# АЛГОРИТМЫ ПОИСКА, ИСПОЛЬЗУЕМЫЕ В LUCENE.NET

М.В. ЖЕРДЕВА, *асп., Технологический университет*[1]

*masha8908@rambler.ru*

[1]ГБОУ ВО МО «Технологический университет», 141074, Московская обл., г. Королев, ул. Гагарина, д. 42

В статье рассмотрены модели поиска, положенные в основу работы Lucene.Net, и описаны особенности ранжирования документов. Одной из важнейших становится задача поиска по содержимому за конечное время в большом объеме документов. Традиционные системы поиска, как правило, ориентируются на работу со структурированными текстовыми данными и мало приспособлены для обработки мультимедийной и неструктурированной информации. Тогда возникает проблема поиска и выборки необходимой информации из большого неструктурированного массива. Одним из факторов, стимулирующих развитие технологий поиска, является появление огромного количества электронных библиотек, содержащих значительные объемы актуальных знаний. В связи с тем, что выбор алгоритма поиска зависит от особенностей проекта, необходима разработка более совершенных методов, берущих за основу приведенные модели и обеспечивающих высокую релевантность найденных документов по исковому запросу пользователя за возможно более короткие сроки, а также обладающие точно вычисляемым сроком выдачи результата. Требуются особые виды поиска и обработки результата, а также особое количество или формат данных в проекте. В данной статье выделены параметры, которые следует учитывать при выборе поискового механизма. Проанализированы существующие подходы к решению задач поиска и предложено их улучшение, основанное на использовании модификации булевого поиска (метод взвешенного зонного ранжирования). Приведены критерии оценки информационного поиска. Показана концептуальная формула для оценки релевантности поиска Lucene.Net.

Ключевые слова: информационный поиск, документ, критерии, релевантность, поисковые системы

В настоящее время происходит повсеместный переход предприятий на электронный документооборот. Благодаря этому большое значение приобретают системы «мгновенного» локального полнотекстового поиска по содержимому документов различных типов. Количество документов, с которыми сотрудник должен работать за день, увеличивается с каждым годом, поэтому особенно важной становится задача поиска по содержимому за конечное время в большом объеме документов (десятки и сотни тысяч документов). Традиционные системы поиска, развивающиеся в тесной взаимосвязи с СУБД, в основном ориентированы на работу со структурированными текстовыми данными и мало приспособлены для обработки мультимедийной и неструктурированной информации. В этом случае возникает проблема поиска и выборки необходимой информации из большого неструктурированного массива. Другим фактором, стимулирующим развитие технологий поиска, является появление большого количества электронных библиотек, содержащих значительные объемы актуальных знаний. Производительность и эффективность любой подобной системы хранения информации напрямую зависит от эффективности и производительности поисковых систем.

**Актуальность** вопросов разработки поисковых систем. На сегодняшний момент к поисковым системам предъявляется ряд дополнительных требований: построение запроса на естественном языке, поиск информации не только по формально заданным терминам, но и автоматический анализ и расширение запроса, возможность создания сложных запросов с итеративным и интерактивным уточнением его параметров, интеллектуальное ранжирование выдаваемой информации. Полноценного решения совокупности указанных задач в данный момент не найдено. Для поиска информации на основании запроса на естественном языке сначала необходимо построить модель знаний пользователя о предметной области. Данная модель строится на основании вводимого пользователем текста запроса с использованием алгоритмов семантико-синтаксического анализа. Построение подобной модели позволяет более точно указать, что именно ищет пользователь. Для расширения границ поиска с целью охвата всей предметной области служат различные алгоритмы, связанные с применением тезауруса. Знания о языке в целом и предметной области в частности представляются в виде фиксированной (иногда динамичной) семантической сети с выделенными отношениями между поня-

тиями (терминами). Большинство ПС после осуществления поиска выполняет ранжирование списка найденных документов. Ранжирование осуществляется путем оценки релевантности документов запросу пользователя. Способы оценки релевантности, применяемые большинством ПС, основаны на сравнении количества совпадающих слов в запросе и на странице с учетом местонахождения слов на странице. Этот подход не всегда дает хороший результат, особенно если запрос был сложным и содержал многозначные понятия. В этом случае целесообразно применять оценку релевантности запроса с использованием аппарата нечеткой логики для сравнения семантических сетей запроса и документа. Подобный подход оценивает содержимое не по формальному наличию искомых слов из запроса, а по смысловому содержанию страницы и его соответствию смысловому содержанию запроса.

