Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторные работы по курсу «Информационный поиск»

Студент: И. В. Сектименко
Преподаватель: А. А. Кухтичев
Группа: М8О-410Б

Дата:
Оценка:
Подпись:

Москва, 2025

Лабораторная работа №1 и 5 «Добыча корпуса документов» и «Поисковой робот»

1 Описание

В качестве источника данных был выбран корпус документов, собранный с сайта `litres.ru`. Каждый документ в корпусе соответствует одной книге и состоит из следующих элементов:

- **Заголовок** — URL-ссылка на страницу книги (использовалась как уникальный идентификатор);
- **Текст документа** — объединённые название книги, её аннотация (описание) и тексты пользовательских комментариев.

Сайт `litres.ru` динамически подгружает часть контента (в частности, комментарии и расширенное описание) с помощью JavaScript. Это потребовало взаимодействия с логической частью сайта через инструменты автоматизации браузера. Для сбора данных был написан скрипт на языке Python с использованием библиотеки `Selenium`. Сбор данных осуществлялся выборочно: была выбрана подборка книг из категории «лёгкое чтение», чтобы обеспечить относительную однородность корпуса и управляемый объём данных.

Все полученные данные были сохранены в локальный текстовый файл в формате «один документ — одна строка», где каждая строка представляет собой JSON-объект с полями `id` (URL), `title`, `description` и `comments`. В дальнейшем этот файл использовался как исходный корпус для последующих лабораторных работ.

2 Статистика корпуса

После предварительной обработки (удаления HTML-разметки, нормализации пробелов и переходов на новую строку) были получены следующие статистические характеристики корпуса:

- Размер «сырых» данных (до очистки): **2,1 ГБ**;
- Количество документов: **31 683**;
- Общий объём извлечённого текста (после очистки): **1,4 ГБ**;
- Средний размер одного документа (в байтах): **44 200 байт**;
- Среднее количество слов в документе: **1 062**.

3 Существующие поисковые системы

Для поиска по выбранному корпусу доступны следующие поисковые инструменты:

- **Встроенный поиск сайта** `litres.ru` — позволяет искать книги по названию, имени автора или ISBN. Однако он не индексирует полный текст описания или пользовательские комментарии, а также не поддерживает семантический или контекстный поиск по содержанию книги.
- **Поисковая система Google** — поддерживает ограниченный поиск по конкретному домену с помощью оператора `site:litres.ru`. Например, запрос
  `site:litres.ru "утопия будущего"`
  вернёт страницы на `litres.ru`, содержащие указанную фразу.

Примеры поисковых запросов и результатов приведены на рисунках 1 и 2.

Недостатки существующих решений:

- Поиск на `litres.ru` ограничен метаданными (название, автор) и не учитывает содержание аннотаций и отзывов;
- Использование Google требует знания специальных операторов (например, `site:`), что неочевидно для большинства пользователей;
- Ни один из существующих поисковиков не позволяет искать по ключевым сюжетным поворотам, эмоциональной окраске или тематике, упомянутой в комментариях;
- Релевантность выдачи часто низка: например, запрос «книга про дружбу кошки и собаки» может не вернуть подходящую художественную литературу, если эта фраза отсутствует дословно в названии или аннотации.

Таким образом, выбранный корпус документов подходит для выполнения последующих лабораторных работ, так как он доступен, структурирован, содержит как метаданные, так и текстовое содержимое, и одновременно демонстрирует недостатки существующих поисковых систем — что делает задачу создания собственного поискового движка актуальной.

