УДК 517.97 : 532.526
DOI: 10.25559/SITITO.019.202301.180-188

Оригинальная статья

Интеллектуально-поисковая система для работы с большими данными

И. Ф. Астахова[1*], К. А. Маковий[2], Л. С. Никитин[1], Ю. В. Хицкова[1]
[1] ФГБОУ ВО «Воронежский государственный университет», г. Воронеж, Российская Федерация
Адрес: 394018, Российская Федерация, г. Воронеж, Университетская площадь, д. 1
[2] ФГБОУ ВО «Воронежский государственный технический университет», г. Воронеж, Российская Федерация
Адрес: 394006, Российская Федерация, г. Воронеж, ул. 20-летия Октября, д. 84
[*] astachova@list.ru

Аннотация

В статье представлена система моделирования информационно-поисковой системы в сети Интернет. Предлагается разработанное приложение, которое обеспечивает работу информационно-поисковой системы по следующим направлениям: по модели сбора данных, по решению проблемы индексирования, по модели ранжирования, по решению проблемы хранения. Решения в этой области развиваются наиболее активно, благодаря прогрессу в сфере искусственного интеллекта, облачных технологий и обработки естественных языков. Эти факторы сделали исследования в области разработки интеллектуальных информационно-поисковых систем (ИПС), осуществляющих сбор информации в сети Интернет и реализующих поиск по найденным данным, доступными при отсутствии внушительных материальных ресурсов. Основные проблемы, которые необходимо решить при разработке ИПС: проблема сбора данных; проблема индексирования; модель индекса, ее выбор и развитие; проблема ранжирования; проблема хранения; проблема оценки качества. Интеллектуальность поиска обеспечена за счёт использования ранжирования с помощью методов tf-idf, векторной модели и ссылочного анализа, позволяющих находить релевантные документы, не содержащие прямого вхождения слов из запросов и сортировать их по степени соответствия запросу. Разработанное приложение основано на языке Python, проведены тестовые запуски системы, показавшие ее работоспособность, объясняется организация интеллектуальной составляющей.

Ключевые слова: информационно-поисковая система, модель ранжирования, база данных, тестирование системы

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Интеллектуально-поисковая система для работы с большими данными / И. Ф. Астахова [и др.] // Современные информационные технологии и ИТ-образование. 2023. Т. 19, № 1. С. 180-188. doi: https://doi.org/10.25559/SITITO.019.202301.180-188

Original article

Intelligent Search System for Working with Big Data

I. F. Astakhova[a*], K. A. Makovy[b], L. S. Nikitin[a], Yu. V. Khitskova[a]
[a] Voronezh State University, Voronezh, Russian Federation
Address: 1 Universitetskaya pl., Voronezh 394018, Russian Federation
[b] Voronezh State Technical University, Voronezh, Russian Federation
Address: 84, 20 letiya Oktyabrya St., Voronezh 394006, Russian Federation
[*] astachova@list.ru

Abstract
The article describes a system for modeling an information retrieval system on the Internet. The developed application is described, which allows the operation of the information retrieval system according to the following parameters: according to the data collection model, to the solution of the indexing problem, according to the ranking model, to the solution of the storage problem. Solutions in this area are developing most actively, thanks to progress in the field of artificial intelligence, cloud technologies and natural language processing. These factors have made re-search, the development of intelligent information retrieval systems (IRS), which collect information on the Internet and implement a search based on the data found. This search is available in the absence of impressive material resources. The main problems to be solved in the development of IRS: the problem of data collection; indexing problem; index model, its choice and development; ranking problem; storage problem; quality assessment problem. Search intelligence is provided through the use of ranking using the tf-idf methods, vector model and link analysis, which allow you to find relevant documents that do not contain direct occurrences of words from queries and sort them according to the degree of matching the query. The developed application in the Python language is described, test runs of the system were carried out, which showed its performance, and the organization of the intellectual component is explained.

Keywords: information retrieval system, ranking model, database, system testing

Conflict of interests: The authors declare no conflict of interest.

For citation: Astakhova I.F., Makovy K.A., Nikitin L.S., Khitskova Yu.V. Intelligent Search System for Working with Big Data. *Modern Information Technologies and IT-Education.* 2023;19(1):180-188. doi: https://doi.org/10.25559/SITITO.019.202301.180-188