**Задачи**, возлагаемые на поисковую систему (ПС), зависят от роли и места ПС в информационной системе. Это может быть подсистема поиска, интегрированная внутрь СУБД, либо стороннее средство, налагаемое на существующий архив информации. Одной из задач, возлагаемых на ПС, является поиск и анализ информации в средствах электронного документооборота. Одно из решений заключается в том, что информация загружается в специальную «систему управления знаниями». В данном случае информация хранится внутри системы и доступ к ней осуществляется средствами специализированных поисковых средств. Другим является подход, при котором на сеть источников накладывается единая информационно-поисковая система, обеспечивающая прозрачный поиск и категоризацию документов. Самой распространенной задачей, ставящейся перед ПС, является задача поиска информации в предварительно проиндексированных полнотекстовых массивах данных. Это могут быть как данные на локальной машине, так и распределенные данные внутри Интранет/Интернет сетей. Подобная задача поиска стоит как перед поисковыми системами Интернет, так и перед специализированными средствами полнотекстового поиска. Выделяются следующие подзадачи: поиск по контексту, тематический поиск, построение карты знаний, авторубрикатор, поиск документов по отношению близости.

### Параметры критериев выбора поискового механизма

В качестве критериев выбора поискового механизма определим следующие параметры:

– скорость индексирования и переиндексации,

– поддерживаемые API,

– поддерживаемые протоколы,

– размер базы и скорость поиска,

– поддерживаемые типы документов,

– работа с разными языками и стемминг,

– поддержка дополнительных типов полей в документах,

– платформа и язык,

– возможность расширения встроенных механизмов ранжирования и сортировки [3–6].

### Основные принципы определения релевантности

1. Количество ключевых слов запроса в тексте документа.

2. Тэги, в которых эти слова располагаются.

3. Местоположение искомых слов в документе.

4. Удельный вес слов, относительно которых определяется релевантность, в общем количестве слов документа.

5. Время – как долго страница находится в базе поискового сервера.

6. Индекс цитируемости – как много ссылок на данную страницу ведет с других страниц, зарегистрированных в базе поисковика [5].

### Критерии оценки информационного поиска

Рассмотрим координаты описания выхода АИПС с точки зрения потребителя информации [1, 2, 7]:

– диаграмма <*L*> или диаграмма Эйлера-Венна (рис. 1);

– таблица сопряженности или диаграмма <*a, b, c, d*> (рис. 2);

– диаграмма <*n*, *x*>– взаимосвязь числа выданных релевантных документов и общего числа выданных документов (рис. 3).

***Первичные координаты.*** Относительно координат <*n, x*> необходимо заметить, что допустимые выдачи (имеющие смысл сочетания числа выданных релевантных − *x* и общего числа выданных документов − *n*) находятся в незаштрихованной области 0Ир0Д, ограниченной прямыми линиями

$$\text{ОИ: } x=n;\ \text{Ир0: } x=x0;$$
$$\text{р0Д: } x = n - (n0 - x0);\ \text{Д0: } x = 0. \quad (1)$$

Очевидна следующая взаимосвязь перечисленных координат:

● число выданных релевантных документов

$$a = x = |L^{И} \cap L^{C}|, \quad (2)$$

● количество выданных документов

$$a + b = n = |L^{C}|, \quad (3)$$

● общее число релевантных

$$a + c = x0 = |L^{И}|, \quad (4)$$

● общее число документов

$$L0 - a + c + d = n0 = |L_0|. \quad (5)$$

***Частные критерии.*** В целях количественного описания уровня качества поиска АИПС исторически первыми были предложены следующие частные критерии оценки системы:

