Техническое задание для хакатона: "DocFlow Content Analyzer — поиск и классификация документов по содержимому"

Цель

Разработать приложение для автоматической обработки PDF-документов, их категоризации и группировки на основе анализа содержимого.

Основные функции

1. Загрузка и обработка документов

- Загрузка PDF-документов (поддержка многостраничных файлов).
- Извлечение текста из PDF (включая обработку сканированных документов через OCR, например, Tesseract).
- Сохранение структуры документа (заголовки, разделы, списки).

2. Автоматическая категоризация

- Классификация документов по темам на основе ключевых слов (например: "Юридические", "Технические", "Финансовые").
- Выявление подгрупп внутри категорий (например, разделение "Юридических" документов на "Договоры", "Исковые заявления").

3. Поиск по содержимому

- Полнотекстовый поиск по документам с подсветкой результатов.
- Фильтрация по категориям, подгруппам и датам.

4. Разделение документов на подгруппы

- Автоматическое разбиение многостраничных PDF на логические части (например, главы, разделы).
- Возможность ручной коррекции группировки.

Требования к реализации

Серверная часть

- **API для загрузки документов**:
  - Прием PDF.
  - Парсинг текста и метаданных (автор, дата создания).
- **Хранение данных**:
  - Текст документов, категории, подгруппы.
  - Поддержка быстрого поиска (индексация).
- **Интеграция алгоритмов анализа**:
  - NLP-модели для классификации (например, TF-IDF, Word2Vec).
  - Кластеризация документов (например, методом k-mean).

Клиентская часть

- **Интерфейс загрузки**:

    - Drag-and-drop для PDF.
    - Отображение прогресса обработки.
- **Визуализация результатов**:
    - Дерево категорий и подгрупп.
    - Просмотр документов с навигацией по разделам.
- **Поиск**:
    - Строка поиска с автодополнением.
    - Фильтры по тегам и датам.

Дополнительные функции (по желанию)

- Ручная корректировка групп: Перетаскивание страниц между подгруппами.
- Суммаризация: Автоматическое создание краткого содержания документа.
- Тегирование: Присвоение тегов на основе ключевых сущностей (имена, даты, организации).
- Экспорт: Выгрузка подгрупп в отдельные PDF-файлы.

Пример сценария использования

Пользователь загружает PDF-документ "Техническое задание проекта.pdf".
Система извлекает текст, определяет категорию "Технические документы" и разбивает файл на подгруппы:

- "Требования" (страницы 1-3).
- "Архитектура" (страницы 4-5).
- "Сроки" (страница 6).

При поиске по слову "архитектура" пользователь видит все документы и разделы, содержащие этот термин.

Критерии оценки

- **Рабочий прототип (50%)**:
    - Загрузка PDF, извлечение текста, категоризация.
    - Базовый поиск по содержимому.
- **Точность анализа (25%)**:
    - Качество автоматической группировки.
    - Релевантность результатов поиска.
- **Интерфейс (15%)**:
    - Удобство навигации по подгруппам.
    - Интуитивность поиска.
- **Документация (10%)**:
    - Описание алгоритмов классификации и примеры запросов.

Рекомендации

- Для обработки PDF: библиотеки вроде PyPDF2 (Python), Apache PDFBox (Java), iText (C#/Java).
- Для NLP: spaCy, Hugging Face Transformers.
- Для фронтенда: фреймворки с поддержкой древовидных структур (например, React + D3.js).