DOI: 10.58168/MIST2025_291-296
УДК 004.9

АНАЛИЗ СУЩЕСТВУЮЩИХ АЛГОРИТМОВ ПОИСКА И РАНЖИРОВАНИЯ

Е.М. Кузнецова[1], Н.Ю. Юдина[1]

[1]ФГБОУ ВО «Воронежский государственный лесотехнический университет имени Г.Ф. Морозова»

Аннотация. В статье рассматривается работа поисковых систем, которая удовлетворяет запросы пользователей на получение информации. Качество результатов зависит от используемых алгоритмов поиска и алгоритмов ранжирования информации. При выполнении данных алгоритмов поисковая система сканирует и web-страницы и выполняет их индексацию. Методы ранжирования обеспечивают сортировку просмотренных web-страниц по релевантности.

Ключевые слова: web-страница, поисковая система, ранжирование, индексация, сортировка, релевантность.

ANALYSIS OF EXISTING SEARCH AND RANKING ALGORITHMS

E.M. Kuznetsova[1], N.Yu. Yudina[1]

[1]Voronezh State University of Forestry and Technologies named after G.F. Morozov

Abstract. The article examines the work of search engines that satisfy user requests for information. The quality of the results depends on the search algorithms and information ranking algorithms used. When executing these algorithms, the search engine scans web pages and indexes them. Ranking methods ensure that the viewed web pages are sorted by relevance.

Key words: web page, search engine, ranking, indexing, sorting, relevance.

Доступность информации в современном мире значительно выросла, вырос и ее объем. А это привело к тому, что задача поиска информации является весьма актуальной. Схема поисковой системы представлена на рисунке 1. Результатом работы любой поисковой системы является удовлетворение запросов пользователей на получение информации. Качество результатов зависит от используемых алгоритмов поиска и алгоритмов ранжирования информации. При

выполнении данных алгоритмов поисковая система сканирует и web-страницы и выполняет их индексацию.

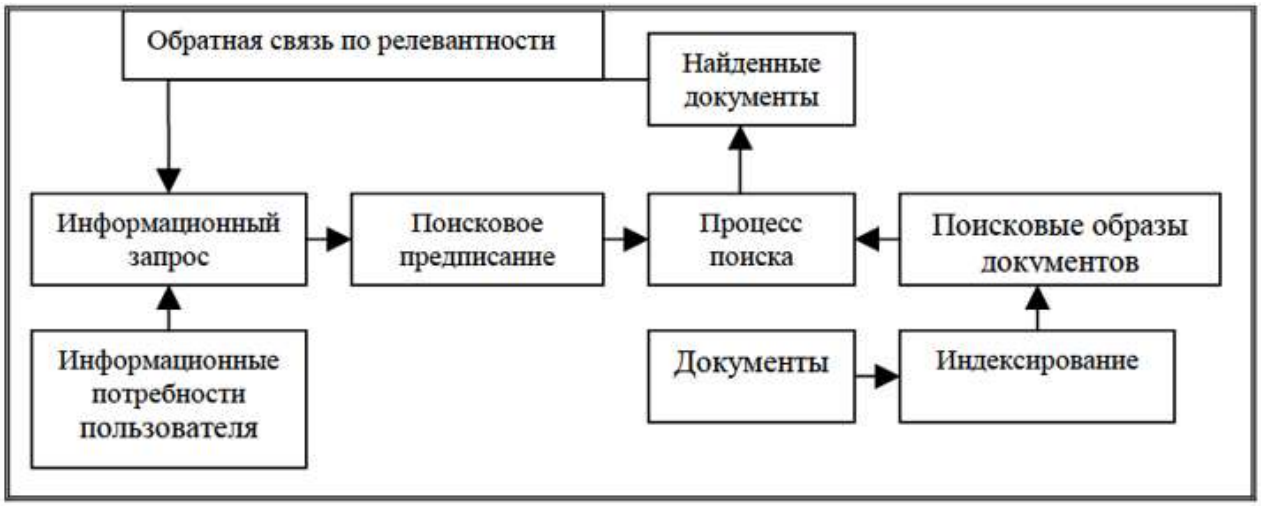

Рисунок 1 – Схема работы поисковой системы

Методы ранжирования обеспечивают сортировку просмотренных web-страниц по релевантности. Для поиска чаще всего используется алгоритм Inverted Index, т.е. алгоритм обратной индексации.

Задачу поиска можно записать следующим образом. Есть коллекция документов

$D = \{d_1, d_2, \dots, d_т\} D\ = \backslash\{d\_1, d\_2, \backslash dots, d\_N\backslash\}$ (1)

где n – количество документов в коллекции.