● полнота (*r* − от *recall ratio*):

$$r = \frac{a}{a+c} = \frac{x}{x_0} = \frac{\left|L^{И} \cap L^{C}\right|}{|L^{И}|}; \quad (6)$$

● точность (*p* − от *precision ratio*):

$$p = \frac{a}{a+b} = \frac{x}{n} = \frac{|L^{И} \cap L^{C}|}{|L^{C}|}; \quad (7)$$

● специфичность:

$$\sigma = \frac{d}{d+b} = 1 - \frac{n-x}{n_0 - x_0} = \frac{|L_0(L^{C} \cup L^{C})|}{|L_0 L^{C}|}; \quad (8)$$

● общность (точность *L*0) характеризует качество поискового массива в целом:

$$p_0 = \frac{a+c}{a+b+c+d} = \frac{x_0}{n_0} = \frac{|L^{И}|}{|L_0|}. \quad (9)$$

Каждая из переменных (6) − (8) изменяется в пределах от 0 до 1.Список перечисленных переменных может быть дополнен величиной *n* (относительный объем выдачи):

$$v = \frac{a+b}{a+b+c+d} = \frac{n}{n_0} = \frac{|L^{c}|}{|L_0|}. \quad (10)$$

Библиотека Lucene.NET, берущаяся за основу данного исследования, является продуктом с открытым исходным кодом, распространяемым по лицензии Apache. Она является мощным и производительным средством полнотекстового поиска, поддерживающим подключение различных моделей ранжирования степени релевантности полученных результатов.

Lucene.Net – это перенесенный с платформы Java поисковый движок Lucene. Он поддерживает тот же API и те же классы, что и оригинальная версия. Это накладывает определенный отпечаток, а также делает индекс обратно совместимым для обеих платформ.

Рассмотрим модели поиска, используемые в Lucene.Net. [8–10]

В настоящее время различают **три общие модели поиска**:

1) Булева модель, когда документы при поиске делятся на две группы – либо соответствующие, либо несоответствующие запросу, при этом никакие их оценки не вычисляются. Так как в этой модели нет оценок релевантности документа запросу, то выдается все множество документов, соответствующих запросу, без какого-либо ранжирования.

2) Векторная модель, когда и запросы, и документы моделируются векторами весов n-мерного пространства

$$v(d) = (w_i, \ldots, w_n),\ V(q) = (v_i, \ldots, v_n), \quad (11)$$

где *n* – общее число различных термов (слов) во всех документах коллекции, каждый уникальный терм – измерение;

$w_i$ и $v_i$ – соответственно веса i-го терма в документе *d* и запросе *q*, веса могут вычисляться как *tf-idf* (term frequency – inverse document frequency, частота терма – обратная частота документа).

Релевантность или подобие между запросом и документом вычисляются расстоянием между этими векторами: чем ближе они расположены, тем больше документ *d* соответствует запросу *q*. В векторной модели

часто используют косинусную оценку релевантности q и d

$$\text{cosineSim}(q, d) = V(q)\cdot V(d)/\ |V(q)||V(d)|, \quad (12)$$

где $V(q)\cdot V(d)$ – скалярное произведение двух векторов, а $|V(q)||V(d)|$ – произведение их длин.

Векторная модель специально не требует, чтобы веса были обязательно *tf-idf*, но использование таких весов дает высокоточный поиск. Lucene использует *tf-idf* – функцию, прямо пропорциональную числу документов коллекции, содержащих этот терм.

3) Вероятностная модель, где вычисляется вероятность того, что документ соответствует запросу с использованием полного вероятностного подхода.

Lucene.Net при реализации функции поиска комбинирует векторную и булевую модели. Подход заключается в том, что отбор документов осуществляется в соответствии с булевой моделью, а их ранжирование – в соответствии с векторной моделью.

