Московский авиационный институт
(национальный исследовательский университет)

Институт №8 «Компьютерные науки и прикладная математика»

**Отчёт по лабораторным работам
по курсу «Информационный поиск»**

Студент: Полиха А. В.
Преподаватель: Кухтичев А. А.
Группа: М8О-403Б-22
Дата:
Оценка:
Подпись:

Москва, 2025

Лабораторная работа №1 «Добыча корпуса документов»

В рамках первой лабораторной работы был выполнен выбор и анализ корпуса документов, который будет использоваться при выполнении всех последующих лабораторных работ.

1 Задание

Разработать поисковую систему для собственного корпуса документов по выбранной тематике - музыканты, певцы, их биография. Для этого требуется:

- Поисковой робот crawler, который собирает данные из выбранных источников, работающий в соответствии с robotx.txt;
- Хранилище документов - MongoDB;
- Выделение текста из сырого html с последующей токенизацией и стеммингом;
- Булев индекс;
- Булев поиск с операторами AND, OR, NOT, скобки;

В качестве первого источника был выбран набор статей из Wikipedia, а именно биографии музыкальных деятелей. Данный ресурс известен большим количеством структурированных текстов и наличием развитой внутренней разметки.

В качестве второго источника был выбран сайт *belcanto*, содержащий около 5 000 биографий. Для сравнения, в Wikipedia подобных статей насчитывается более 30 000.

2 Формат сырых документов

Сырые документы сохраняются в виде отдельных файлов в формате JSON (один файл — один документ). Каждый документ содержит основные поля:

- id документа;
- url - нормализованный URL;
- title - заголовок страницы;
- html_content - исходный HTML-код страницы;
- source_name - источник;
- crawl_date - время обкачки в Unix timestamp;

- content_hash - хэш, чтобы понимать, нужно ли обкачивать еще раз.

3 Характеристики корпуса

- Общее количество документов - 51 722;
- Wikipedia - 45 190;
- Belcanto - 6 532;
- Размер данных - 5.719 GB;
- Средний размер документа - 113.23 KB;
- Среднее количество символов - 10 287.

4 Существующие поисковые системы

- Google - поиск по запросу с ограничением сайта site:ru.wikipedia.org
- Встроенный поиск Wikipedia - поиск по статьям;

Недостатки Google:

- Неопределенность результатов: Использование общих фраз может привести к тому, что результаты будут включать много нерелевантных страниц.
- Фильтрация информации: Google может не показывать все страницы, особенно если они не имеют высокой релевантности или популярности.
- Актуальность данных: Результаты могут содержать устаревшую информацию, поскольку индекс Google обновляется не мгновенно.

Недостатки поиска Wikipedia:

- Ограниченность по языку: Внутренний поиск может выдавать результаты только на выбранном языке, что ограничивает доступ к международной информации.
- Поиск по названиям: Внутренний поиск может не находить статьи, если они названы не так, как ожидается, что затрудняет поиск менее известных артистов.

Лабораторная работа №2 «Поисковой робот»

1 Цель работы

Целью данной лабораторной работы являлась разработка поискового робота, осуществляющего автоматический сбор документов из выбранных источников.

Поисковой робот был реализован на языке Python. В процессе его разработки особое внимание уделялось соблюдению правил, описанных в файле `robots.txt`, а также корректной обработке HTTP-запросов.

Робот выполняет следующие этапы:

- формирование очереди URL для обхода;
- загрузка HTML-документов;
- обработка кодировки страниц;
- сохранение исходных HTML-файлов;
- запись мета-информации (URL, источник, дата загрузки).

Результатом работы поискового робота является набор сырых HTML-документов, которые в дальнейшем используются для очистки, токенизации и индексирования. Если вдруг случится какой-то сбой, от которого наш робот умрет, то при запуске он продолжит с того же места, попутно проверяя, обновилась ли информация в уже существующих в базе документах.

2 Конфиг для робота

В конфиге есть секции:

- db — настройки базы данных MongoDB;
- logic — настройки логики робота, изменение ресурса для обкачки, таймаут, задержка обкачки и т. д.

Лабораторная работа №3 «Токенизация»

1 Правила тоценизациии

Токенизация представляет собой процесс разбиения текста документа на последовательность элементарных единиц — токенов.