В каждом документе есть набор слов

$d_i = \{w_{i1}, w_{i2}, \dots, w_{im}\} d\_i\ = \backslash\{w\_\{i1\}, w\_\{i2\}, \backslash dots, w\_\{im\}\backslash\}$ (2)

где $w\_\{ij\}$ – j-ое слово в документе.

Набор слов в документе можно рассматривать как словарь $VS = \{v_1, v_2, \dots, v_m\}$, m – количество уникальных слов. Обозначим множество индексов документов как $P(D)$, тогда обратный индекс можно записать как $I: V \rightarrow P(D)\ I: V\ \backslash to\ P(D)$. А с учетом, что v_i есть частота появления слова в документе, то обратный индекс можно представить формулой (3).

$I(v_i)\ = (d_j, f_{ij})\ I(v\_i)\ = \backslash left(\ d\_j, f\_\{ij\}\ \backslash right)$ (3)

где d_j – просматриваемый j-ый документ, а f_{ij} – частота появления слова.

Алгоритм Inverted Index обеспечивает поиск всех документов, содержащих данный запрос. Запрос может состоять из нескольких слов, а также может быть фразой. Время поиска прямо пропорционально числу просматриваемых документов. Алгоритм Inverted Index легко масштабируется.

Для более точного поиска информации используется алгоритм Keyword Search.

Существует множество ключевых слов, используемых для поиска информации k_j, которые будут использованы для поиска в документах d_i. В данном алгоритме вводится понятие функции принадлежности слова к просматриваемой странице $\phi(d_i, k_j)$. Данная функция равна 1, если слово содержится в документе, в противном случае $\phi(d_i, k_j) = 0$.

В результате поиска будет получено подмножество документов, для которых будет выполняться условие $\exists k_j \in K: \phi(d_i, k_j) = 1$. Тогда модель задачи поиска можно записать в виде $R = \{d_i \in D | \exists k_j \in K: \phi(d_i, k_j\} = 1$

Реализация алгоритма поиска по ключевому представлена на рисунке 2.

```
def keyword_search(documents, keywords):
  result = []
    for document in documents:
    for word in keywords:
      if word in document:
        result. append(document)
        break #
//Если нашли хотя бы одно совпадение, переходим к следующему
документу
  return result
```

Рисунок 2 – Алгоритм поиска Keyword Search.

Выполняя поиск алгоритм Keyword Search ищет совпадения в базе данных, не анализируя при этом контекст. В связи с этим требуется хранить все документы и ключевые слова, что усложняет его работу. Время поиска зависит от объема просматриваемого документа.

Для повышения производительности данного алгоритма используют инвертированные индексы и хеш-таблицы. Эти подходы позволяют значительно сократить время поиска до логарифмической или даже константной зависимости от числа документов и ключевых слов.

Порядок отображения веб-страниц определяется алгоритмами ранжирования, которые определяют набор правил и математических моделей, используемых для сортировки документов по релевантности. При выполнении алгоритма ранжирования оценивается текст, содержащийся на странице, степень соответствия запросу, количество и качество ссылок на просматриваемую страницу, скорость загрузки сайта. его адаптивность под различные устройства.

Ярким представителем алгоритмов ранжирования является алгоритм PageRank, который анализирует количество и качество внешних ссылок на страницу. Авторитетной считается та, которая имеет большее количество качественных ссылок. Таким образом, задачей данного алгоритма является присвоение просматриваемой странице рейтинг, отображающий ее относительную важность.

Алгоритм PageRank решается с помощью итераций, т.е. сначала устанавливается начальное значение $PR(v_i) = \frac{1}{n}$ для всех i, где v_i – количество страниц. Значения рейтинга вычисляются по формуле (4)

$$PR^{t+1} = \frac{1-d}{n}1 + d \cdot M \cdot PR^t \quad (4)$$

Процесс продолжается пока значения вектора PR не стабилизируется, т.е. разница между последующим и предыдущим значениями станет достаточно малой $\|PR^{t+1} - PR^t\| < \varepsilon$, где ε – бесконечно малое положительное число.

Алгоритм использует демпфирующий коэффициент для учёта вероятности случайного перехода и сводит задачу нахождения важности страниц к задаче нахождения фиксированной точки рекуррентной формулы, что делает его эффективным инструментом для ранжирования веб-страниц.

К недостаткам алгоритма PageRank относится его зависимость от количества входящих ссылок на страницу, небольшое внимание на качество ссылок. Также алгоритм медленно адаптируется к изменениям, не всегда учитывается поведение пользователя на странице. PageRank имеет ограниченную эффективность при работе с мультимедийными ресурсами. Современные поисковые системы используют множество методов машинного обучения для обеспечения более точного ранжирования.

