МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский Авиационный Институт»

(Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

Отчет по лабораторной работе

по предмету «Информационный поиск»

«Поисковой движок»

Группа: М8О-412Б-22

Студент: Муратов А.А.

Оценка:

Дата сдачи:

Москва, 2025

Задание

Токенизация

Нужно реализовать процесс разбиения текстов документов на токены, который потом будет использоваться при индексации. Для этого потребуется выработать правила, по которым текст делится на токены. Необходимо описать их в отчёте, указать достоинства и недостатки выбранного метода. Привести примеры токенов, которые были выделены неудачно, объяснить, как можно было бы поправить правила, чтобы исправить найденные проблемы.

В результатах выполнения работы нужно указать следующие статистические данные:

Количество токенов.

Среднюю длину токена.

Кроме того, нужно привести время выполнения программы, указать зависимость времени от объёма входных данных. Указать скорость токенизации в расчёте на килобайт входного текста. Является ли эта скорость оптимальной? Как её можно ускорить?

Закон Ципфа

Для своего корпуса необходимо построить график распределения терминов по частотностям в логарифмической шкале, наложить на этот график закон Ципфа. Объяснить причины расхождения.

Стемминг

Добавить в созданную поисковую систему лемматизацию. В простейшем случае, это просто поиск без учёта словоформ.

Токенизация

Токенизация – это разбиение текста на токены (слова/термы) с нормализацией регистра и фильтрацией стоп-слов. Она формирует основу словаря и обратного индекса; сокращает шум (стоп-слова) и унифицирует написания.

Реализация:

- Файлы: labs/tokenizer.hpp, labs/tokenizer.cpp

- HTML очищается от тегов (`strip_tags`), токен — любая последовательность букв/цифр/UTF-8 байт; ASCII приводится к нижнему регистру.

- Фильтр стоп-слов для русского и служебных слов (URL, технические токены).

- Статистика: счётчики токенов, уникальных термов, длина, входные байты.

В данный момент реализация не оптимизирована под многопоточность и кэширование, которые могли бы увеличить скорость токенизации

Результат:

[STATS]

Documents: 33182

Tokens: 145429934

Unique: 568183

Avg len: 7.2075

Avg Speed: 672 Kb/s

Bytes in: 1734001972

Закон Ципфа

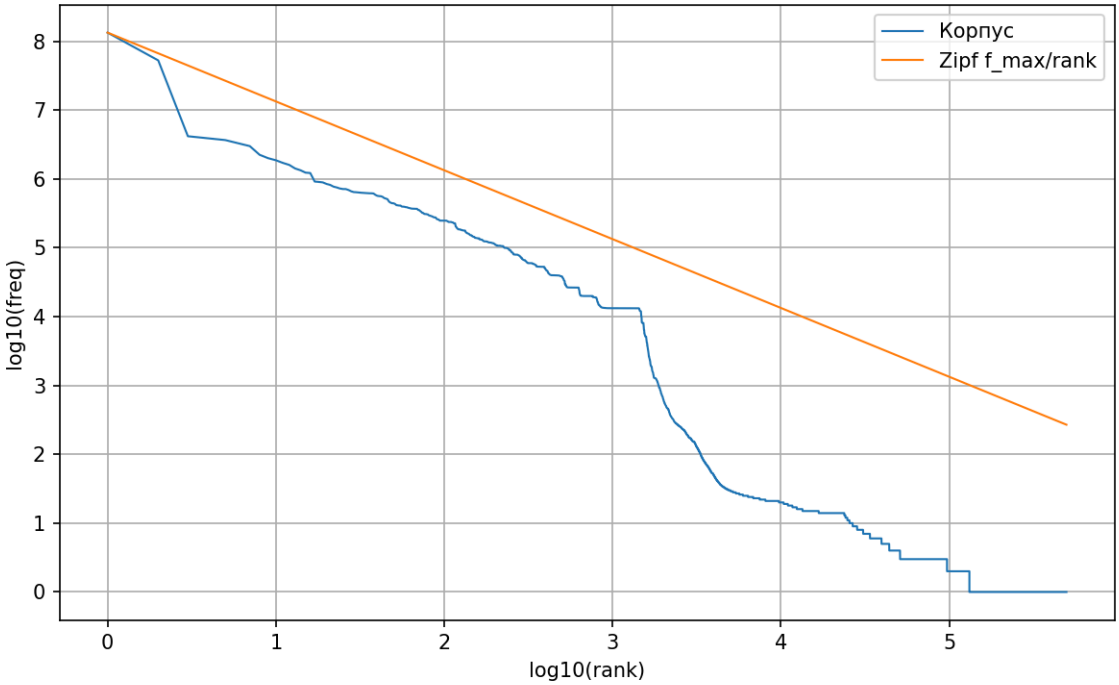
Закон Ципфа - это е правило, заключающееся в том, что частота терма обратно пропорциональна его рангу: f(r) = C / r^s. Он используется для проверки естественности корпуса, оценка распределения частот, подготовка данных для графиков.

