DOI 10.37882/2223–2966.2021.04.36

ИСПОЛЬЗОВАНИЕ APACHE SOLR ДЛЯ АНАЛИЗА ДАННЫХ

USING APACHE SOLR FOR DATA ANALYSIS

S. Khrapov

Summary. Apache Solr is an open source, highly scalable full-text search engine. This allows you to store, search and analyze large amounts of data quickly and in near real time. This is typically used as the underlying technology mechanism that enables applications that have complex search functionality and requirements. Such a system is perfect for solving the problem of information retrieval in a large storage of text data. The paper discusses the language models used in Solr, which provide fast search for documents of different sizes and formats.

Keywords: Apache solr, information retrieval, text processing, search relevance, language models.

Храпов Сергей Дмитриевич
Аспирант, ГБОУ ВО Московской области «Технологический университет», г. Королев
hrapov123@mail.ru

Аннотация. Apache Solr — это легко масштабируемая система полнотекстового поиска с открытым исходным кодом. Это позволяет хранить, искать и анализировать большие объемы данных быстро и практически в реальном времени. Обычно это используется в качестве базового механизма технологии, которая обеспечивает работу приложений, имеющих сложные функции и требования поиска. Такая система отлично подойдет для решения проблемы информационного поиска в большом хранилище текстовых данных. В работе рассмотрены языковые модели, используемые в Solr, которые обеспечивает быстрый поиск по документам разных размеров и форматов.

Ключевые слова: Apache solr, информационный поиск, обработка текстовой информации, релевантность поиска, языковые модели.

Solr — это распределенный механизм полнотекстового поиска и анализа, построенный на основе Lucene, библиотеки поисковых систем, написанной на Java. После первого выпуска в 2007 году Solr получил широкое распространение как в крупных, так и в небольших организациях, для различных вариантов использования. В последних выпусках Solr больше внимания уделяется отказоустойчивости, что повышает уверенность пользователей в возможности использовать Solr в качестве инструмента хранения данных, а не только в качестве механизма полнотекстового поиска [1].

С постоянно развивающимися и растущими объемами данных организациям необходимо находить полезные технологии для своего бизнеса. Solr дает возможность сделать огромные объемы разнородных, неиндексированных данных пригодными для использования. Это не только дает возможность создавать рабочие поисковые решения для огромного объема данных, но также может служить хранилищем данных NoSQL. Потребители, которые ищут информацию о продукте из каталогов веб-сайтов, зачастую сталкиваются с такими проблемами, как длительный поиск информации о продукте. Это приводит к плохому пользовательскому опыту и, в свою очередь, к отсутствию потенциального клиента. Сегодня бизнес ищет альтернативные способы хранения большого объема данных таким образом, чтобы поиск был быстрым.

Этого можно достичь, приняв NOSQL вместо RDBMS (система управления реляционными базами данных) для хранения данных. Solr использует NOSQL, потому что:

- это просто в использовании;
- имеет большое сообщество;
- имеет совместимость с JSON;
- широкие варианты использования [2].

Эта технология является отличным решением для информационного поиска, так как в ней используются различные поисковые алгоритмы. В Solr по умолчанию используется метод оценки BM25, однако есть и другие модели поиска, которые показывают высокую производительность.

Языковые модели широко используются в области обработки естественного языка (NLP) для задач распознавания и генерации речи. Языковая модель для данной строки сообщает вам о вероятности появления строки в языке. Для языка с определенным словарем языковая модель представляет распределение вероятностей терминов в словаре [3]

Например, есть простой язык со следующей лексикой: {галактика, космос, звезда, спутник, орбита}

Языковая модель для этого языка может иметь распределение вероятностей, показывающее вероятность

Таблица 1. Распределение вероятностей

Языковая модель: M	
Галактика	0.4
Космос	0.3
Звезда	0.1
Спутник	0.1
орбита	0.1

появления различных словарных терминов, представленное в таблице 1.

Таким образом, вероятность того, что модель M сгенерирует термин «галактика», равна 0,4.

Существуют различные типы языковых моделей. Самым простым (представленным выше) является модель языка Unigram. Модель языка Unigram предполагает, что термины встречаются независимо друг от друга. Для нашей модели это будет означать, что «галактика» в документе не влияет на вероятность появления термина «звезда» в том же документе. Хотя это не всегда так, и многие термины, как правило, встречаются вместе. Часто это достаточно хорошее приближение для поиска информации.

