Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторные работы по курсу «Информационный поиск»

Студент: А. С. Лупанова
Преподаватель: А. А. Кухтичев
Группа: М8О-401Б-22

Дата:
Оценка:
Подпись:

Москва, 2025

Лабораторная работа №1 «Добыча корпуса документов»

В качестве источников для сбора документации была выбрана биологическая тематика, что обусловлено её актуальностью и возможностью работы с разнообразными текстами. Для этого были использованы два источника данных: Википедия и Большая Российская Энциклопедия (БРЭ).

Википедия является одним из самых популярных и доступных источников информации. Она охватывает широкий спектр тем, в том числе биологию, и обновляется на регулярной основе, что делает её актуальной. В Википедии статьи подвергаются коллективному редактированию, что повышает вероятность наличия достоверной информации. Одним из преимуществ Википедии является наличие статей на разных языках, что позволяет легко расширять корпус, если в будущем потребуется добавление данных на других языках.

Большая Российская Энциклопедия (БРЭ) — это авторитетный источник, предоставляющий качественную информацию по широкому спектру научных дисциплин, включая биологию. Использование БРЭ позволяет добавить в корпус более глубинные статьи, доступные только в рамках энциклопедии, а не в открытых источниках.

1 Характеристики корпуса

После загрузки и очистки данных корпус включает в себя:

- Общее количество документов: 32 157
- Из Википедии: 28 434 документа
- Из БРЭ: 3 723 документа
- Общий размер данных (raw_html): 2.86 ГБ
- Средний размер одного документа: 89.02 КБ
- Общий размер clean_text: 707.8 МБ
- Средний размер clean_text одного документа: 22.01 КБ

2 Метаинформация и разметка документов

Каждый документ в корпусе включает в себя важную метаинформацию, такую как:

- URL документа: Нормализованный URL, который позволяет легко идентифицировать и ссылаться на статьи.
- Текст документа:
  - `raw_html` — оригинальный HTML-текст документа.
  - `clean_text` — очищенная версия текста, которая используется для индексации и поиска.
- **Метаданные**:
  - `title` — заголовок статьи.
  - `categories` — категории для Википедии.
  - `language` — язык документа.

Тексты документов были очищены от HTML-тегов и других ненужных элементов, чтобы сохранить только саму информацию.

3 Использование поисковых систем

Для более глубокого анализа биологической тематики важно уметь задавать более сложные поисковые запросы, чтобы точнее извлекать релевантную информацию.

Примеры запросов:

- "Эволюция человека и влияние окружающей среды на адаптацию"— запрос, требующий извлечения информации о взаимодействии эволюции и экологии.
- "Клеточные механизмы в биологии старения"— сложный запрос для поиска научных материалов о молекулярных и клеточных аспектах старения.
- "Биологическая роль микробиома человека в иммунной системе"— запрос, который требует понимания взаимосвязей между микробиотой и иммунитетом.

Каждый из этих запросов требует точности в формулировке, чтобы получить нужную информацию.

Недостатки существующих поисковых систем:

- Встроенный поиск Википедии:
  - Ограниченный синтаксис запросов. Для более сложных запросов, таких как использование нескольких логических операторов и сочетаний категорий, встроенный поиск не всегда даёт точные результаты.

  - Недостаточная спецификация тем. Например, запрос "микробиома человека"может вернуть страницы о микробах вообще, а не о человеческом микробиоме в частности.

- Google Search с ограничением по сайту:
  - Google, хотя и позволяет задать ограничение по сайту, часто включает в результаты материалы, не полностью соответствующие запросу, особенно если запрос сложный и многозначный.
  - В поисковой выдаче может смешиваться информация из разных областей биологии, например, результаты, связанные с генетикой и молекулярной биологией, могут быть перепутаны, что делает результаты менее релевантными для специфических научных запросов.

Лабораторная работа №2 «Поисковый робот»

1 Основные функции робота

Поисковый робот состоит из нескольких ключевых функций:

- **Инициализация и настройка робота**: На основе конфигурационного файла робот инициализирует необходимые параметры, такие как задержка между запросами, максимальная глубина обхода, минимальное количество слов в статье и другие.
- **Рекурсивный обход**: Робот рекурсивно обходит категории на Википедии или страницы на БРЭ, извлекая URL-адреса страниц для дальнейшей обработки.
- **Загрузка и парсинг страниц**: После извлечения URL роботом загружается HTML-страница и передается в парсер, который очищает страницу от ненужной разметки и извлекает текст.
- **Сохранение данных в MongoDB**: После обработки данных робот сохраняет их в базе данных MongoDB.
- **Обновление данных**: Робот проверяет, нужно ли обновлять уже сохраненные данные, основываясь на времени последнего обновления и изменениях на странице.
- **Поддержка продолжения работы**: Робот сохраняет текущее состояние и может быть остановлен и возобновлен с того места, где был остановлен.