Введение

В настоящее время большая часть информации, накопленной человечеством, хранится на электронных носителях: веб-серверах, электронных таблицах, файлах приложений, как в структурированном, так и неструктурированном виде, а объёмы и темпы накопления данных возрастают экспоненциально. В сложившихся условиях главным способом осуществления потребности в получении новых сведений на заданные темы и разрешении возникающих вопросов стал глобальный поиск в сети Интернет. Основными поставщиками продуктов, предоставляющих доступ к таким услугам, являются компании гиганты сферы информационных технологий, такие как Google, Microsoft, Yandex др. Все они обладают внушительными средствами, необходимыми для построения систем, решающих нетривиальную задачу поиска нужных документов среди всего набора Интернет-ресурсов. Вдобавок, каждый сервис, приложение или web-сайт, содержащий обширный набор данных реализует функционал выборки по базе внутренних ресурсов. Поисковые системы прошли долгий путь развития, расширяясь одновременно с сетью Интернет, используемые алгоритмы и масштабы кардинально менялись с течением времени для удовлетворения пользовательских запросов при экспоненциально увеличивающемся объёме цифровой информации в мире. Представить использование интернет-среды без поисковых систем в настоящее время довольно трудно. Приходилось бы иметь закладки на все случаи жизни или знать доменные имена необходимых веб-сайтов, а новые ресурсы не имели бы возможности попасть в поле зрения интернет-пользователей без оплаты реклама-мы на авторитетном источнике. Помимо определения прямого соответствия содержания документа запросу, в задачу поиска также входит расширение запроса (пользовательский ввод, зачастую, недостаточно чёткий для прямого поиска), ранжирование 4 результатов по множеству весовых факторов (авторитетность ресурса, дата публикации, популярные запросы на текущее время, предпочтения конкретного пользователя и др.). Перечисленное выше можно обобщить понятием «интеллектуальный поиск». Данная работа посвящена именно этой области проблемы, так как решения в ней развиваются наиболее активно, благодаря прогрессу в сфере искусственного интеллекта, облачных технологий и обработки естественных языков. Эти факторы сделали исследования, разработку интеллектуальных информационно-поисковых систем (ИПС), осуществляющих сбор информации в сети Интернет и реализующих поиск по найденным данным, доступными при отсутствии внушительных материальных ресурсов.

Основные проблемы, которые необходимо решить при разработке ИПС:

1. Проблема сбора данных.
2. Проблема индексирования.
3. Модель индекса, ее выбор и развитие.
4. Проблема ранжирования.
5. Проблема хранения.
6. Проблема оценки качества.

Анализ существующих решений позволил выделить следующие разработки:

1. Модель информационно-поисковой системы Яндекс.
2. Модель информационно-поисковой системы ElasticSearch.

ИПС Яндекс удовлетворяет требованиям высокопроизводительной поисковой системы в сети Интернет, реализуя распределённый сбор, хранение, анализ медиа данных разных форматов и выполняя пользовательские запросы за приемлемое время, обеспечивая приемлемую точность и полноту поиска. Недостатком системы является её закрытость, в силу коммерческой направленности, и невозможность, вследствие, 40 использовать систему в своих целях, определять параметры ранжирования, собирать всю статистику для исследовательских целей. ИПС ElasticSearch, в свою очередь, также удовлетворяет соответствующим требованиям ИПС для сети Интернет, является открытой и предоставляет возможность самостоятельной выборке данных, использования в личных целях, сбора любой статистики и редактирования алгоритмов ранжирования и работы системы в целом. Недостатком является отсутствие встроенного модуля сбора и хранения необработанных (включающих HTML код) данных из сети Интернет, встроенная система хранения предназначена лишь для частично обработанных данных. В результате, для использования системы с целью поиска, исследования данных, возможности восстановления структур из сырых данных, необходимо реализовывать собственные модули сбора и хранения, а также взаимодействие этих модулей с системой ElasticSearch. Разрабатываемая в рамках данной работы поисковая система призвана покрыть недостатки вышеперечисленных решений, являясь открытым продуктом, реализующим полную цепочку работы с информацией, от сбора, до результатов поискового запроса, возможностью модификации и отслеживания на любом из шагов цепочки.

1. Этапы исследования

С целью осуществления сбора открытых данных из сети Интернет, их хранения, анализа, предоставления возможности поиска по данным и сбора статистической информации, необходимо разработать систему полного цикла обращения и обработки данных, содержащую следующие модули:

- Модуль сбора данных из сети Интернет.
- Модуль хранения собранных данных.
- Модуль индексации и хранения индекса.
- Модуль ранжирования.
- Модуль обработки пользовательских запросов на русском языке.

Основные критерии производительности системы:

- Осуществлять сбор, обработку информации в режиме многопоточности.
- Осуществлять сбор, обработку текстов, запросов на русском языке.
- Иметь возможность хранить от 50 тыс. документов на дисковом пространстве 50 Гб.
- Производить обработку пользовательских запросов в пределах 5 сек.
- Оценивать релевантность документов методом анализа ссылок и соответствия запросу.

Представленные требования помогут создать достаточно производительную полноценную поисковую систему, способную самостоятельно формировать набор данных для исследова-

ния методов анализа естественного языка, алгоритмов поиска и ранжирования [1-3].

Сбор информации решено осуществлять с портала wikipedia.org. Русскоязычный раздел сайта содержит порядка 1,800,000 статей широкого спектра тематики. Отличительной особенностью портала является большое количество ссылок между статьями внутри их текста, это позволит эффективно реализовать метод ссылочного анализа документов. Предположительно, выбор подобного источника с широкой тематикой откроет возможность предоставлять пользователю осмысленный ответ на запросы различных формулировок. Так как возможности сбора ограничены размером дискового пространства и временем беспрерывной работы вычислительной машины, в рамках данной работы другие источники использоваться не будут. В качестве основы для робота сбора используется библиотека Scrapy, основания приведены в первой главе.

