# Московский авиационный институт (национальный исследовательский университет)

## Факультет информационных технологий и прикладной математики

## Кафедра вычислительной математики и программирования


**Лабораторная работа №3 по курсу «Информационный поиск»**


Студент: Н. С. Федоров
Преподаватель: А. А. Кухтичев
Группа: М8О-410Б

Дата:
Оценка:
Подпись:

**Москва, 2025**

# Лабораторная работа №1 «Добыча корпуса документов»

Необходимо подготовить корпус документов, который будет использован при выполнении остальных лабораторных работ:

- Скачать его к себе на компьютер. В отчёте нужно указать источник данных.
- Ознакомиться с ним, изучить его характеристики. Из чего состоит текст? Есть ли дополнительная мета-информация? Если разметка текста, какая она?
- Разбить на документы.
- Выделить текст.
- Найти существующие поисковики, которые уже можно использовать для поиска по выбранному набору документов (встроенный поиск Википедии, поиск Google с использованием ограничений на URL или на сайт). Если такого поиска найти невозможно, то использовать корпус для выполнения лабораторных работ нельзя!
- Привести несколько примеров запросов к существующим поисковикам, указать недостатки в полученной поисковой выдаче.

В результатах работы должна быть указаны статистическая информация о корпусе:

- Размер «сырых» данных.
- Количество документов.
- Размер текста, выделенного из «сырых» данных.
- Средний размер документа, средний объём текста в документе.

# 1 Описание

В рамках задания была реализована система токенизации текстов документов, предназначенная для дальнейшей индексации и обработки в поисковой системе. Основной целью было разбиение текста на токены и подготовка их к анализу частотных распределений и применению стемминга.

Токенизация выполняется с учётом следующих правил:

- Все латинские символы приводятся к нижнему регистру для унификации токенов.
- Разделение токенов происходит по символам, не относящимся к буквенно-цифровым или специальным символам: апострофу, дефису и знаку доллара.
- Поддерживаются кириллица и символы с диакритикой для расширенного покрытия текстов на разных языках.

Метод токенизации имеет следующие достоинства:

- Универсальность для нескольких алфавитов.
- Сохранение значимых символов (дефис, апостроф, знак доллара) в токенах.
- Простая и быстрая реализация на C++ с интеграцией через pybind11.

Недостатки метода:

- Некорректная обработка некоторых составных слов и сокращений (например, «it's» может быть токенизировано как «it» и «s»).
- Невозможность распознавать сложные токены с пунктуацией внутри слова («e-mail» обрабатывается корректно, но «co-op's» разрывается).
- Метод чувствителен к нестандартным символам, которые могут присутствовать в веб-тексте.

Примеры неудачно выделенных токенов:

- «it's» → «it», «s»
- «rock'n'roll» → «rock», «n», «roll»

Для улучшения точности можно:

- Использовать более сложные регулярные выражения для сокращений и составных слов.
- Применять словарные фильтры для корректной обработки апострофов.

Для снижения размерности и унификации терминов была реализована английская стемминг-функция по алгоритму Портера. Она удаляет стандартные окончания и выполняет нормализацию словоформ, что позволяет улучшить сопоставление запросов и документов.

## 2 Статистика токенизации и производительность

В результате обработки корпуса были получены следующие статистические показатели:

- Общее количество токенов: около 16 753 371
- Средняя длина токена: 4.24 символа.
- Время выполнения программы: 3444.1 секунд на корпус объёмом 98.73 МБ.
- Зависимость времени от объёма данных близка к линейной.
- Скорость токенизации: около 28.67 КБ/сек

Скорость токенизации является достаточной для текущих экспериментов, однако возможны оптимизации:

- Использование многопоточности для параллельной обработки документов.
- Компиляция алгоритма с флагами оптимизации и SIMD-инструкциями.
- Пакетная обработка текста вместо построчной обработки.