2 Реализация поиска по Википедии

Поиск по Википедии осуществляется с использованием библиотеки `pywikibot`, которая позволяет программно взаимодействовать с Wikimedia API. Робот начинает с заданной категории и рекурсивно обходит все страницы в категории "Биология"до заданной глубины, извлекая статьи и их метаданные.

3 Реализация поиска по БРЭ

Для сбора URL статей используется `sitemap.xml`, который содержит ссылки на статьи. Робот извлекает все URL статей и передает их в очередь на обработку.

4 Запись данных в базу данных

Каждый документ содержит следующие поля:

- **url**: Оригинальный URL страницы, с которой был извлечен текст.
- **normalized_url**: Нормализованный URL, который представляет собой URL без параметров и фрагментов.
- **raw_html**: Сырой HTML-код страницы, полученный до обработки.
- **clean_text**: Очищенный текст, извлеченный из HTML-страницы.
- **word_count**: Количество слов в статье.
- **crawled_at**: Время обкачки документа в формате Unix timestamp.
- **updated_at**: Время последнего обновления документа в базе данных.
- **metadata**: Метаинформация о документе, включая заголовок страницы и категории.

Лабораторная работа №3 «Токенизация»

Текст разбивается на токены, включая как отдельные слова, так и более сложные элементы, такие как термины с дефисами и числа. Для обработки корпуса, содержащего биологическую тематику, необходимо учитывать специфическую терминологию.

1 Правила токенизации

Основные правила, используемые для токенизации:

- Текст разбивается на токены по пробелам и знакам препинания.
- Слова с дефисами сохраняются как единичные токены.
- Числа и даты также считаются отдельными токенами.
- Апострофы в словах сохраняются в качестве части слова.
- Токены приводятся к нижнему регистру.
- Специальные символы (например, &) заменяются на соответствующие символы (&).

2 Пример токенизации

Пример исходного текста и его токенов:

> Исходный текст: "Биология — это наука, изучающая жизнь. Взаимодействие живых существ с окружающей средой изучает биолог."

После применения процесса токенизации:

- "биология"
- "это"
- "наука"
- "изучающая"
- "жизнь"
- "взаимодействие"

- "живых"
- "существ"
- "с"
- "окружающей"
- "средой"
- "изучает"
- "биолог"

3 Статистические данные

- **Общее количество токенов**: 26 863 959
- **Средняя длина токена**: 7.27 символов
- **Время выполнения токенизации**: 31.73 секунд
- **Скорость токенизации**: 31 551.19 КБ/с
- **Скорость токенов**: 1 992 698 токенов/с

Лабораторная работа №4 «Стемминг»

Стемминг — это процесс приведения слов к их корню, который помогает избежать обработки разных словоформ как отдельных слов. В данной задаче был использован алгоритм стемминга на основе алгоритма Портера с модификациями для обработки специфической биологической терминологии, такой как названия видов, латинские термины и научные слова.

1 Правила стемминга

Для слов, содержащих кириллические буквы, используется алгоритм стемминга, который удаляет общие суффиксы и приводит слово к его основе. Например:

- Суффиксы для глаголов: `ивши`, `вшись`, `ся`, `сь`.
- Суффиксы для прилагательных: `ее`, `ые`, `ого`, `ими`.
- Суффиксы для существительных: `ов`, `ев`, `ий`, `ям`.

Все буквы `ё` заменяются на `е` для уменьшения шума в процессе стемминга.

Числовые токены исключаются из дальнейшего стемминга, так как они не требуют изменения.

В случае латинских и других символов, токены остаются без стемминга.

2 Пример стемминга

Исходные токены и результат их обработки с помощью стемминга представлены в таблице ниже:

Исходный токен	**Токен после стемминга**
тюлень	тюлен
морское	морск
млекопитающее	млекопит
семейства	семейст
буровато-серая	буровато-сер

Таблица 1: Пример стемминга

3 Статистические данные

- **Количество токенов до**: 26 863 959
- **Количество токенов после**: 24 527 256
- **Числовые токены**: 2 341 197
- **Измененные токены**: 20 154 263

Для оценки качества поиска после внедрения стемминга было проведено несколько тестов. Были сравнины результаты поиска до и после применения стемминга для различных типов запросов.