Хранение информации реализовано с помощью СУБД PostgreSQL, так как она предоставляет инструменты реляционной базы данных в комбинации с инструментами нереляционной, удовлетворяя потребности в хранении на раннем этапе развития системы и имеет потенциал расширения до распределённой системы хранения при увеличении количество собираемой информации. К тому же, большая часть таблиц схемы, представленной далее, не нуждаются в нереляционных инструментах и могут поддерживаться SQL системой. По сравнению с MongoDB, выбранное средство обеспечит быстрый старт, и работу без дополнительных настроек. В дополнение, встроенный тип данных TOAST поддерживает запись длинных полей, таких как текст документа, и может быть сжат встроенными средствами, для достижения необходимой эффективности хранения.

Индексацию решено проводить комбинацией методов инвертированного индекса, в сочетании с оценкой BM25 для терминов в 43 документах, учётом позиции термина методом, приведённым в разделе 1.2.3, и векторной модели, разработанной по словарю. Инвертированный индекс с дополнительными факторами облегчит поиск и улучшит ранжирование документов по запросу. А векторная модель позволит производить более широкий анализ документов и находить релевантные запросу даже без прямого вхождения слов из запроса в них. Также векторная модель упрощает индексацию документов, давая возможность добавлять термины в словарь в изначальном виде, без приведения к базовой форме. Расширение пользовательского запроса с помощью поиска ближайших соседей терминов в модели нейтрализует недостаток подхода с отсутствием приведения терминов перед добавлением в словарь.

Ранжирование реализовано на основе оценок, определяемых при индексации (BM25, позиция), коэффициента расширения запроса (документы, содержащие прямое вхождение слов из запроса, имеют приоритет над документами, содержащих похожие слова), а также ссылочного анализа. Ссылочный анализ проводится, аналогично методу PageRank, проходами по графу ссылок между документами, распределением веса между ними с коэффициентом угасания.

Для учёта всех факторов разработана следующая модель ранжирования:

$$score(d,t) = (PR(d) + bm25(d,t) * (1{+}{+} q * pos)) * neigh\,bor(d,t) \quad (1)$$

где $score(d,t)$ – итоговая оценка релевантности документа d расширенному запросу t;

$PR(d)$ – оценка ссылочного анализа документа d;

$bm25(d,t)$ – суммарная оценка релевантности BM25 для терминов расширенного запроса t входящих в документ d;

pos – позиционный фактор, вещественное число [0,1];

q – коэффициент, ограничивающий влияние позиционного фактора, q = 0.15, что означает модификатор позиционного бонуса к ранжированию до 15%;

$neigh\,bor(d,t)$ – коэффициент косинусного расстояния между векторов терминов оригинального запроса и расширенного запроса, входящих в документ, если терминов несколько высчитывается среднее арифметическое расстояний, Приведённая формула позволяет объединить оценку всех факторов, подсчитываемых в системе.

Основной вклад в итоговую оценку вносят результат ссылочного анализа PR() и оценка bm25(), bm25 также имеет позиционный бонус. При нахождении термина в нужной позиции документа, он получает увеличение оценки bm25 на коэффициент от 0 до q процентов. Максимальный размер позиционного бонуса решено оценить в 15%. Полученная сумма оценок дополнительно умножается на коэффициент neighbor(), который поощряет документы, содержащие слова из пользовательского запроса, и штрафует доку-менты со словами из расширенного запроса. Таким образом, документы, подходящие под изначальную формулировку, имеют более высокий приоритет. В итоге выбранные документы сортируются по величине итогового счёта, определяемого на предыдущих шагах, и отображаются пользователю.

Прием пользовательских запросов и вывод результатов будут производиться через консоль. В дальнейшем, имеет смысл выводить результаты на HTML странице с помощью веб-сервера для обеспечения быстрого доступа по ссылкам без необходимости копирования и расширения возможностей отображения результатов.

При разработке программного продукта использовались нижеперечисленные средства. Язык программирования Python 3.8 предоставляет самый практичный другими языками программирования, реализации проектов в области «больших» данных, машинного обучения и нейросетей, одним из которых и является поставленная задача. PyCharm 2021.1 (Community Edition) – одна из наиболее современных и комфортных сред разработки для языка Python и следующие библиотеки:

Библиотека Scrapy – сбор и первичная обработка текстовых данных из документов сети Интернет.

Библиотека Pandas – удобная работа со списками, массивами данных большого объёма.

Библиотека NumPy – оптимизированные операции для работы с векторами.

Библиотека SpaCy – фильтрация токенов (отсев часто используемых и незначимых, например, предлогов и союзов) с использованием модели, подготовленной для обработки русского языка.