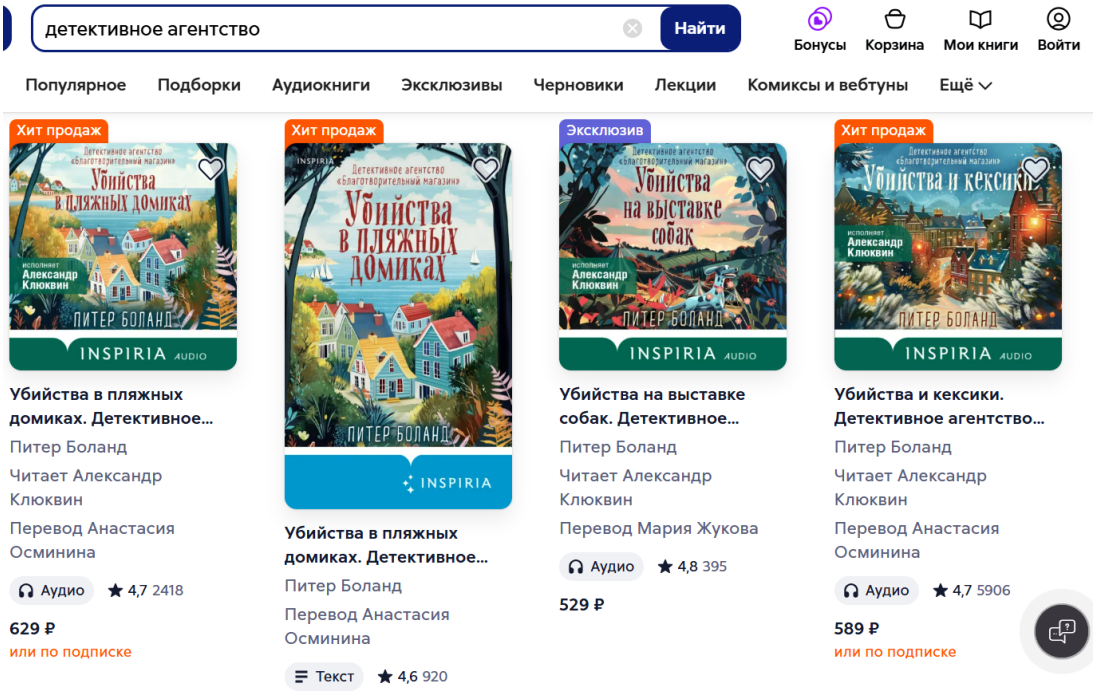

Рисунок 1 - Пример поиска на сайте `litres.ru`

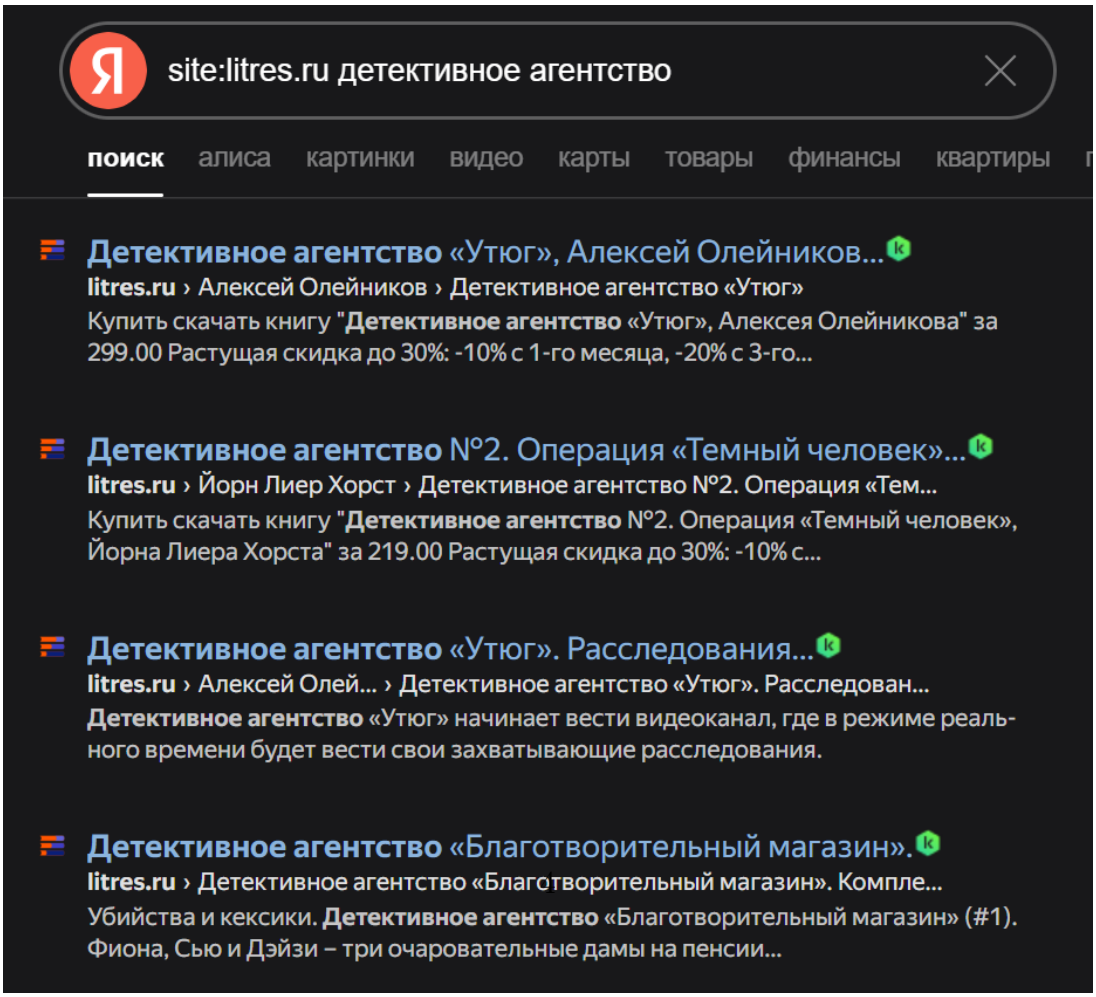

Рисунок 2 – Пример поиска в Google

4 Выводы

Выполнив первую и вторую лабораторные работы по курсу «Информационный поиск», я научилась самостоятельно собирать и структурировать корпус документов из реального веб-источника. Я освоила базовые методы веб-скрапинга с учётом динамической загрузки контента, познакомилась с проблемами, связанными с масштабированием сбора данных (медленная загрузка, блокировки, ограничения сайта). Кроме того, я получила практическое понимание того, как устроены публичные поисковые системы, и убедилась в их ограниченности при решении задач контекстного или тематического поиска. Эти навыки будут полезны при разработке собственного поискового движка, индексации текстов, а также в любых задачах, связанных с обработкой и анализом больших объёмов неструктурированного текста.

Лабораторная работа №2-3 «Булевый индекс» и «Булевый поиск»

1 Описание

Для реализации булева поиска была разработана система, способная обрабатывать сложные запросы с использованием логических операторов: (И), `||` (ИЛИ), `!` (НЕ). Допускаются пробелы, амперсанды, вертикальные палочки и восклицательный знак; также поддерживаются скобки для задания приоритета операций.

Процесс выполнения запроса состоит из следующих этапов:

1. **Парсинг запроса** — входная строка анализируется на наличие логических операторов. Запрос разбивается на положительные (обязательные) и отрицательные (запрещённые) токены.

2. **Стемминг токенов** — каждый извлечённый токен проходит через ту же функцию стемминга, что и при индексации документов (описано в следующих лабораторных работах). Это обеспечивает единообразие представления слов в индексе и запросе.

3. **Формирование фильтра** — на основе обработанных токенов формируется логический фильтр, который передаётся в базу данных, реализованную на основе прямого индекса).

4. **Выполнение поиска** — система возвращает список документов, удовлетворяющих условиям запроса. В качестве результата пользователь получает ссылки на документы (URL-адреса книг с сайта `litres.ru`), соответствующие его запросу.

Для демонстрации функциональности выводится 50 документов в виде ссылок на оригинальные страницы.

2 Статистика

Скорость выполнения запросов

Поиск выполняется за **8 мс** на среднем документе (при тестировании на корпусе из 31 683 документов). Это свидетельствует о высокой эффективности реализованного алгоритма и структур данных.

Тестирование корректности

Корректность поисковой выдачи проверялась путём сравнения результатов с ручным анализом содержимого документов. Для каждого запроса были выбраны контрольные документы, и проверялось, попадают ли они в выдачу. Также проводилось тестирование граничных случаев: пустой запрос, запрос с одним символом, запрос только из отрицаний.