Подход к использованию языкового моделирования для поиска документов заключается в следующем. Для каждого документа в нашей коллекции мы создаем отдельную языковую модель и ранжируем документы по вероятности их генерации заданного запроса [4]. Точнее, сначала мы строим отдельную языковую модель для каждого документа в коллекции:

$$doc_1 -> M_{d1},$$
$$doc_2 -> M_{d2},$$
$$doc_n -> M_{dn},$$

Чтобы найти наиболее подходящие документы для данного запроса q, для каждого документа d в коллекции мы оцениваем вероятность соблюдения q в d. Или, другими словами, мы оцениваем, какова вероятность того, что языковая модель M_d документа сгенерирует q: $P(q \mid M_d)$. Затем мы ранжируем документы по этим оценкам вероятностей. Стоит отметить, что исходная проблема, заключающаяся в том, чтобы узнать, является ли документ релевантным для запроса, в подходе к языковому моделированию сводится к вычислению вероятности того, что запрос извлечен из документа.

Логика, лежащая в основе этого подхода, заключается в следующем: когда пользователь нуждается в информации, он представляет, какие документы удовлетворят эту потребность; он строит в своем воображении модель этих документов, визуализируя, какие термины являются репрезентативными для этих документов. Основываясь на этих воображаемых моделях документов и их представительных терминах, пользователь формулирует запрос, предсказывая, что этот запрос должен генерировать нужные ему документы. Во время поиска процесс обращается вспять. Для каждого документа оценивается вероятность того, что пользователь будет использовать запрос, чтобы найти этот документ. Другими словами, какова вероятность того, что эта модель документа ответит на данный запрос [5]?

Если у нас есть запрос q, состоящий из одного термина t, как мы можем рассчитать вероятность того, что языковая модель документа Md сгенерирует термин t? Самый простой способ оценки вероятности основан на подсчете этого термина, фигурирующего в документе [5]:

$$P(t|M_d) = \frac{tf_{t,d}}{L_d}, \quad (1)$$

где t — термин в запросе

d — документ

M_d — языковая модель документа

$tf_{t,d}$ — частота появления термина t в документе d, сколько раз t встречается в d

L_d — длина документа d — общее количество терминов, которые d содержит

$P(t \mid M_d)$ — вероятность возникновения термина t, заданного M_d.

Таким образом, чем чаще t появляется в d, тем выше вероятность того, что t можно использовать для получения d. Обратите внимание, что для терминов, не фигурирующих в документе, эта вероятность равна 0. Если запрос состоит из нескольких терминов, предполагается, что термины не зависят друг от друга (униграммная модель), и, таким образом, для вычисления вероятности для всего запроса нам нужно умножить вероятности отдельных терминов:

$$P(q|M_d) = \Pi_{t \in q} P(t|M_d) = \Pi_{t \in q} \frac{tf_{t,d}}{L_d}, \quad (2)$$

где q — запрос, состоящий из одного или нескольких терминов t.

Приведенная выше формула называется моделью вероятности запроса, так как она оценивает вероятность запроса, заданного документом, или то, насколько хорошо документ d соответствует конкретному запросу q [4, 1].

Приведем пример расчета вероятностей с использованием формулы (2) для запроса q, состоящего из двух терминов «космос» и «земля» для следующей коллекции из трех документов:

d_1: «Каков космос? Обитаем ли космос? Галактика наполнена планетами, и где-то должна быть такая, как Земля.»

d_2: «Первая фотография Земли из космоса была получена 24 октября 1946.»

d_3: «Земля — единственная известная в настоящий момент планета, на которой есть жизнь.»

Для документа d_1 мы видим, что tf («космос») равно 2, потому что есть 2 вхождения термина, и L_d равно 15, потому что d_1 имеет длину 15 терминов. Для второго термина tf («земля») равен 1 для этого документа. Таким образом, мы умножаем их вместе, чтобы получить 0,0089:

$$P(q|M_{d1}) = \frac{2}{15} \times \frac{1}{15} = 0.0089 \quad (3)$$

Аналогично для документов d_2 и d_3

$$P(q|M_{d2}) = \frac{1}{30} \times \frac{1}{30} = 0.0011 \quad (4)$$

$$P(q|M_{d1}) = \frac{0}{11} \times \frac{1}{11} = 0 \quad (5)$$

Если необходимо ранжировать эти документы, документ d_1 с наивысшей оценкой будет иметь самый высокий рейтинг, а документ d_3 с наименьшей оценкой — самый низкий. Чтобы повысить точность этих моделей документов, нам нужно сгладить их с помощью фоновой модели. Модель для всей коллекции документов может служить хорошей фоновой моделью, поскольку она содержит не один, а несколько документов — коллекцию моделей документов M_c.

Общая идея сглаживания заключается в корректировке модели документа Md на основе модели коллекции M_c, что делает Md немного похожим на M_c путем переноса некоторой массы документов из Mс в Md [5]. Это позволяет нам повысить точность модели, а также избежать нулевых вероятностей для отсутствующих слов.