Перед началом токенизации выполняется очистка HTML-документов: удаляются служебные теги, такие как `script`, `style`, `meta`, элементы навигации и рекламные блоки.

После очистки текста выполняются следующие операции:

- Извлечение текста из HTML: удаляются HTML-теги, скрипты и стили, извлекается только текстовое содержимое.
- Нормализация регистра: все символы приводятся к нижнему регистру. Для кириллицы реализована поддержка UTF-8 с корректной обработкой русских букв, включая букву Ё.
- Разбиение на токены: текст последовательно обрабатывается посимвольно с учетом UTF-8. Токены формируются из последовательностей букв и цифр, разделители (пробелы, знаки препинания) служат границами токенов.
- Фильтрация по длине: токены короче минимальной длины (по умолчанию 2 символа) отбрасываются.
- Удаление стоп-слов (опционально): при включенной опции удаляются частые служебные слова (русские и английские), не несущие смысловой нагрузки.

Особую сложность представляют слова с дефисами, например «абдаль-мумин», что требует применения дополнительных правил токенизации.

Лабораторная работа №4 «Стемминг»

1 Цель работы

Целью данной лабораторной работы являлось применение алгоритмов стемминга для нормализации словоформ.

Стемминг позволяет привести различные формы слова к общему основанию — стему. Это существенно уменьшает размер словаря и ускоряет процесс поиска.

Результат: основа слова, объединяющая различные словоформы.
Преимущества стемминга:

- уменьшение количества уникальных термов;
- ускорение поиска;
- снижение объёма индекса.

Недостатком стемминга является невозможность точного поиска по словоформе. Например, запрос «пианист» после стемминга может находить документы, содержащие слова «пианино» и «фортепиано».

2 Статистика

- Документов: 51,722
- Всего токенов: 53,295,593
- Уникальных токенов: 1,321,300
- Уникальных стемов: 835,856
- Средняя длина токена: 7.15 символов
- Размер индекса: 155.86 MB
- Скорость токенизации: 20812.43 КБ/сек

Лабораторная работа №5 «Закон Ципфа»

В рамках данной лабораторной работы была исследована статистическая закономерность, известная как закон Ципфа.

Закон Ципфа утверждает, что частота слова обратно пропорциональна его рангу в списке слов, отсортированных по убыванию частоты.

Для корпуса документов была построена зависимость частоты токенов от их ранга, что позволило подтвердить выполнение закона Ципфа на реальных текстовых данных.

Анализ распределения показал, что небольшое количество слов встречается крайне часто, в то время как большая часть слов имеет очень низкую частоту.