Библиотека Gensim – формирование и обучение модели векторов слов с гибкими настройками различных параметров, таких как контекстное окно (сколько токенов до и после текуще-

го будет проанализировано), размер дополнительно исследуемых подстрок токена, величина негативного семплирования, количество потоков, используемых для обучения.

Библиотека rank_bm25 – предоставление функций TF/IDF формулы для расчёта весов векторов слов, относительно документа.

Библиотека NMSlib – формирование поискового индекса на основе имеющихся векторов документов для значительного ускорения операции поиска по запросу.

СУБД PostgreSQL 13 – создание и поддержка базы данных с возможностью расширения до распределённой базы.

2. Структура проекта

2.1. Структура базы данных

Представлено следующее описание сущностей базы данных поисковой системы:

• documents_raw – содержит малообработанный текст страниц, получаемый из модуля сбора данных, а также содержащиеся в документы ссылки. Поле, сохраняющее текст страницы имеет специальный тип TOAST, присущий СУБД PostgreSQL, и позволяющих сохранять данные большой длины;

• documents – основная таблица, содержащая docId, ссылку на документ для быстрого доступа, дата появление/обновления содержания документа в базе, служит основой для расписания работы модуля сбора;

• terms – таблица с лексиконом поисковой системы, содержит termId, и сам термин, вторая основная сущность поисковой системы;

• index – объединяет термины и документы, в которых они содержатся, при сортировке по termId это инвертированный индекс, при сортировке по docId – прямой, termVector содержит в бинарном виде информацию о позиции термина в документе, количестве его вхождений, факторе модификации счёта для термина, также предоставляет данные для модуля сбора данных, также использует тип данных TOAST;

• links_relation – таблица, помогающая оценить рейтинг надёжности сайта с помощью анализа входящих и исходящих ссылок, содержит docId документа источника и docId документа по ссылке (если он уже был проиндексирован);

• pr_score – сохраняет результаты ссылочного анализа: docId, счёт, количество входящих и исходящих ссылок, который учитывается при ранжировании. Также при исполнении программы создаётся и сохраняется индекс векторной модели представления документов, формируемый с помощью библиотеки Pickle. Из-за специфического формата хранения (p) и небольшого размера решено не сохранять его в базе данных, а хранить отдельным файлом и загружать в оперативную память во время исполнения программы.

2.2. Модули проекта

Для каждого модуля, кроме паука, функции вынесены в отдельный файл Python скрипта и хранятся в папке modules. Проект состоит из следующих модулей [4-6]:

Spider parser, модуль, управляющий роботом автоматического сбора данных с сайтов, построенным на основе библиотеки Scrapy. Из-за структуры библиотеки робот управляется консольными командами и конфигурацией из файлов в соответствующих папках. Для управления роботом создан отдельный класс с функциями введения консольных команд (запуск бота с параметрами, остановка бота), изменения конфигурационных файлов робота. Scrapy управляет сетевым поведением бота: подчинение инструкциям; взаимодействие модулей robots.txt, ограничение количество запросов минуту, многопоточное выполнение. В рамках разработки остаётся настройка парсера под конкретные сайты, а также манипуляции его конфигурацией.

Document manager модуль реализует функции работы с таблицами документов documents_raw и documents. Инкрементальная запись, чтение по идентификаторам, выборка содержимого из длинных и двоичных полей для каждой таблиц. Создание базы данных, таблиц, а также запросы к ним реализованы на языке SQL с помощью библиотеки работы с PostgreSQL для Python – psycopg2.

Index manager модуль производит операции базы данных с таблицами index, terms. Так как добавление документа в индекс представляет многошаговую сложную задачу из очищения, разбиении текста на слова, добавления новых терминов в словарь, подсчёт количества терминов в документе, считывание их позиций, добавление ссылок в соответствующую таблицу его функционал выделен в отдельный класс[1] [7-10]. Получает содержание документов от document manager, разбивает на составные части и, работа модуля сбора распределяет их по вспомогательным структурам. Модуль использует большую часть библиотек, перечисленных в средствах реализации, для работы с текстом построения индекса и векторной модели представления документов. Для обеспечения качественной работы векторной модели терминов, необходимо провести обучения на большом наборе текстов [11-14]. В результате, при добавлении основной части документов в индекс, векторная модель не будет доступна, так как она будет находится в процессе обучения [15-25]. После её обучения модель станет доступна для полноценной работы.

Link manager модуль производит операции базы данных с таблицами link_relations и pr_score. Получает данные о документе и его исходящих ссылках от index manager при индексации документов, производит расчёт итоговых весов документов, обходя граф ссылок по модели PageRank, сортирует и записывает данные в таблицу pr_score. Документы, не имеющие исхо-

[1] Beebe N. H. F. A Bibliography of Publications about the Google PageRank Algorithm. Version 1.160. University of Utah, 2023. URL: https://ftp.math.utah.edu/pub/tex/bib/pagerank.pdf (дата обращения: 29.11.2022).