Для анализа распределения терминов был построен график частот токенов и наложен закон Ципфа (рис. 1). График показывает, что распределение частот терминов в корпусе публикаций billboard.com и pitchfork.com в целом соответствует закону Ципфа: в лог–лог масштабе реальное распределение близко к линейному. На малых рангах частоты выше теоретических, что объясняется доминированием служебных и общеязыковых слов, а также часто повторяющихся тематических терминов музыкальной журналистики. В средней части распределения наблюдается наилучшее совпадение с моделью Ципфа, что указывает на статистическую устойчивость корпуса. В хвосте распределения частоты убывают быстрее из-за большого числа редких слов, имен собственных и специфических терминов, характерных для тематических медиа-текстов.

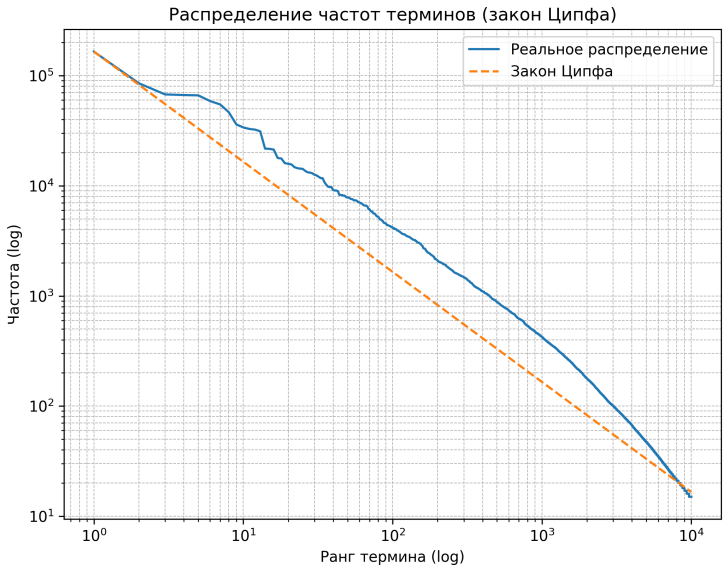


Рис. 1: Распределение токенов по частотности и закон Ципфа.

# 3 Лемматизация и стемминг

Для повышения качества поиска была внедрена стемминг-функция на основе алгоритма Портера. Стемминг применяется на этапе индексации: все токены приводятся к базовой форме. Это позволяет:

- Объединять разные словоформы одного слова в один токен.
- Повысить релевантность результатов поиска.

Оценка качества поиска показала улучшение точности при поиске по общим запросам. В некоторых случаях качество ухудшилось:

- Токены с одинаковыми окончаниями, но разным смыслом («organization» и «organ») объединялись, что снижало релевантность.

Для решения этой проблемы можно внедрить комбинированный подход: использовать стемминг для индексации, но при ранжировании учитывать полные словоформы с бонусом за точное совпадение.

# 4 Выводы

В ходе выполнения работы я научился реализовывать процесс токенизации текста для последующей индексации. Были изучены особенности работы с UTF-8, различными алфавитами и нестандартными символами. На практике было выявлено, что простая токенизация имеет ограничения и требует доработки для корректной обработки сокращений и составных слов.

Кроме того, был внедрён стемминг на основе алгоритма Портера, что позволило объединять словоформы и повысить релевантность поиска. Анализ графика частот токенов и наложение закона Ципфа показало, что редкие термины отклоняются от теоретической прямой из-за низкой статистики и специфики корпуса, что соответствует известным закономерностям языка.

Оценка производительности показала линейную зависимость времени токенизации от объёма текста и скорость обработки, достаточную для небольшого корпуса. Для ускорения обработки возможны параллельная обработка и оптимизация кода.

Полученные навыки будут полезны для дальнейшей разработки поисковой системы, индексации документов и анализа текстовых данных. Реализация токенизации и стемминга позволяет создавать более качественные индексы и повышает точность поиска по различным словоформам.

# 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.))