Настройка сглаживания

Для терминов, которых нет в M_d, мы присваиваем некоторую вероятность, которая будет частью вероятности M_c. Это предотвратит генерацию в M_d нулевых вероятностей для отсутствующих слов. Для терминов, присутствующих в M_d, мы скомбинируем обе вероятности из M_d и часть из M_c, чтобы исключить первоначальную вероятность из M_d. Это должно улучшить общую точность модели документа.

В Solr представлены два метода — сглаживание Джелинека-Мерсера и сглаживание Дирихле.

Рассмотрим сглаживание на примере метода Дирихле.

Логика, лежащая в основе сглаживания Дирихле, состоит в том, что к каждому документу в коллекции добавляется μ терминов и они распределяются в соответствии со статистикой всей коллекции [5]. Например, если μ = 100, добавляется 5,357 терминов «космос» (100 * 3/56) к документу d_3 из предыдущего раздела. С этим и другими терминами длина документа $d3$ будет увеличена на 100. Конечно, в действительности, в документы не будут добавлены какие-либо термины, они используются для вычисления распределения вероятности, используя следующую формулу:

$$P(t|d)\frac{tf_{t,d} + \mu P(t|M_{c)}}{L_d + \mu}, \quad (6)$$

где μ — мю, значение > 0;

$tf_{t,} d$ — частота появления термина t в документе d, сколько раз t встречается в d;

Ld — длина документа d — общее количество терминов в d;

M_c — языковая модель для всей коллекции.

Для запроса, состоящего из нескольких терминов, используется следующая формула:

$$P(q|d)\prod_{t\in q} P(t|d) = \prod_{t\in q}\frac{tf_{t,d} + \mu P(t|M_c}{L_d + \mu}, \quad (7)$$

Выбор μ должен зависеть от длины документа. Чем длиннее документ, тем большее значение μ должно быть для заполнения воздействия.

μ — значение по умолчанию 2000 в Solr и Lucene.

Вот пример расчета баллов по формуле (7). Воспользуемся той же коллекцией документов. Примем μ = 100, так как используемые документы небольшие

tf («космос»): 3, общее количество случаев появления термина «космос во всех документах.

tf («земля»): 3, общее количество появления термина «земля» во всех документах.

L_c: 56, всего в коллекции 56 терминов. Из этого имеем:

$$P(q|M_{d1}) = \frac{2+100\times\frac{3}{56}}{15+100} \times \frac{1+100\times\frac{3}{56}}{15+100} = 0.00354,$$

$$P(q|M_{d2}) = \frac{1+100\times\frac{3}{56}}{30+100} \times \frac{1+100\times\frac{3}{56}}{30+100} = 0.0024,$$

$$P(q|M_{d1}) = \frac{0+100\times\frac{3}{56}}{11+100} \times \frac{1+100\times\frac{3}{56}}{11+100} = 0.00276.$$

Можно убедиться, что нулевое значение для Md_3 исчезло.

Сглаживание Дирихле — хороший выбор для многих задач информационного поиска. Как и во всех методах сглаживания, он никогда не назначает нулевую вероятность появления термина.

- Это байесовский апостериор, который учитывает, чем документ отличается от коллекции.
- Нормализуется по длине документа, что позволяет оценить документы абсолютно разной длины.
- Работает быстрее многих других техник сглаживания. Использование языковой модели с применением сглаживания Дирихле отлично подойдет для решения проблемы информационного поиска в корпоративном хранилище.

В работе было рассмотрено применение Solr для информационного поиска в большом объеме данных. В частности, был исследован метод информационного поиска, основанный на языковых моделях. Было обнаружено что производительность поиска обычно чувствительна к сглаживанию параметров. Для сглаживания параметров хорошо подходит метод Дирихле, так как он более гибок для текстов(документов) разных размеров.

ЛИТЕРАТУРА

1. Информация о версиях Solr [Электронный ресурс]: Информация о версии Solr 8.2/ URL: https://cwiki.apache.org/confluence/display/SOLR/Solr8.2 (Дата обращения: 15.12.2020)
2. Apache Solr — Основы поисковой системы [Электронный ресурс]: Apache Solr — Основы поисковой системы / https://coderlessons.com/tutorials/bolshie-dannye-i-analitika/uznaite-apache-solr/uchebnik-po-apache-solr (Дата обращения: 15.12.2020)
3. Кумар Ш., Шукла П. Elasticsearch, Kibana, Logstash и поисковые системы нового поколения. СПб.: Прогресс книга, 2019. 352 с.
4. Croft W. B., Lafferty J. Language Modeling for Information Retrieval. Springer Science & Business Media. 2003. 246 p.
5. Zhai Ch., Lafferty J. A study of smoothing methods for language models applied to Ad Hoc information retrieval // In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '01). ACM. New York. USA. P. 334–342.

Журнал «Современная наука: актуальные проблемы теории и практики»