дящих ссылок, при расчётах, удаляются за несколько итераций во избежание чрезмерной аккумуляции в них весов без дальнейшего Схема обработки документов модуле индексации распределения по вершинам. Данные, предоставляемые модулем, играют большую роль при ранжировании.

В Query модуль передаётся текст пользовательских запросов, расширяется и, с помощью агрегированной информации из остальных модулей в нём определяется поисковой ответ. Расширение запроса происходит с помощью векторной модели на словаре. Расширенный запрос передаётся на обработку в модуль индексации, откуда получается список идентификаторов релевантных документов, для этого списка из менеджера ссылок запрашивается ссылочная оценка. С помощью комбинации оценок выводится итоговая оценка документов и пользователю презентуется конечная поисковая выдача.

По результатам подсчёта итоговой оценки по формуле (1), документы сортируются по убыванию и отображаются пользователю в виде списка, содержащего название документа и ссылку. Для наглядности также выводится итоговая оценка документа.

Тестирование приложения

В результате процесса сбора информации за 10 часов было загружено и порядка 52 тыс. страниц с ресурса ru.wikipedia.org. Обработка текста и создание вспомогательных структур заняло ещё около 12 часов. С увеличением количества проанализированных документов обработка каждого нового занимала всё меньше времени. Занимаемое системой дисковое пространство при этом количестве документов не превысило 30 Гб.

Взаимодействие с программой происходит через меню в консоли. Представлено два раздела меню: для администратора и для пользователя. Через меню администратора доступны функции управления модулями. В меню пользователя предоставлен доступ к введению запросов. Представлены сценарии управления модулями индексирования (часть Б) и сбора информации (часть А) посредством меню администратора. В обоих случаях производится запуск работы модулей.

Происходит демонстрация работы программы с введением поискового запросов в роли пользователя. По причине недостаточно большой выборки документов выводятся 5 наиболее релевантных документов, их заголовок, ссылка и итоговый счёт документа для наглядности.

По запросу «философия древней Греции» результирующая выдача получена за удовлетворительный срок меньше 2 секунд. Релевантность выдачи относительно запроса и выборки документов выглядит субъективно приемлемо. Статьи схожей с запросом тематики, но общей направленности, получили более высокую итоговую оценку, что может быть объяснено структурой ссылочных связей источника. На ресурсе wikipedia.org ссылки от узкоспециализированных статей к базовым встречаются чаще, чем от базовых к специализированным, в результате статьи общей направленности получают высокую ссылочную оценку и попадают в верх выдачи

Похожее поведение наблюдается и в запросе, состоящем из одного слова: ссылочная структура сильно влияет на итоговый счёт документов и статьи общего характера оказываются на верху выдачи. По запросу «МАТРИЦА» высшие позиции заняли документы, относящиеся к математическим терминам, вероятно, по причине того, что количество документов из сферы математической науки, попавших в индекс, оказалось больше, чем количество документов из сферы кинематографа. Время выполнения запроса уменьшилось, так как в выборке по запросу из одного слова оказалось меньше документов и количество вычислений сократилось.

При увеличении количества проиндексированных документов будет иметь смысл расширить количество документов выдаче, а также модифицировать формулу итогового ранжирования для того, чтобы узкоспециализированные документы, соответствующие запросу, получали оценку выше, чем документы общей тематики.

Заключение

В результате работы была создана система информационного поиска в сети Интернет, реализующая сбор, хранения, обработку данных и предоставление возможности поиска по ним. Интеллектуальность поиска обеспечена за счёт использования ранжирования с помощью методов tf-idf, векторной модели и ссылочного анализа, позволяющих находить релевантные документы, не содержащие прямого вхождения слов из запросов и сортировать их по степени соответствия запросу. За время работы системы было собрано, сохранено и проанализировано 52 тысячи документов. Итоговый размер исходных и вспомогательных данных не превысил 30 Гб, а среднее время выполнения запросов составило меньше 5 секунд.

Список использованных источников

[1] Кириллов А. В. Поисковые системы: компоненты, логика и методы ранжирования // Бизнес-Информатика. 2009. № 4. С. 51- 59. EDN: KZXGGJ

[2] Галиев Т. Э. Методы ранжирования поисковой информации в корпоративных поисковых системах // Открытое образование. 2012. № 1. С. 46-51. EDN: PLQHGJ

[3] Марина М. С. Поисковая система Яндекс // Вестник магистратуры. 2014. Т. 1, № 4. С. 82-84. EDN: SAVBVD

[4] Трифонов А. А. Алгоритмы построения инвертированного индекса для коллекции текстовых данных // Известия высших учебных заведений. Поволжский регион. Технические науки. 2013. № 3(27). С. 52-61. EDN: SBVDQP

[5] Sankpal L. J., Patil S. H. Rider-Rank Algorithm-Based Feature Extraction for Re-ranking the Webpages in the Search Engine // The Computer Journal. 2020. Vol. 63, issue 10. P. 1479-1489. https://doi.org/10.1093/comjnl/bxaa032

[6] Patel P., Patel K. A Review of PageRank and HITS Algorithms // International Journal of Advance Research in Engineering, Science & Technology. 2015. Vol. 2, issue 1. P. 2394-2444.

