Инструмент для поиска плагиата в исходном коде

Куратор	Дмитрий Иванов, 6304
Лидер	Корытов Павел, 6304
Разработчики	Артём Бутко, 8304 Дмитрий Перелыгин, 8303 Александр Алтухов, 8304 Александр Рыжиков, 8304

Постановка задачи

✔️	Тестирование методов нечеткого поиска по исходному коду
✔️	Реализация поиска по StackExchange/StackOverflow
❌	Реализация поиска с поисковыми системами (Google)
✔️	Создание БД, реализация управления БД через REST API
❌✔️	Реализация работы с репозиториями GitHub
❌	Подключение фронтэнда к REST API

Методы решения

- Язык программирования - Python
  - BeautifulSoup4 для парсинга датасета StackOverflow
  - Flask - веб-сервер
- PostgreSQL + fuzzystrmatch для поиска похожего кода в БД
  - Конструктор запросов к SQL - PyPika
- Vue.js + Bootstrap - фронтэнд

Результаты. Нечеткий поиск

Попробовали варианты:

- ElasticSearch
- PostgreSQL + pg_trgm
- PostgreSQL + fuzzystrmatch - остановились здесь

Алгоритм предполагает построчное разбиение исследуемого кода, преобразование полученных строк с помощью функции metaphone, предоставляемой модулем fuzzystrmatch, нахождение расстояния Левенштейна между закодированными строками. Данный подход позволит учесть релевантность возможных совпадений.

Пример работы алгоритма

Сравниваются следующие строки:

function drawMatrix(matrix, offset)

function drawMtx(mtx, offset)

Data Output | Explain | Messages | Notifications

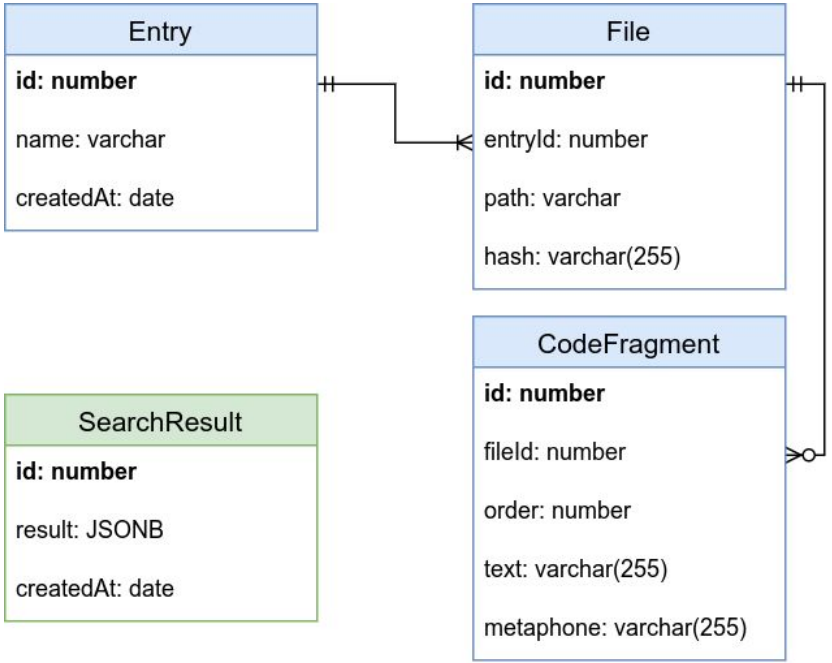
	metaphone text	metaphone text	levenshtein integer
1	FNKXNTRMTRKSMTRKSFST	FNKXNTRMTKSMTKSFST	2

Расстояние Левенштейна между кодами этих строк равно 2, что позволяет считать их достаточно похожими. Максимальное допустимое расстояние для принятия решения о плагиате должно зависеть от длин строк.

Поиск по открытым источникам

- Поиск исходного кода через Google осуществить не удалось
- API GitHub и StackExchange использовать не удалось из-за квоты на запросы и максимального размера запроса
- Результат - реализована выгрузка и разбор датасета StackOverflow
  - Из тестового xml файла, содержащего информацию из 3000 постов со Stackoverflow было найдено:
  - 38 строчек кода на Python
  - 92 строчек кода на Java и JS
  - 12 строчек кода на C

БД и REST API



- Реализованы CRUD-запросы в REST API к БД, в т.ч. загрузка файлов

Работа с репозиториями

- Реализовано:
  - Выгрузка одиночного репозитория
  - Выборка из репозитория исходного кода
- Не реализовано:
  - Работа с репозиториями организации на GitHub

Инструкция по запуску

- Парсинг датасета StackOverflow
  - Установить зависимости Python (pip install -r requirements.txt)
  - После запуска программы(python pars.py) ввести путь до файла, который необходимо парсить
  - В консоль будут выведены теги языков из обработанного кода
  - В файлы с соответствующими названиями будут добавлены фрагменты кода разделенные по языкам
  - В файл o.txt будут сохранены все фрагменты кода
- Выгрузка репозитория с GitHub
  - python delete_files.py <URI репозитория> <путь>
  - В папке <путь> будут полученные файлы с исходным кодом
- Разворачивание REST API
  - Установить зависимости Python (pip install -r requirements.txt)
  - Воспользоваться командой export FLASK_APP=app.py (Linux) или
  - set FLASK_APP=app.py (Windows)
  - Запустить сервер командой python -m flask run

Планы на следующую итерацию

- Подключение выгрузки репозитория к REST API
- Реализация работы с репозиториями организаций на GitHub
- Реализация алгоритма поиска плагиата
- Реализация загрузки датасета StackOverflow в БД
- Подключение фронтэнда к REST API
- Модульное тестирование