Уравнение для *cosineSim*(*q*,*d*) можно рассматривать как скалярное произведение нормализованных векторов весов, в том смысле, что деление вектора $V(q)$ на его длину есть его нормализация к единичному вектору.

Lucene уточняет оценку векторной модели *cosineSim*(*q*,*d*) как с точки зрения качества поиска, так и удобства ее вычисления.

1) Нормализация $V(d)$ к единичному вектору может быть проблематичной в том смысле, что таким образом удаляется информация о длине документа. Чтобы избежать этой проблемы, используется множитель, учитывающий его длину, который приводит *V(d)* к вектору длиной, равной или большей единицы: *docLenNorm*(*d*).

2) При индексации документа пользователи могут указать, что одни документы важнее, чем другие, путем присвоения документу показателя важности (т. е. одни документы имеют предпочтение перед другими при прочих равных условиях). Значит, что и оценка каждого документа получит дополнительный множитель важности документа *dBoost*(*d*).

3) Особенностью модели документа Lucene является то, что документ рассматривается как совокупность полей (полей метаданных). В связи с этим каждый терм относится к какому-то конкретному полю. Нормализация длины документа представляет собой нормализацию длин полей документа. Помимо того, что есть множитель важности документа, существуют также множители важности отдельных его полей (например, 0,5 для поля autor, 0,3 – для title и 0,2 – для body).

4) Одно и то же поле может присутствовать в документе многократно (например,

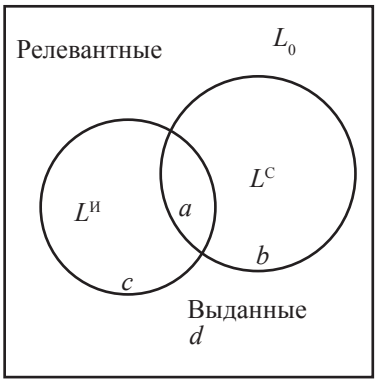


Рис. 1. Диаграмма Эйлера-Венна
Fig. 1. Diagram of the Euler-Venn

| | Релевантные | Нерелевантные |
|---|---|---|
| Выданные | *a* | *b* |
| Невыданные | *c* | *d* |

Рис. 2. Таблица сопряженности выдачи и релевантности
Fig. 2. Crosstabs of issue and relevance

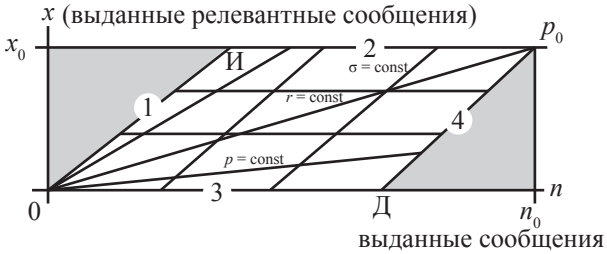


Рис. 3. Диаграмма <*n*,*x*>
Fig. 3. Diagram <*n*, *x*>

в том случае, когда документ имеет несколько авторов), поэтому важность этого поля равна произведению множителей важности отдельных его экземпляров в документе.

5) Во время поиска пользователи могут задать важность для каждого запроса, подзапроса и каждого терма запроса, поэтому вклад каждого терма запроса оценки документа умножается на важность этого терма запроса $qBoost(q)$.

6) Документ может соответствовать некоторым термам запроса и не содержать при этом все его термы (справедливо для некоторых запросов), поэтому имеет смысл повышать оценку релевантности тех документов, которые содержат больше поисковых термов. Для этого в оценку вводят множитель $coord(q,d)$.

На основании вышеизложенного и предполагая для упрощения, что индекс создается для одного поля, получим концептуальную формулу для оценки релевантности поиска Lucene.Net

$$s(q,d)=coord(q,d)qBoost(q)\frac{V(q)V(d)}{|V(q)|}\times \\ \times docLenNorm(d)dBoost(d). \quad (13)$$

Данная концептуальная формула упрощена в том смысле, что принимает во внимание документ, а не его поля, и важность определяется не для запроса, а для термов запроса.