[7] Тагаров Б. Ж. Развитие рынка поисковой оптимизации в России // Креативная экономика. 2018. Т. 12, № 9. С. 1373-1384. https://doi.org/10.18334/ce.12.9.39379
[8] Латыпов А. Р. Обзор влияния поведения пользователей в поисковых алгоритмах // Современные материалы, техника и технологии. 2015. № 2(2). С. 92-97. EDN: UNUUBP
[9] Brin S., Page L. The anatomy of a large-scale hypertextual Web search engine // Computer Networks and ISDN Systems. 1998. Vol. 30, issues 1-7. P. 107-117. https://doi.org/10.1016/S0169-7552(98)00110-X
[10] Васяева Н. С., Дегаев М. Н. Формализация модели построения индексов для информационно-поисковых систем // Международный научно-исследовательский журнал. 2022. №. 6-1(120). С. 56-60. https://doi.org/10.23670/IRJ.2022.120.6.007
[11] SetRank: Learning a Permutation-Invariant Ranking Model for Information Retrieval / L. Pang [и др.] // Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20). New York, NY, USA : Association for Computing Machinery, 2020. P. 499-508. https://doi.org/10.1145/3397271.3401104
[12] Жердева М. В., Артюшенко В. М. Стемминг и лемматизация в Lucene.Net // Вестник Московского государственного университета леса – Лесной вестник. 2016. Т. 20, № 3. С. 131-134. EDN: WKNMTN
[13] Thota P., Ramez E. Web Scraping of COVID-19 News Stories to Create Datasets for Sentiment and Emotion Analysis // Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference (PETRA '21). New York, NY, USA : Association for Computing Machinery, 2021. P. 306-314. https://doi.org/10.1145/3453892.3461333
[14] Сорокин В. Е. Хранение и эффективная обработка нечетких данных в СУБД PostgreSQL // Программные продукты и системы. 2017. Т. 30, № 4. С. 609-618. https://doi.org/10.15827/0236-235X.030.4.609-618
[15] Avci C., Tekinerdogan B., Athanasiadis I. N. Software architectures for big data: a systematic literature review // Big Data Analytics. 2020. Vol. 5. Article number: 5. https://doi.org/10.1186/s41044-020-00045-1
[16] Xiaojie X., Yuan F., Jian W. The Basic Principle and Applications of the Search Engine Optimization // Proceedings of the 2012 International Conference of Modern Computer Science and Applications. Advances in Intelligent Systems and Computing ; ed. by Z. Du. Vol. 191. Berlin, Heidelberg : Springer, 2013. P. 63-69. https://doi.org/10.1007/978-3-642-33030-8_11
[17] Big Data architecture for intelligent maintenance: a focus on query processing and machine learning algorithms / C. Lehmann [и др.] // Journal of Big Data. 2020. Vol. 7. Article number: 61. https://doi.org/10.1186/s40537-020-00340-7
[18] Lee D., Camacho D., Jung J.J. Smart Mobility with Big Data: Approaches, Applications, and Challenges // Applied Sciences. 2023. Vol. 13, no. 12. Article number: 7244. https://doi.org/10.3390/app13127244
[19] Sparck Jones K., Walker S., Robertson S. E. A probabilistic model of information retrieval: development and comparative experiments: Part 1 // Information Processing & Management. 2000. Vol. 36, issue 6. P. 779-808. https://doi.org/10.1016/S0306-4573(00)00015-7
[20] Application of Recommender System in Intelligent Community under Big Data Scenario / C. Liu, Z. Chen, D. Cao, M. Shang // Proceedings of the 2nd International Conference on Big Data Technologies (ICBDT'19). New York, NY, USA : Association for Computing Machinery, 2019. P. 92-96. https://doi.org/10.1145/3358528.3359551
[21] Sun Z., Huo Y. A Managerial Framework for Intelligent Big Data Analytics // Proceedings of the 2nd International Conference on Software Engineering and Information Management (ICSIM'19). New York, NY, USA : Association for Computing Machinery, 2019. P. 152-156. https://doi.org/10.1145/3305160.3305211
[22] Serrano W. A Big Data Intelligent Search Assistant Based on the Random Neural Network // Advances in Big Data. INNS 2016. Advances in Intelligent Systems and Computing ; ed. by P. Angelov, Y. Manolopoulos, L. Iliadis, A. Roy, M. Vellasco. Vol. 529. Cham : Springer; 2017. P. 254-261. https://doi.org/10.1007/978-3-319-47898-2_26
[23] Sanderson M. Test Collection Based Evaluation of Information Retrieval Systems // Foundations and Trends in Information Retrieval. 2010. Vol. 4, no. 4. P. 247-375. https://doi.org/10.1561/1500000009
[24] Information retrieval algorithms and neural ranking models to detect previously fact-checked information / N. Chakraborty, V. La Gatta, V. Moscato, G. Sperlì // Neurocomputing. 2023. Vol. 557. Article number: 126680. https://doi.org/10.1016/j.neucom.2023.126680
[25] Sun Z. Intelligent Big Data Analytics: A Managerial Perspective // Managerial Perspectives on Intelligent Big Data Analytics ; ed. by Z. Sun. Hershey, PA : IGI Global, 2019. P. 1-19. https://doi.org/10.4018/978-1-5225-7277-0.ch001

Поступила 29.11.2022; одобрена после рецензирования 27.01.2023; принята к публикации 21.02.2023.

