БУ ВО «Сургутский государственный университет»

Политехнический институт

Кафедра автоматизированных систем обработки информации и управления

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

ПО ТЕМЕ «Создание мини-поисковой системы»

ПО ДИСЦИПЛИНЕ «Поисковые алгоритмы в информационном пространстве»

Выполнил: студент группы №606-12,

Демьянцев Виталий Владиславович

Принял: ст. преподаватель,

Гавриленко Анна Владимировна

Сургут 2024

# Введение

Цель проекта — создать поисковую систему для поиска информации о музыкальных жанрах, таких как рок, метал, блюз и их поджанры. Система собирает данные из Википедии, обрабатывает их и предоставляет пользователю релевантные результаты с удобным интерфейсом. Основная задача — сделать поиск быстрым, точным и понятным, чтобы пользователь мог легко найти статьи о жанрах, например, «блэк-метал» или «рок-музыка», и получить краткое описание.

Проект включает сбор данных (краулер), их обработку (индексатор), ранжирование результатов (алгоритмы BM25 и PageRank) и красивый интерфейс с тёмной темой. В ходе работы были решены проблемы, такие как ошибки в обработке запросов и ограниченный вывод результатов, чтобы система находила все поджанры, например, для запроса «метал».

# Описание системы

## Сбор данных

Для сбора данных используется программа на Python (краулер), которая скачивает статьи с Википедии, начиная со страницы «Список музыкальных жанров, направлений и стилей». Краулер:

* Переходит по ссылкам, собирая только статьи, связанные с музыкой.
* Сохраняет заголовок, URL, текст и ссылки каждой статьи в базе данных.
* Использует приоритет по «глубине» (близости к начальной странице), чтобы фокусироваться на релевантных статьях.

## Обработка данных

После сбора данные обрабатываются индексатором, который:

* Разбивает текст статей на слова.
* Считает, сколько раз каждое слово встречается в каждой статье.
* Сохраняет эту информацию в базе данных для быстрого поиска.

Это помогает системе понимать, какие статьи связаны с запросом.

## Алгоритмы поиска

Система использует два алгоритма для ранжирования результатов:

* **BM25**: Оценивает, насколько статья соответствует запросу. Например, если вы ищете «рок музыка», BM25 смотрит, как часто слова «рок» и «музыка» встречаются в статье, и насколько они редкие в базе данных. Статьи с большим числом совпадений получают высокий рейтинг.
* **PageRank**: Проверяет «важность» статьи на основе ссылок. Если на статью «рок-музыка» ссылаются другие статьи, она считается более значимой.

Результаты сортируются по комбинированному рейтингу: BM25.

## Архитектура

Система состоит из трёх частей:

* **Бэкенд**: Обрабатывает запросы пользователей, ищет статьи в базе данных, применяет BM25 и PageRank, возвращает результаты. Например.
* **Фронтенд**: Показывает интерфейс, где пользователь вводит запрос, видит результаты и подсказки. Интерфейс имеет тёмную тему и удобный дизайн.
* **База данных**: Хранит статьи, слова и их частоту. Это позволяет быстро находить нужные данные.

# Пользовательский интерфейс

Интерфейс создан с тёмной темой, чтобы выглядеть стильно и быть удобным для глаз. Основные элементы:

* Поле для ввода запроса с подсказками (например, при вводе «рок» предлагаются «рок-н-ролл», «рок-музыка»).
* Кнопка «Искать» для запуска поиска.
* Список результатов в виде карточек, где каждая карточка показывает:
  + Заголовок статьи (например, «Блэк-метал»).
  + Короткий отрывок (сниппет) с выделенными словами запроса.
  + Рейтинги (общий, BM25, PageRank) для прозрачности.
* Кнопки «Назад» и «Вперёд» для переключения страниц результатов.

  
