**Работа IR- системы на основе кастомного парсера, chroma-db и GigaChat.**

Общий обзор работы кода

1. Загрузка документа формата DOCX:

- Загружается документ формата .docx.

- Преобразуется в текстовый формат .txt, данные таблиц форматируются в теги, понятные llm.

2. Извлечение параграфов и таблиц:

- Текст из параграфов и таблиц извлекается и сохраняется в текстовый файл .txt.

3. Обработка текста:

- Текст из параграфов и таблиц разбивается на документы(чанки) с использованием RecursiveCharacterTextSplitter.

- Текст загружается в векторное хранилище Chroma для последующего семантического поиска.

4. Создание RetrievalQA цепочки:

- Создается цепочка RetrievalQA на базе LangChain, которая по запросам пользователя ищет релевантные куски текста в документе и формирует ответ, используя заданную языковую модель GigaChat.

5. Логика для работы с таблицами:

- Код содержит логику понимания тегов для формата таблиц в текстовом виде.

**Плюсы и минусы такого подхода**

Плюсы

1. Универсальность:

- Подходит для обработки различных документов формата .docx.

2. Интеграция с системой векторизации:

- Используется векторное хранилище Chroma (созданное специально для семантического поиска).

3. Возможность прямого доступа к ячейкам:

- Таблицы сохраняются в формате тегов, что позволяет обращаться к конкретным ячейкам.

Минусы

1. Стоимость токенов для GigaChat-Max

-Для лучшей производительности, скорости и понимания контекста требуется самая дорогая модель среди GigaChat, однако эта стоимость вполне оправдана.

2. Нет обработки многоуровневых заголовков:

- Код не поддерживает обработку многоуровневых заголовков.