Одним из популярных алгоритмов поиска информации является алгоритм Term Frequency-Inverse Document Frequency (TF-IDF). При работе с документом он учитывает два аспекта – это появление слова в документе обратную частоту его появления в других документах. Алгоритм имеет два компонента: Term Frequency и Inverse Document Frequency.

Первый компонент TF определяет частоту появления искомого в документе:

$$\mathrm{TF}(t, d) = \frac{f(t,d)}{\sum_{\omega \in d} f(\omega,d)} \quad (5)$$

где $f(t, d)$ – число вхождений слова f в документ d, $\sum_{\omega \in d} f(\omega, d)$ – общее количество слов в документе.

Вторая составляющая данного алгоритма – Inverse Document Frequency. Она определяет важность слова для всей коллекции документов, т.е. если слово встречается во всех документах, то оно менее информативно. Если слово встречается редко, то оно представляет более высокую ценность для поиска. Сказанное можно описать формулой (6) :

$$IDF(t) = log\left(\frac{n}{DF(t)}\right) \quad (6)$$

где n общее число документов; $DF(t)$ – число документов, которые содержат слово t.

Для избежания деления на ноль, когда искомого слова нет используют формулу (7) :

$$IDF(t) = log\left(\frac{n+1}{DF(t)+1}\right) \quad (7)$$

После определения составляющих их можно объединить (8) :

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \quad (8)$$

К преимуществам $TF - IDF$ можно отнести следующее: интуитивно понятен; достаточно просто можно вычислить частоту вхождения слова в документ; скорость работы с большими документами.

В качестве недостатков можно отметить то, что он не принимает в расчет порядок слов, зависит от коллекции документов; не работает с синонимами.

Улучшением алгоритма $TF - IDF$ является алгоритм Best Matching 25. Этот алгоритм учитывает частоту слов и параметр. Который регулирует длину документа.

Для улучшения результатов ранжирования современные поисковые системы активно применяют машинное обучение. Модели, основанные на искусственном интеллекте, могут анализировать огромные объемы данных и выявлять скрытые паттерны, которые трудно обнаружить с помощью традиционных методов. Модели, такие как RankNet, LambdaMART и XGBoost, используются для обучения на исторических данных и повышения качества ранжирования.

Список литературы

1. Зибарева, И. В. Информационно-поисковая система SciFinder : учебно-методическое пособие / И. В. Зибарева. – Москва : Директ-Медиа, 2020. – 161 с.

2. Справочно-правовые информационно-поисковые системы : учебно-методическое пособие / составитель М. В. Матвеева. – Воронеж : ВГУ, 2020. — 27 с.

3. Горбунов, В. Г. Сравнительный анализ методов «Прометей» и нечетких отношений в условиях принятия решений / В. Г. Горбунов, О. Л. Бордюжа, А. А. Пак // Моделирование систем и процессов. – 2024. – Т. 17, № 1. – С. 42-56. – DOI 10.12737/2219-0767-2024-17-1-42-56. – EDN LSMBMD.

4. Зольников В.К., Гамзатов Н.Г., Анциферова В.И., Полуэктов А.В., Фиронов В.А. Экспериментальные исследования радиационного воздействия на микросхемы FRAM // Моделирование систем и процессов. – 2022. – Т. 15, № 3. – С. 16-24.

5. Полуэктов, А. В. Моделирование колебательных процессов в пакете MVSTUDIUM / А. В. Полуэктов, К. В. Зольников, В. И. Анциферова // Моделирование систем и процессов. – 2021. – Т. 14, № 4. – С. 139-148. – DOI: 10.12737/2219-0767-2021-14-4-139-148.

References

1. Zibareva, I. V. Information search engine SciFinder : an educational and methodological guide / I. V. Zibareva. Moscow : Direct-Media, 2020, 161 p.

2. Legal reference information and search engines : an educational and methodical manual / compiled by M. V. Matveeva. Voronezh : VSU, 2020. 27 p.

3. Gorbunov V. G., Bordyuzha O. L., Pak A. A. Comparative analysis of Prometheus methods and unclear relationships in decision-making // Modeling of systems and processes. – 2024. – Vol. 17, No. 1. – pp. 42-56. – DOI 10.12737/2219-0767-2024-17-1-42-56. – EDN LSMBMD.

4. Zolnikov V.K., Gamzatov N.G., Antsiferova V.I., Poluektov A.V., Fironov V.A. Experimental studies of radiation effects on FRAM chips // Modeling of systems and processes. – 2022. – Vol. 15, No. 3. – pp. 16-24.

5. Poluektov, A. V. Modeling of oscillatory processes in the MVSTUDIUM package / A. V. Poluektov, K. V. Zolnikov, V. I. Antsiferova // Modeling of systems and processes. – 2021. – Vol. 14, No. 4. – pp. 139-148. – DOI: 10.12737/2219-0767-2021-14-4-139-148.