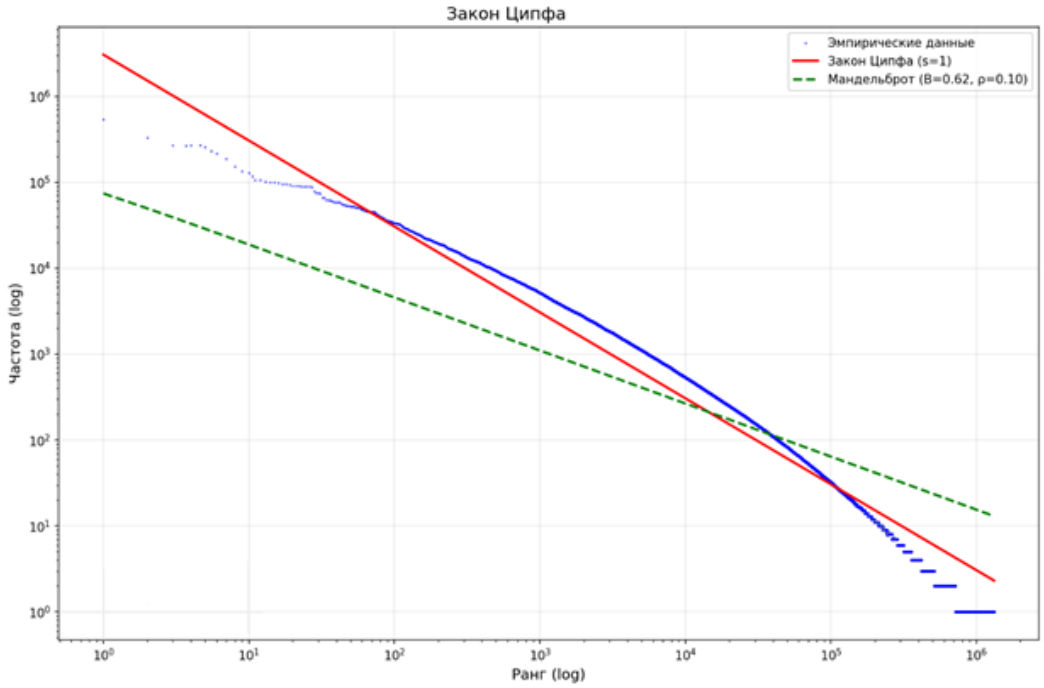

Рис. 1: Закон Ципфа

Лабораторная работа №7 «Булев индекс»

В данной лабораторной работе был реализован булев индекс, используемый для хранения информации о вхождении термов в документы.

Рассматривались два типа индексов:

- прямой индекс — хранит информацию о термах для каждого документа;
- обратный индекс — хранит список документов для каждого терма.

Лабораторная работа №8 «Булев поиск»

Целью лабораторной работы являлась реализация булевого поиска на основе построенного обратного индекса.
Были реализованы следующие логические операции:

- AND — логическое И;
- OR — логическое ИЛИ;
- NOT — логическое НЕ;
- группировка выражений с помощью круглых скобок.

Поисковый запрос преобразуется в логическое выражение, после чего над списками документов выполняются соответствующие операции.

Результатом поиска является список документов, удовлетворяющих заданному логическому условию, представленный в виде ссылок на источники.

Примеры запросов
Запрос "musician"

```
./search .\index.bin -q "musician"

=== Query: musician ===
=== Query: musician ===
Found: 680 in   87.13 ms

1.  Дебюсси. Двадцать четыре прелюдии (Préludes) | Belcanto.ru
   https://www.belcanto.ru/debussy_preludes.html

2.  Александер Кэмпбелл Маккензи | Belcanto.ru
   https://www.belcanto.ru/mackenzie.html

3.  Пабло де Сарасате | Belcanto.ru
   https://www.belcanto.ru/sarasate.html

4.  Густав Холст | Belcanto.ru
   https://www.belcanto.ru/holst.html

5.  Венсан д&#x27;Энди | Belcanto.ru
   https://www.belcanto.ru/indy.html

6.  Иегуди Менухин | Belcanto.ru
   https://www.belcanto.ru/menuhin.html

7.  Клара-Джуми Кан (Clara-Jumi Kang) | Belcanto.ru
   https://www.belcanto.ru/kang_clara.html

8. Абд аль-Кадир аль-Мараги
   https://ru.wikipedia.org/wiki/Абд_аль-Кадир_аль-Мараги

9. Абрамс, Грэйси
   https://ru.wikipedia.org/wiki/Абрамс,_Грэйси
```

Рис. 2: Пример запроса

Запрос "Басист"

```
>search index.bin -q "басист"

=== Query: басист ===
Found: 792 in   42.12 ms

1.  Орнетт Коулман | Belcanto.ru
   https://www.belcanto.ru/coleman.html

2.  Диззи Гиллеспи | Belcanto.ru
   https://www.belcanto.ru/dizzy.html

3. Коллинз, Аллен
   https://ru.wikipedia.org/wiki/Коллинз,_Аллен

4. Абё, Елена
   https://ru.wikipedia.org/wiki/Абё,_Елена

5. Yes (группа)
   https://ru.wikipedia.org/wiki/Yes_(группа)

6. Фолк-рок
   https://ru.wikipedia.org/wiki/Фолк-рок

7. The Yardbirds
   https://ru.wikipedia.org/wiki/The_Yardbirds

8. Led Zeppelin
   https://ru.wikipedia.org/wiki/Led_Zeppelin

9. The Firm
   https://ru.wikipedia.org/wiki/The_Firm
```