Реализация

- Файлы: labs/zipf.hpp, labs/zipf.cpp

- Строится упорядоченный список термов по частоте, вычисляются log(rank), log(freq) и ожидаемая Zipf-кривая.

- Результат сохраняется в `labs/data/zipf.tsv`; график можно построить скриптом labs/graph.py.



Одной из причин расхождения могу назвать тематический перекос - некоторые темы, а значит и слова, могли попадать в корпус чаще других, и получилось расхождение.

Стемминг

Стемминг - это приведение словоформ к общей основе (stem) путём отсечения типичных суффиксов.Она уменьшает размер словаря, объединяет формы слов, улучшает полноту поиска.

Реализация

- Файлы: labs/stemmer.hpp, labs/stemmer.cpp.

- Простой суффиксный стриппер для русских/латинских окончаний ("овать", "ировать", "ами", "ость", и т.п.).

- Стем применяется при построении индекса и при обработке поискового запроса.

Булев индекс

Булев индекс - обратный индекс, хранящий для каждого терма список документов с частотами.Используется как поддержка быстрых булевых запросов (AND/OR), хранение DF/CF для аналитики.

Реализация

- Файлы: labs/index.hpp, labs/index.cpp.

- Структура `InvertedIndex`: `TokenInfo{cf, df, postings(doc_id, tf)}`; при добавлении документа его текст токенизируется, затем сохраняются только метаданные (title/url/source) для экономии памяти.

- Сохранение артефактов: `vocabulary.tsv` (ранг, терм, CF, DF), `inverted_index.tsv` (терм, doc_id, tf), `docs.tsv` (метаданные).

Булев индекс

Булев поиск – это поиск с операторами AND (`&`) и OR (`|`) над термами/строками.

Он позволяет формулировать точные запросы по нескольким условиям.

Реализация

- Файлы: labs/index.cpp — функция `search`.

- Разбор запроса: группы через `|` (OR между группами), внутри группы — `&` (AND).

- Термы нормализуются и стеммируются. Результаты пересекаются/объединяются с учётом групп; скоринг — сумма TF найденных термов в документе.

- В интерактивном режиме (`labs_app` без `--no-search`) выводится топ-10 по score.

Примеры поиска:

Enter boolean queries (use '&' for AND, '|' for OR). Empty line to exit.

> Сыр

1. doc 31324 score=15 | https://tass.ru/press/8196 | https://tass.ru/press/8196

2. doc 773 score=2 | https://www.interfax.ru/russia/1061981 | https://www.interfax.ru/russia/1061981

3. doc 17894 score=1 | https://www.interfax.ru/business/1033100 | https://www.interfax.ru/business/1033100

> Айти

1. doc 8076 score=2 | https://www.interfax.ru/russia/1044416 | https://www.interfax.ru/russia/1044416

2. doc 11282 score=1 | https://www.interfax.ru/business/1040046 | https://www.interfax.ru/business/1040046

> Налог

1. doc 30722 score=17 | https://tass.ru/press/9069 | https://tass.ru/press/9069

2. doc 27154 score=7 | https://tass.ru/press/16713 | https://tass.ru/press/16713

3. doc 20830 score=3 | https://tass.ru/ekonomika/25883857 | https://tass.ru/ekonomika/25883857

4. doc 671 score=2 | https://www.interfax.ru/russia/1062090 | https://www.interfax.ru/russia/1062090

5. doc 4070 score=2 | https://www.interfax.ru/business/1058453 | https://www.interfax.ru/business/1058453

6. doc 7342 score=2 | https://www.interfax.ru/business/1045189 | https://www.interfax.ru/business/1045189

7. doc 12700 score=2 | https://www.interfax.ru/russia/1038588 | https://www.interfax.ru/russia/1038588

8. doc 26468 score=2 | https://tass.ru/press/18145 | https://tass.ru/press/18145

9. doc 29186 score=2 | https://tass.ru/press/12389 | https://tass.ru/press/12389

10. doc 29677 score=2 | https://tass.ru/press/11327 | https://tass.ru/press/11327

... and 9 more

Выводы

Данная лабораторная работа подразумевала создание поискового движка на языке C++, с токенизацией, стеммингом, подходящим корпусом документов, булевым поиском и индексом. Итоговая программа, созданная мной, отделяет в корпусе слова, приводит их к единой форме, использует их для булевого индекса. Булев поиск же позволяет эффективно найти необходимые документы в уже обработанном программой корпусе. Я считаю все пункты выполненными в полном объёме.