В результате было замечено, что в некоторых случаях стемминг улучшил результаты поиска, так как теперь все формы слова приводятся к единому корню. Однако, в некоторых случаях, особенно для специфических биологических терминов, стемминг привел к ухудшению точности, так как специфические термины были преобразованы в более общие формы.

Лабораторная работа №5 «Закон Ципфа»

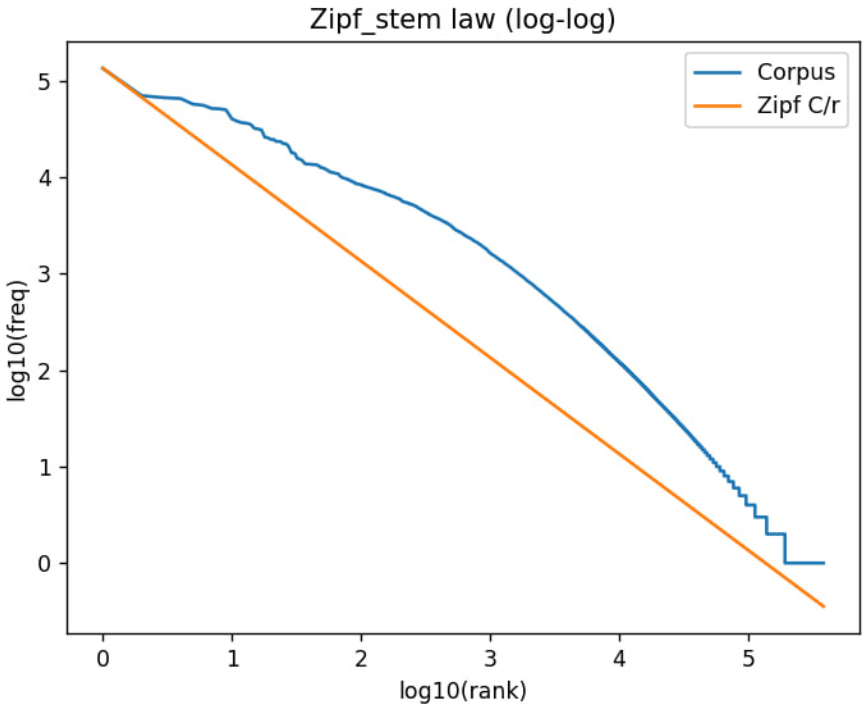

Рис. 1: Распределение терминов по частотам и линия Ципфа

В процессе выполнения работы был построен график распределения частот терминов по логарифмической шкале. График строится с использованием данных из файла, содержащего частоты терминов, которые были извлечены в процессе токенизации и стемминга. Линия Ципфа накладывается на график для оценки того, насколько хорошо распределение терминов соответствует этому закону.

Из построенного графика видно расхождение с идеальной линией. В корпусе содержится большое количество специализированных терминов, которые используются реже, чем стандартные слова, что вызывает отклонения от теоретического распределения. Стемминг и токенизация могут приводить к удалению редких слов или изменению их формы, что также влияет на частотное распределение.

Лабораторная работа №6 «Булев индекс»

- **Прямой индекс**: Для каждого документа сохраняется его идентификатор (`doc_id`, 8 байт), заголовок (`title`, длина строки + сама строка), и URL (`url`, длина строки + сама строка).

  Прямой индекс сохраняется в том порядке, в котором документы были обработаны, так как он используется для быстрого доступа к информации о каждом документе.

- **Обратный индекс**: Для каждого термина сохраняется его длина (`term_size`, 4 байта), сам термин, количество документов, в которых он встречается (`doc_count`, 4 байта), и список идентификаторов документов.

  Для обратного индекса использована сортировка терминов в лексикографическом порядке, а затем — по частоте их появления в документах. Это позволяет эффективно искать термины и соответствующие им документы.

Формат представления данных расширяем, что позволяет в будущем добавить дополнительные поля или оптимизации.

1 Статистические данные

- **Количество терминов**: 10 543 916
- **Средняя длина терма**: 6.42 символов
- **Средняя длина токена**: 6.58 символов
- **Общее количество токенов**: 24 527 256
- **Время индексации**: 162.89 секунды
- **Скорость индексации**: 150,8 токенов в секунду
- **Скорость индексации на один документ**: 763.3 токенов на документ

Итоговый индекс занимает 416 МБ

Лабораторная работа №7 «Булев поиск»

Система принимает запросы, состоящие из терминов и логических операторов.

