Документация

Цель:

Сделать сервис для сравнения работы различных LLM моделей, а так же различных методов улучшения контекстов (прямой ввод данных, RAG, GraphRAG)

Структура проекта:

- Папка RAG содержит скрипты работы с LLM моделью. Файл classic_rag.py содержит класс, позволяющий конструировать запросы к LLM на базе классического RAG
- Папка API содержит back-end часть проекта, написанная с помощью библиотеки FastAPI

Для пользователя доступен следующий функционал:

- загрузка своих данных
- выбор параметров построения системы RAG
- выбор LLM модели
- возможность задать вопрос по введённым текстам
- возможность переключаться между построенными базами данных и LLM
- простая аналитика по текстам

Сервер выполняет следующий функции:

POST

- формирует базу данных на основе заданных пользователем текстов и параметров
- отправляет запрос к выбранной LLM с использованием введеных пользователем параметров ретривера и модели

GET

- получение списка постронных RAG с метаинформацией

DELETE

- возможность удалять ненужные базы знаний
- возможность удалять загруженные файлы
- возможность удалять ненужные конфиги LLM

Работа сервиса:

- Начальная страница

- Страница с загрузкой данных.

Тексты подаются в формате .txt и сразу строится база знаний

- Создание модели.

Выбор гиперпараметров для модели

- Выбор модели.

Позволяет пользователю переключаться между созданными моделями и текстами для запросов

- Аналитика по тексту
- Вопрос пользователя к модели
- Удаление неиспользующихся моделей

Чтобы запустить сервис нужно запустить скрипт из папки checkpoint4 командой:

python3 stapp_main.py

Чтобы запустить докер контейнер, нужно в папке checkpoint4 с установленным docker запустить команды:

docker-compose build

docker-compose up