3 Выводы

Выполнив лабораторные работы 2 и 3 по курсу «Информационный поиск», я научилась реализовывать булев поисковый движок с поддержкой логических операторов и скобок, разрабатывать устойчивый парсер запросов, игнорирующий лишние пробелы и допускающий незначительные ошибки ввода и оценивать производительность системы и выявлять узкие места. Эти навыки полезны при создании любых поисковых систем, особенно в условиях ограниченных ресурсов или необходимости точного соответствия запроса.

Лабораторная работа №6, 8 «Токенизация» и «Стемминг»

1 Описание

Для подготовки корпуса документов к индексации необходимо произвести токенизацию — процесс разбиения текста на отдельные единицы (токены), которые будут использоваться при построении индекса. В рамках данной работы были разработаны правила токенизации, позволяющие эффективно обрабатывать естественный язык.

Правила токенизации

Текст документа разбивается на токены по следующим правилам:

- Удаляются все знаки препинания и специальные символы;
- Разделителем служит пробел или последовательность пробельных символов;
- Все слова приводятся к нижнему регистру;
- Числа и цифры сохраняются как отдельные токены;
- Учитываются только алфавитно-цифровые последовательности.

После токенизации применяется **стемминг** — процесс приведения слов к их основной форме (stem). Используется простой русскоязычный стеммер, который удаляет окончания, чтобы формы одного слова («книга», «книги», «книгой») объединялись в один токен — `книг`.
Это позволяет:

- Уменьшить размер индекса за счёт сокращения количества уникальных токенов;
- Обеспечить релевантность поиска при запросах в разных формах слов;
- Улучшить точность поиска без потери семантики.

Однако это может привести к потере контекста: не учитывается многозначность слов (например, «банк» — финансовая организация или берег реки).

2 Статистические данные

По результатам обработки всего корпуса (31 683 документа) получены следующие показатели:

- Количество токенов в среднем на документ: 351;
- Средняя длина токена: 11 символов.

В курсе OTEA средняя длина токена была вычислена по сырому тексту без удаления знаков препинания и без стемминга. Здесь же мы:

- Удаляем пунктуацию — это уменьшает длину токенов;
- Применяем стемминг — это ещё больше сокращает длину (обрезаем окончания);
- Фильтруем короткие слова (например, предлоги, союзы) — это может увеличивать среднюю длину, если оставляем только значимые слова.

Таким образом, различия в средней длине обусловлены разными целями обработки: в OTEA — анализ языка, здесь — оптимизация поиска.

Обработка всего корпуса заняла **101,256 секунды**. Это соответствует скорости.

3 Выводы

Выполнив данную лабораторную работу я научилась:

- Разрабатывать и описывать правила токенизации текста, учитывающие особенности естественного языка;
- Применять стемминг для нормализации слов и повышения качества поиска.

Эти навыки являются фундаментальными для построения любых поисковых систем — от простых булевых до сложных ранжирующих. Я также осознала важность выбора метода обработки текста: даже небольшие изменения в правилах токенизации могут существенно повлиять на качество поиска и размер индекса.

Лабораторная работа №7 «Закон Ципфа»

1 Описание

Закон Ципфа — это эмпирическое правило, утверждающее, что в естественном языке частота встречаемости слова обратно пропорциональна его рангу при убывании частоты:

$$f(r) \approx \frac{C}{r},$$

где $f(r)$ — частота слова с рангом r, а C — константа, зависящая от корпуса.

Цель данной лабораторной работы — проанализировать распределение частот токенов в подготовленном корпусе документов (31 683 книги с сайта `litres.ru`) и проверить, насколько оно соответствует закону Ципфа. Для этого был построен график в двойном логарифмическом масштабе, где по оси абсцисс отложен ранг токена (r), а по оси ординат — его частота (f).

2 Результаты и анализ графика

На рисунке 3 представлен график распределения частоты токенов по их рангу в логарифмическом масштабе.