Так как отбор документов осуществляется в соответствии с булевой моделью, использование модифицированного булевого поиска (метод взвешенного зонного ранжирования) позволит улучшить алгоритм поиска Lucene.Net. В методе взвешенного зонного ранжирования происходит деление документа на зоны, например: заголовок, содержание, дата публикации и автор. Метод взвешенного зонного ранжирования присваивает паре ($d$, $q$) значение релевантности на отрезке, вычисляя линейную комбинацию зонных показателей, в которую каждая зона документа вносит булево значение. Пусть существует множество документов, каждый из которых имеет $l$ зон. Тогда $g_1, g_2, ..., g_i \in [0, 1]$, так что

$$\sum_{i=1}^{l} g_i = 1, \quad (14)$$

$S_i$ – булева величина, означающая соответствие (или его отсутствие) между запросом $q$ и $i$-й зоной, где $1 \le i \le l$. Например, если все термины запроса принадлежат конкретной зоне, то ее булево значение должно быть равным единице, а если нет – нулю. Это отображение может осуществлять любая булева функция, показывающая наличие терминов запроса в зоне множества {0,1}. Таким образом, взвешенную зонную релевантность можно определить как

$$Score(d,g) = \sum_{i=1}^{l} g_i s_i. \quad (15)$$

## Заключение

Разработка более совершенных методов алгоритмов поиска Lucene.Net (а именно, использование модифицированного булевого поиска для отбора документов), берущих за основу приведенные модели и обеспечивающих высокую релевантность найденных документов поисковому запросу пользователя за возможно более короткие сроки, а также обладающие точно вычисляемым сроком выдачи результата хотя бы для определенных предметных областей является чрезвычайно актуальной задачей в связи с тем, что базы знаний как частных, так и государственных организаций в настоящее время содержат очень большое количество документов и продолжают пополняться новыми документами.

Самостоятельных поисковых решений, с открытым кодом, которые полностью реализуют индексацию и поиск по собственных алгоритмах на рынке немного, и какое решение выбрать – зависит от особенностей проекта, т. к. для каких-то проектов подойдет простое решение. Следует учитывать, чем сложнее приложение и структура контента или если требуются особые виды поиска и обработки результата, или особое количество или формат данных в проекте, то потребуется собственная поисковая система.

## Библиографический спикок

1. Артюшенко, В.М. Современные исследования в области теоретических основ информатики, системного анализа, управления и обработки информации: монография / В.М. Артюшенко, Т.С. Аббасова, И.М. Белюченко и др. – Королев: ГБОУ ВПО ФТА, 2014. – 318 с.

2. Интеллектуальные технологии и системы // Сб. уч.-метод. работ и статей аспирантов и студентов. Вып. 8. – М.: НОК «CLAIM», 2006. – 326 с.
3. Нистратова, М.В. Возможности использования систем полнотекстового поиска в дистанционном обучении / М.В. Нистратова // Междунар. науч.-практ. конф. «Перспективы, организационные формы и эффективность развития сотрудничества российских и зарубежных ВУЗов» (24–25 апреля 2014 г.).
4. Нистратова, М.В. Возможности использования систем полнотекстового поиска в системе электронного документооборота / М.В. Нистратова // IX Междунар. заоч. науч.-практ. конф. «Тенденции и перспективы развития современного научного знания».
5. Обзор решений для полнотекстового поиска в веб-проектах: Sphinx, Apache Lucene, Xapian http://alexandr.logdown.com/posts/22560.
6. Полнотекстовый поиск в веб-проектах: Sphinx, Apache Lucene, Xapian. http://habrahabr.ru/post/30594/
7. Полнотекстовый поиск и его возможности. http://habrahabr.ru/post/40218/
8. Полнотекстовый поиск с использованием Apache Lucene. http://www.waveaccess.ru/
9. Полнотекстовый поиск в веб-приложениях. http://rsdn.ru/article/inet/full-text_search_for_web-apps.xml
10. Сравнение движков полнотекстового поиска. http://lib.custis.ru.