Об авторах:

Астахова Ирина Федоровна, профессор кафедры математического обеспечения ЭВМ факультета прикладной математики, информатики и механики, ФГБОУ ВО «Воронежский государственный университет» (394018, Российская Федерация, г. Воронеж, Университетская площадь, д. 1), доктор технических наук, профессор, **ORCID: https://orcid.org/0000-0002-2627-8508,** astachova@list.ru

Маковий Катерина Александровна, доцент кафедры систем управления и информационных технологий в строительстве факультета информационных технологий и компьютерной безопасности, ФГБОУ ВО «Воронежский государственный технический университет» (394006, Российская Федерация, г. Воронеж, ул. 20-летия Октября, д. 84), кандидат технических наук, **ORCID: https://orcid.org/0000-0003-3921-6047,** makkatya@mail.ru

Никитин Лев Сергеевич, студент кафедры математического обеспечения ЭВМ факультета прикладной математики, информатики и механики, ФГБОУ ВО «Воронежский государственный университет» (394018, Российская Федерация, г. Воронеж, Университетская площадь, д. 1), **ORCID: https://orcid.org/0009-0007-7962-261X**

Хицкова Юлия Владимировна, доцент кафедры региональной экономики и территориального управления экономического факультета, ФГБОУ ВО «Воронежский государственный университет» (394018, Российская Федерация, г. Воронеж, Университетская площадь, д. 1), кандидат экономических наук, **ORCID: https://orcid.org/0000-0001-6322-8633,** prosvetovau@list.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

References

[1] Kirillov A. Search Engines: Components, Logic and Ranking Methods. *Business Informatics*. 2009;(4):51- 59. (In Russ., abstract in Eng.) EDN: KZXGGJ

[2] Galiev T.A. Methods of ranking of searching information in corporate searching systems. *Open Education*. 2012;(1):46-51. (In Russ., abstract in Eng.) EDN: PLQHGJ

[3] Marina M.S. Yandex Search Engine. *Vestnik Magistratury*. 2014;1(4):82-84. (In Russ., abstract in Eng.) EDN: SAVBVD

[4] Trifonov A.A. Algorithms of inverted index construction for text data collection. *University proceedings. Volga region. Technical sciences*. 2013;(3):52-61. (In Russ., abstract in Eng.) EDN: SBVDQP

[5] Sankpal L.J., Patil S.H. Rider-Rank Algorithm-Based Feature Extraction for Re-ranking the Webpages in the Search Engine. *The Computer Journal*. 2020;63(10):1479-1489. https://doi.org/10.1093/comjnl/bxaa032

[6] Patel P., Patel K. A Review of PageRank and HITS Algorithms. *International Journal of Advance Research in Engineering, Science & Technology*. 2015;2(1):2394-2444.

[7] Tagarov B.Zh. The development of the market of search optimization in Russia. *Creative Economy*. 2018;12(9):1373-1384. (In Russ., abstract in Eng.) https://doi.org/10.18334/ce.12.9.39379

[8] Latypov A.R. Review of the impact of user behavior in search algorithms. *Sovremennye materialy, tehnika i tehnologii.* 2015;(2):92-97. (In Russ., abstract in Eng.) EDN: UNUUBP

[9] Brin S., Page L. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*. 1998;30(1-7):107-117. https://doi.org/10.1016/S0169-7552(98)00110-X

[10] Vasyaeva N.S., Degaev M.N. Formalization of an index construction model for search engines. *International Research Journal*. 2022;(6-1):56-60. (In Russ., abstract in Eng.) https://doi.org/10.23670/IRJ.2022.120.6.007

[11] Pang L., Xu J., Ai Q., Lan Y., Cheng X., Wen J. SetRank: Learning a Permutation-Invariant Ranking Model for Information Retrieval. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20). New York, NY, USA: Association for Computing Machinery; 2020. p. 499-508. https://doi.org/10.1145/3397271.3401104

[12] Zherdeva M.V., Artyushenko V.M. Stemming and lemmatization in Lucene.Net. *Lesnoy Vestnik* = Forestry Bulletin. 2016;20(3):131-134. (In Russ., abstract in Eng.) EDN: WKNMTN

[13] Thota P., Ramez E. Web Scraping of COVID-19 News Stories to Create Datasets for Sentiment and Emotion Analysis. In: Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference (PETRA'21). New York, NY, USA: Association for Computing Machinery; 2021. p. 306-314. https://doi.org/10.1145/3453892.3461333