Рис. 3: Пример запроса

Запрос "!Русский певец"

```
>search index.bin -q "!русский певец"

=== Query: !русский певец ===
Found: 528 in   57.08 ms

1.  Луи Армстронг | Belcanto.ru
   https://www.belcanto.ru/armstrong_louis.html

2.  Жорж Брассенс | Belcanto.ru
   https://www.belcanto.ru/brassens.html

3.  Марио Дель Монако | Belcanto.ru
   https://www.belcanto.ru/delmonaco.html

4.  Алессандро Бончи | Belcanto.ru
   https://www.belcanto.ru/bonci.html

5.  Уго Бенелли (Ugo Benelli) | Belcanto.ru
   https://www.belcanto.ru/benelli.html
```

Рис. 4: Пример запроса

Запрос "(композитор музыкант) !русский инструменталист"

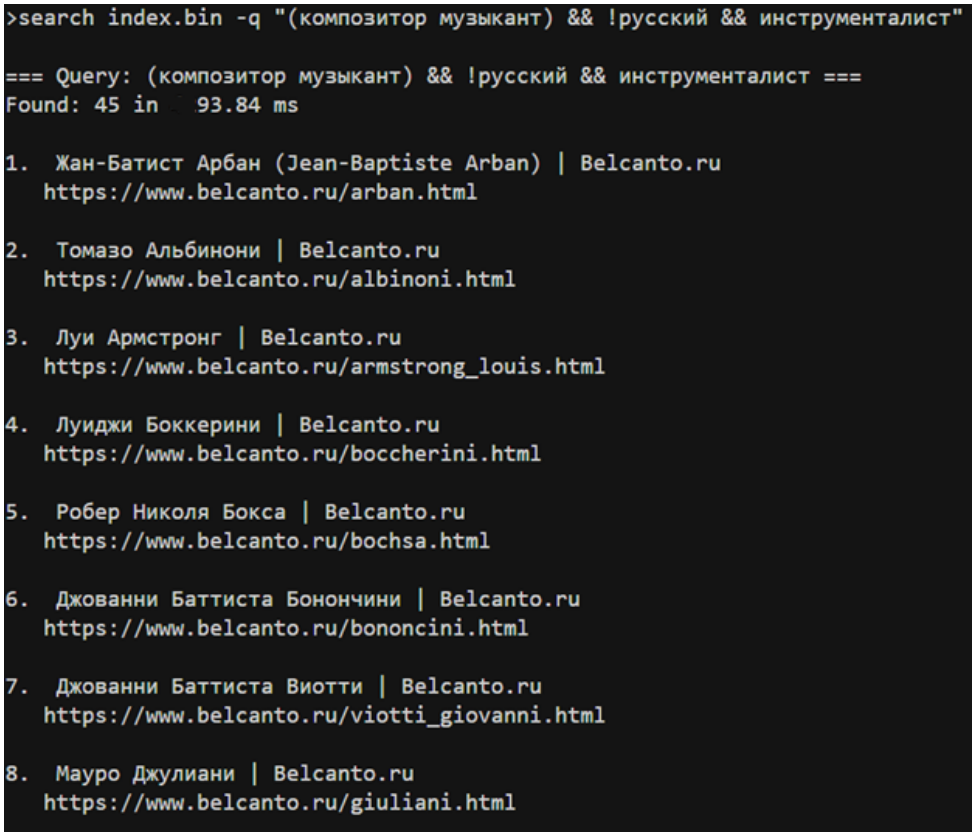

Рис. 5: Пример запроса

1 Выводы

В ходе выполнения лабораторных работ был изучен полный цикл построения поисковой системы.

Я получил практические навыки разработки поискового робота, очистки и нормализации текстовых данных, токенизации, стемминга, построения индексов и обработки поисковых запросов.

Список литературы

[1] Как работают поисковые системы / Christina29 // Habr.com (Блог компании Яндекс). — 27 авг. 2019. — URL: https://habr.com/ru/companies/yandex/articles/464375/ (дата обращения: 28.11.2025).

[2] Как устроен поиск / Habr.com (Блог компании hh). — URL: https://habr.com/ru/companies/hh/articles/413261/ (дата обращения: 28.11.2025).

[3] Алгоритмы поиска, обратный индекс — Часть 1 / Habr.com (Статьи). — URL: https://habr.com/ru/articles/53987/ (дата обращения: 29.11.2025).