## SEARCH ALGORITHMS USED IN LUCENE.NET

**Zherdeva M.V.,** pg. University of Technology[(1)]

masha8908@rambler.ru,
[(1)]University of Technology, st. Gagarina, 42, Korolev, Moscow region, Russia

The search models which have been the basis for work of Lucene.Net are considered and some specific techniques of ranging documents are described. The problem of the search of certain contents in a large number of documents during a limited period of time is becoming one of the major challenges. As a rule, the traditional systems of search are oriented at work with the structured data text and are poorly adapted for processing the multimedia and unstructured information. Then there is a problem of the search and that of selecting the relevant information from the big unstructured massif. One of the factors stimulating the development of search technologies is that a huge number of the electronic libraries containing considerable amount of actual knowledge has become available at present. As the choice of search algorithm depends on the features of a project, the development of more perfect methods, which use the given models as a basis and provide high relevance of the found documents to a search query of a user as soon as possible and also possess a precisely calculated term of the result availability, is necessary. Special types of search and processing the result, and special quantity or a format of data in the project are also required. In this article the parameters which should be taken into account while choosing the search mechanism are shown. The existing approaches to the solution of the search problem are analyzed and their improvement is based on the use of Boolean search modification (a method of the balanced zonal ranging) suggested. The criteria of an assessment of information search are given. The conceptual formula for an assessment of Lucene.Net search relevance is shown.

Keywords: information search, document, criteria, relevance, search engines

### References

1. Artyushenko V.M., Abbasova T.S., Belyuchenko I.M., Vasil'yev N.A., Zinov'yev V.N., Strenalyuk YU.V., Vokin G.G., Samarov K.L., Stavrovskiy M.Ye., Poserenin S.P., Razumovskiy I.M., Fominskiy V.YU. *Sovremennyye issledovaniya v oblasti teoreticheskikh osnov informatiki, sistemnogo analiza, upravleniya i obrabotki informatsii* [Modern researches in the field of theoretical fundamentals of informatics, the system analysis, management and information processing]. Korolev, FTA, 2014. 318 p.
2. *Intellektual'nyye tekhnologii i sistemy* [Intellectual technologies and systems] Collection of educational methodical works and articles of graduate students and students. Release 8. Moscow: NOC of «CLAIM», 2006. 326 p.
3. Nistratova M.V. *Vozmozhnosti ispol'zovaniya sistem polnotekstovogo poiska v distantsionnom obuchenii* [Possibilities of use of systems of full text search in distance learning] International scientific and practical conference «Prospects, Organizational Forms and Efficiency of Development of Cooperation of the Russian and Foreign higher education institutions» (April 24-25, 2014).
4. Nistratova M.V. *Vozmozhnosti ispol'zovaniya sistem polnotekstovogo poiska v sisteme elektronnogo dokumentooborota* [Possibilities of using full-text search systems to electronic document management system]. IX International extramural scientific-practical conference «Trends and prospects of development of modern scientific knowledge».
5. *Obzor resheniy dlya polnotekstovogo poiska v veb-proektah: Sphinx, Apache Lucene, Xapian* [The review of decisions for full text search in web projects: Sphinx, Apache Lucene, Xapian.]. http://alexandr.logdown.com/posts/22560
6. Full-text search in web projects: Sphinx, Apache Lucene, Xapian. http://habrahabr.ru/post/30594/
7. Full-text search and features. http://habrahabr.ru/post/40218/
8. Full-text search with Apache Lucene. http://www.waveaccess.ru/
9. Full-text search in web-applications. http://rsdn.ru/article/inet/full-text_search_for_web-apps.xml
10. A comparison of full-text search engines. http://lib.custis.ru.