[14] Sorokin V.E. Fuzzy Data Storing and Efficient Processing in PostgreSQL DBMS. *Software & Systems*. 2017;30(4):609-618. (In Russ., abstract in Eng.) https://doi.org/10.15827/0236-235X.030.4.609-618

[15] Avci C., Tekinerdogan B., Athanasiadis I.N. Software architectures for big data: a systematic literature review. *Big Data Analytics*. 2020;5:5. https://doi.org/10.1186/s41044-020-00045-1

[16] Xiaojie X., Yuan F., Jian W. The Basic Principle and Applications of the Search Engine Optimization. In: Du Z. (ed.) Proceedings of the 2012 International Conference of Modern Computer Science and Applications. Advances in Intelligent Systems and Computing. Vol. 191. Berlin, Heidelberg: Springer; 2013. p. 63-69. https://doi.org/10.1007/978-3-642-33030-8_11

[17] Lehmann C., Goren Huber L., Horisberger T. et al. Big Data architecture for intelligent maintenance: a focus on query processing and machine learning algorithms. *Journal of Big Data*. 2020;7:61. https://doi.org/10.1186/s40537-020-00340-7

[18] Lee D., Camacho D., Jung J.J. Smart Mobility with Big Data: Approaches, Applications, and Challenges. *Applied Sciences.* 2023;13(12):7244. https://doi.org/10.3390/app13127244

[19] Sparck Jones K., Walker S., Robertson S.E. A probabilistic model of information retrieval: development and comparative experiments: Part 1. *Information Processing & Management*. 2000;36(6):779-808. https://doi.org/10.1016/S0306-4573(00)00015-7

[20] Liu C., Chen Z., Cao D., Shang M. Application of Recommender System in Intelligent Community under Big Data Scenario. In: Proceedings of the 2nd International Conference on Big Data Technologies (ICBDT'19). New York, NY, USA: Association for Computing Machinery; 2019. p. 92-96. https://doi.org/10.1145/3358528.3359551

[21] Sun Z., Huo Y. A Managerial Framework for Intelligent Big Data Analytics. In: Proceedings of the 2nd International Conference on Software Engineering and Information Management (ICSIM'19). New York, NY, USA: Association for Computing Machinery; 2019. p. 152-156. https://doi.org/10.1145/3305160.3305211

[22] Serrano W. A Big Data Intelligent Search Assistant Based on the Random Neural Network. In: Angelov P., Manolopoulos Y., Iliadis L., Roy A., Vellasco M. (eds.) Advances in Big Data. INNS 2016. Advances in Intelligent Systems and Computing. Vol. 529. Cham: Springer; 2017. p. 254-261. https://doi.org/10.1007/978-3-319-47898-2_26

[23] Sanderson M. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends in Information Retrieval*. 2010;4(4):247-375. https://doi.org/10.1561/1500000009

[24] Chakraborty N., La Gatta V., Moscato V., Sperlì G. Information retrieval algorithms and neural ranking models to detect previously fact-checked information. *Neurocomputing*. 2023;557:126680. https://doi.org/10.1016/j.neucom.2023.126680

[25] Sun Z. Intelligent Big Data Analytics: A Managerial Perspective. In: Sun Z. (ed.) Managerial Perspectives on Intelligent Big Data Analytics. Hershey, PA: IGI Global; 2019. p. 1-19. https://doi.org/10.4018/978-1-5225-7277-0.ch001

Submitted 29.11.2022; approved after reviewing 27.01.2023; accepted for publication 21.02.2023.

About the authors:

Irina F. Astachova, Professor of the Chair of Computer Hardware, Faculty of Applied Mathematics, Informatics and Mechanics, Voronezh State University (1 Universitetskaya pl., Voronezh 394018, Russian Federation), Dr. Sci. (Eng.), Professor, **ORCID: https://orcid.org/0000-0002-2627-8508,** astachova@list.ru

Katerina A. Makoviy, Associate Professor of the Department of Control Systems and Information Technologies in Construction, Faculty of Information Technologies and Computer Security, Voronezh State Technical University (84, 20 letiya Oktyabrya St., Voronezh 394006, Russian Federation), Cand. Sci. (Tech.), **ORCID: https://orcid.org/0000-0003-3921-6047,** makkatya@mail.ru

Lev S. Nikitin, student of the Chair of Computer Hardware, Faculty of Applied Mathematics, Informatics and Mechanics, Voronezh State University (1 Universitetskaya pl., Voronezh 394018, Russian Federation), **ORCID: https://orcid.org/0009-0007-7962-261X**

Yuliya V. Khitskova, Associate Professor of the Department of Regional Economics and Territorial Administration of the Faculty of Economics, Voronezh State University (1 Universitetskaya pl., Voronezh 394018, Russian Federation), Cand. Sci. (Econ.), **ORCID: https://orcid.org/0000-0001-6322-8633,** prosvetovau@list.ru

All authors have read and approved the final manuscript.