- Пробел или два амперсанда `&&` — логическая операция `И`.
- Две вертикальные палочки `||` — логическая операция `ИЛИ`.
- Восклицательный знак `!` — логическая операция `НЕ`.
- Скобки `()` используются для группировки выражений.

1 Алгоритм работы парсера

Парсер выполняет следующие шаги:

- Разделение входного запроса на токены.
- Обработка логических операций: `AND`, `OR`, `NOT`.
- Применение операций поочередно, начиная с операндов.
- Обработка скобок для изменения порядка выполнения операций.
- Вывод результата, который содержит список идентификаторов документов, соответствующих запросу.

2 Пример запросов

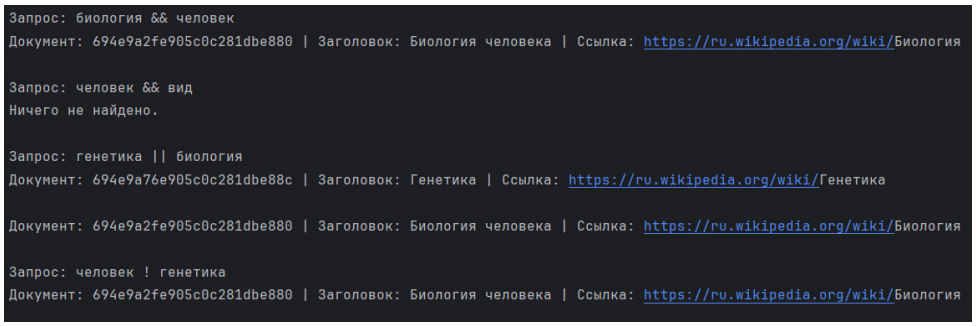

Рис. 2: Результат работы Булева индекса

Среднее время выполнения запроса: 5.1 милисекунд

Запуск системы поискового робота

Запуск поискового робота выполняется с помощью команд:

```
docker compose build
docker compose up
```

Все логи обхода станиц будут отображатся в консоли

Для того, чтобы запустить "серверы"tokenazer,stemmer,zipf и indexer нужно ввести команду:

```
docker compose --profile name run --rm name
```

В консоли отобразятся логи и статистика работы сервисов

Для того, чтобы запустить бинарный поиск (searching) требуется запустить команду:

```
cat test1.txt | docker-compose --profile searching run --rm searching
```

Выводы

В ходе работы были решены все поставленные задачи, включая создание поискового робота, построение булевого индекса и реализацию поиска. Работа системы продемонстрировала высокую скорость выполнения запросов и хорошую точность поиска, хотя были выявлены определённые ограничения, связанные с возможностью расширения корпуса и увеличением объема данных.

Кроме того, проведённый анализ и тестирование показали, что предложенные методы и алгоритмы, такие как токенизация, стемминг и булев индекс, являются эффективными для обработки биологической тематики, однако для более сложных научных запросов потребуется дополнительная настройка и оптимизация. В будущем можно улучшить систему, расширив её функциональность для более точного поиска и обработки специфических терминов, а также повысить производительность за счёт использования более мощных алгоритмов индексирования и параллельной обработки данных.

Список литературы

[1] Маннинг, Рагхаван, Шютце. *Введение в информационный поиск* — Издательский дом «Вильямс», 2011. Перевод с английского: доктор физ.-мат. наук Д. А. Клюшина — 528 с. (ISBN 978-5-8459-1623-4 (рус.)).

[2] Introduction to Information Retrieval [Электронный ресурс] // GitHub Gist. — URL: https://gist.github.com/AaradhyaSaxena/e20e7dd1556683523f58fe3d5cc463d2 (дата обращения: 28.12.2025).

[3] Как поисковые системы оценивают тексты: закон Ципфа, индекс туманности и прочее [Электронный ресурс] // Pitcher.agency. — URL: https://pitcher.agency/blog/zakon-cipfa (дата обращения: 28.12.2025).

[4] Основы полнотекстового поиска в ElasticSearch. Часть вторая [Электронный ресурс] // Хабр: Sportmaster Lab. — URL: https://habr.com/ru/companies/sportmaster$_l$*ab*/*articles*/756270/(: 28.12.2025).

[5] Как устроен поиск [Электронный ресурс] // Хабр: HeadHunter. — URL: https://habr.com/ru/companies/hh/articles/413261/ (дата обращения: 28.12.2025).