
Время (мс)	14	774
Кол-во результатов	1874	6
Длинные запросы (3 и более слов)		
Время (мс)	319	773
Кол-во результатов	9656	0

Анализ результатов

Как видно из табл. 1 полнотекстовый поиск с помощью библиотеки Lucene имеет заметное преимущество по времени над поиском с помощью SQL-запросов с применением оператора «LIKE». Особенно заметна разница в коротких запросах (один-два слова), где Lucene требуется на выполнение запроса не более нескольких десятков миллисекунд, в то время как запросы к базе данных выполняются почти целую секунду. Достаточно короткое время поиска в Lucene достигается за счет использования индекса, и соответственно вместо того, чтобы анализировать текст напрямую, Lucene анализирует индекс. MS SQL Server, напротив, проходит по каждой записи в базе данных, что в итоге отражается на длительности времени поиска. Кроме того, в MS SQL Server время поиска не зависит от длины запроса – оно одинаково для каждой группы запросов. Однако стоит отметить, что с повышением количества слов запроса, разница по времени уже не оказывается такой большой, хотя все еще остается ощутимой. Но поскольку в Lucene результаты ранжируются по релевантности, то существует возможность ограничить их количество при выводе (ведь результаты с низкой релевантностью особого интереса не представляют), а это в свою очередь уменьшает и так низкое время поиска.

С количеством выданных результатов ситуация не такая однозначная. MS SQL Server неплохо справляется с запросами, состоящими всего из одного слова. Здесь ключевую роль играет простота запроса, ведь SQL-оператор «LIKE» просто анализирует вхождение строки запроса в текст аннотации. Поэтому в запросах с одним словом MS SQL Server выдает достаточно приемлемое количество результатов. Более того, в запросах с аббревиатурой или названием чего-либо на латинице MS SQL Server выдает даже большее количество результатов, нежели полнотекстовый поиск Lucene. В данном случае немалое значение имеет специфика запросов из этой группы, ведь каждый запрос представляет из себя слово, которое в русском языке обычно не склоняется (склонение аббревиатур в русском языке хоть и допускается, однако всё же зачастую этого не делается, а

слова на латинице не склоняются всегда). Но основная же причина, почему MS SQL Server выдал большее количество результатов, чем Lucene заключается в том, что некоторые слова из данной группы запросов являются составной частью других слов, так, например, при поисковом запросе «СУБД» MS SQL Server выдавал нерелевантную аннотацию со словом «субдифференциальный». С двумя и более словами в запросе MS SQL Server справляется уже заметно хуже, выдавая в редких случаях всего пару результатов, что очевидно, ведь необходимо полное вхождение строки запроса в текст аннотации. Lucene же наоборот с увеличением слов в запросе выдает только большее количество результатов. Этому способствуют различные предобработки текста, такие как токенизация, удаление стоп-слов и пунктуации, приведение слов к нормальной форме и т.д.

Заключение

Системы полнотекстового поиска отлично подходят для быстрого поиска в больших объемах неструктурированных или слабоструктурированных данных по определенному слову или целому предложению. Кроме того, данные системы выполняют сложное ранжирование по релевантности для упорядочения результатов по тому, насколько хорошо они соответствуют поисковому запросу. Данные системы также обеспечивают различные формы обработки естественного языка для понимания намерений пользователя. Эти возможности позволяют поисковым системам иметь гораздо более интуитивно понятный и персонализированный интерфейс для пользователя. Реляционные СУБД, наоборот, отлично справляются с хранением, выборкой и манипулированием структурированных данных – всего того, что находится в табличном формате строк и столбцов.

Полученный результат показывает, что применение технологии полнотекстового поиска позволяет увеличить эффективность выполнения поисковых запросов по аннотациям, а также другим подобным запросам. В частности, применение Lucene позволило снизить время поиска на 50% в случае длинных запросов.

Литература

1. Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск // Вильямс. 2011. 512 с.
2. К. Мурс. Поиск информации, рассматриваемый как временная сигнализация, в трудах Международного конгресса математиков. 1950. Том 1. С. 572–573.
3. ГОСТ 7.73-96 СИБИД. Поиск и распространение информации. Термины и определения. Москва. Изд-во стандартов. 1997. 16 с.
4. Веретенников А.Б. Эффективный полнотекстовый поиск с использованием дополнительных индексов часто встречающихся слов // Системы управления и информационные технологии. 2016. № 4 (66). С. 52-60.
5. Веретенников А.Б. Эффективный полнотекстовый поиск с учетом близости слов при помощи трехкомпонентных ключей // Системы управления и информационные технологии. 2017. № 3 (69). С. 25-32.
6. Виноградова М.В., Барашкова Е.С., Березин И.С., Ореликов М.Г., Лузин Д.С. Обзор системы полнотекстового поиска в постреляционной базе данных PostgreSQL // E-Scio. 2020. № 5 (44). С. 754-778.
7. Жердева М.В., Артюшенко В.М. Стемминг и лемматизация в LUCENE.NET // Вестник Московского государственного университета леса - Лесной вестник. 2015. Т. 19. № 6. С. 168-173.
8. Документация Apache Lucene 8.8.1. Классическая формула оценки [Электронный ресурс]. URL: https://lucene.apache.org/core/8_8_1/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html (дата обращения: 04.03.2020).
9. Раджараман А., Уллман Дж. Д. Майнинг массивных наборов данных. URL: http://i.stanford.edu/~ullman/mmds/ch1.pdf. doi:10.1017/CBO9781139058452.002 (дата обращения: 04.03.2020).
10. Русская морфология для Apache Lucene [Электронный ресурс]. URL: https://github.com/AKuznetsov/russianmorphology (дата обращения: 04.03.2020).

Literature

1. Manning C., Raghavan P., Schütze H. An Introduction to Information Retrieval // Williams. 2011. 512 p.
2. C. Mooers. Information retrieval viewed as temporal signaling, in Proc. of the International Congress of Mathematicians. 1950. Vol. 1. P. 572–573.
3. GOST 7.73-96 SIBID. Search and dissemination of information. Terms and definitions. Moscow. Standards Publishing House. 1997. 16 p.
4. Veretennikov A.B. Effective full-text search using additional indexes of frequently occurring words / / Management systems and information technologies. 2016. № 4 (66). P. 52-60.
5. Veretennikov A.B. Effective full-text search taking into account the proximity of words with the help of three-component keys / / Management systems and information technologies. 2017. No. 3 (69). P. 25-32.
6. Vinogradova M.V., Barashkova E.S., Berezin I.S., Gorelikov M.G., Luzin D.S. Review of the full-text search system in the PostgreSQL post-relational database. 2020. № 5 (44) // E-Scio. P. 754-778.
7. Zherdeva M.V., Artyushenko V.M. Stemming and lemmatization in LUCENE.NET // Bulletin of the Moscow State University of the Forest - Forest bulletin. 2015. Vol. 19. No. 6. P. 168-173.
8. Apache Lucene 8.8.1 Documentation. Classic Scoring Formula [Electronic resource]. URL: https://lucene.apache.org/core/8_8_1/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html (дата обращения: 04.03.2020).
9. Rajaraman A., Ullman J.D. Mining of Massive Datasets. URL: http://i.stanford.edu/~ullman/mmds/ch1.pdf. doi:10.1017/CBO9781139058452.002 (accessed: 04.03.2020).
10. Russian Morphology for Apache Lucene [Electronic resource]. URL: https://github.com/AKuznetsov/russianmorphology (accessed: 04.03.2020).