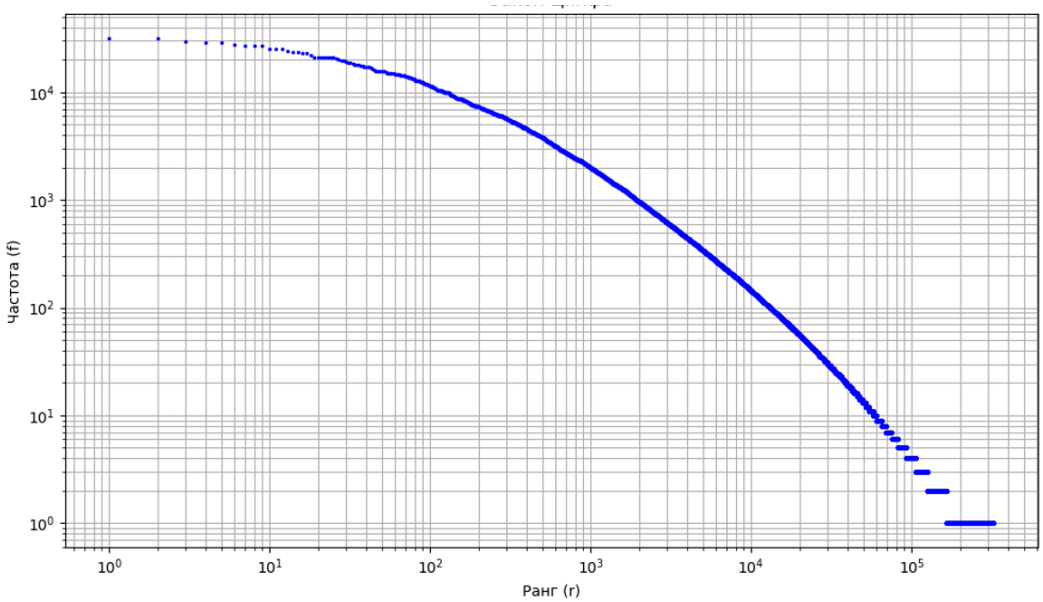

Рисунок 3 - Закон Ципфа

Характеристики графика

- **Начальный участок (ранги 10^0–10^2)** — наблюдается «плато»: несколько самых частых токенов имеют почти одинаковую частоту (около 10^4). Это связано с тем, что в корпусе доминируют функциональные слова («и», «в», «не», «на», «что»), которые встречаются крайне часто и практически не различаются по частоте. Такое поведение типично для больших корпусов естественного языка.

- **Средний участок (ранги 10^2–10^5)** — наблюдается плавное снижение частоты по степенному закону. На этом участке график близок к прямой линии в логарифмическом масштабе, что подтверждает выполнение закона Ципфа: $f(r) \propto r^{-1}$.

- **Конечный участок (ранги $> 10^5$)** — частота токенов стабилизируется на уровне 10^0 (то есть 1 раз за весь корпус). Это объясняется тем, что большинство слов в корпусе встречаются только один раз — они являются редкими или уникальными (например, имена собственные, термины, опечатки).

- **Разрывы и «ступеньки»** — на графике видны горизонтальные отрезки («ступеньки») в области низких частот. Это связано с дискретностью данных: при большом количестве токенов с одинаковой частотой (например, 2, 3, 5) они группируются на одном уровне, создавая визуальные «ступени».

3 Выводы

Выполнив лабораторную работу по анализу распределения частот токенов по закону Ципфа, я убедилась, что в подготовленном корпусе (31 683 документа) действительно наблюдается степенная зависимость частоты от ранга — это подтверждает, что корпус соответствует законам естественного языка, оценила практическую значимость результатов: знание распределения позволяет оптимизировать индексацию (например, игнорировать редкие токены), выбирать методы ранжирования (TF-IDF) и строить эффективные схемы сжатия.

Эти навыки важны для понимания структуры текстовых данных и проектирования поисковых систем, поскольку закон Ципфа является фундаментальным свойством естественного языка, влияющим на все этапы обработки — от токенизации до ранжирования.

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.))