Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления

Кафедра Интеллектуальных информационных технологий

**ОТЧЁТ**

**По лабораторной работе № 6**

по дисциплине «Естественно-языковой интерфейс интеллектуальных систем»

по теме: “Естественно-языковой интерфейс диалоговой системы с пользователем”

Выполнил: Веркович Е. В.

Лукашевич К. С.

гр. 221703

Проверил: Крапивин Ю. Б.

Минск 2025

**Цель работы:**

освоить принципы разработки диалоговых систем с поддержкой естественного языка.

**Задание:**

-изучить основы создания диалоговых систем с поддержкой естественного языка;

­ закрепить навыки программирования при решении задач организации

диалогового взаимодействия с поддержкой естественного языка.

Требуется спроектировать и программно реализовать структуры хранения данных, алгоритмы их обработки, необходимые в рамках следующих базовых требований к разрабатываемому приложению:

­ входные данные – текстовое сообщение на заданном естественном

языке;

­ выходные данные – автоматическая реакция системы на входное

сообщение на естественном языке путем формирования ответного сообщения

согласно варианту задания;

­ взаимодействие с пользователем посредством графического

интерфейса (интерфейс должен быть интуитивно понятным и дружественным

пользователю);

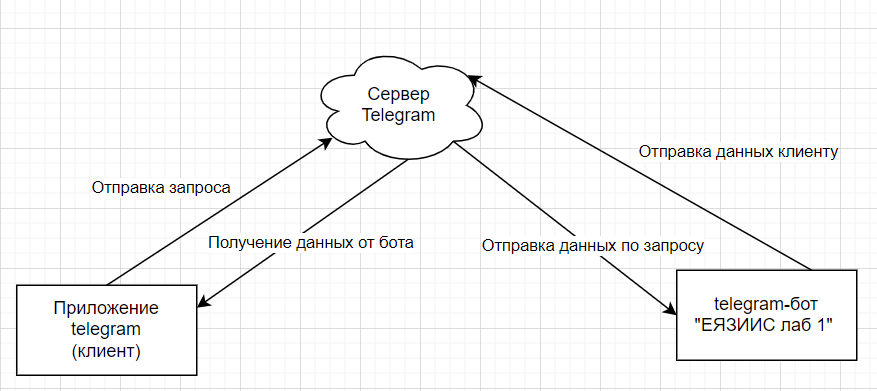
­ наличие системы средств помощи пользователю;

­ обеспечение возможности ведения диалога с пользователем на

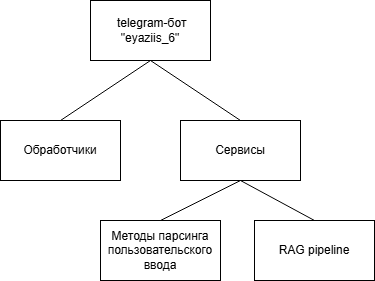
естественном языке, сохранения, просмотра, редактирования истории диалога

либо заданной его части.

**Структурно функциональная схема разработанного приложения**



Пользователь загружает word-документы (.docx файл) с помощью приложения telegram, отправляет команды в чат с ботом, а также ведет диалог текстовыми сообщениями. Запросы отправляются на сервер telegram, который отправляет обновления о состоянии чата. Бот соответственно получает апдейты(обновления) посредством поллинга(постоянного опроса сервера на наличие обновлений, адресованных боту).



Само приложение(бот) состоит из двух основных модулей - обработчиков и сервисов.

Обработчики или хендлеры реализуют обработку запросов от пользователя: реагируют на команды и различный пользовательский ввод.

К сервисам относятся методы парсинга пользовательского ввода, то есть вспомогательные функции для обработчиков, а также основная логика, которая включает RAG pipeline для ответов на вопросы пользователя и ведения диалога.

**Структуры хранения данных.**

В состоянии FSMContext хранится список документов documents, каждый элемент которого содержит:

text - извлечённый текст из DOCX или пользовательский текст.

metadata - метаданные, например, имя файла или источник текста.

documents = [

{

"text": "Текст из документа или пользовательский ввод",

"metadata": {

"file\_name": "example.docx" # или "source": "user\_text"

}

},

...

]

**Алгоритмы обработки**

***Обработка документа (DOCX)***

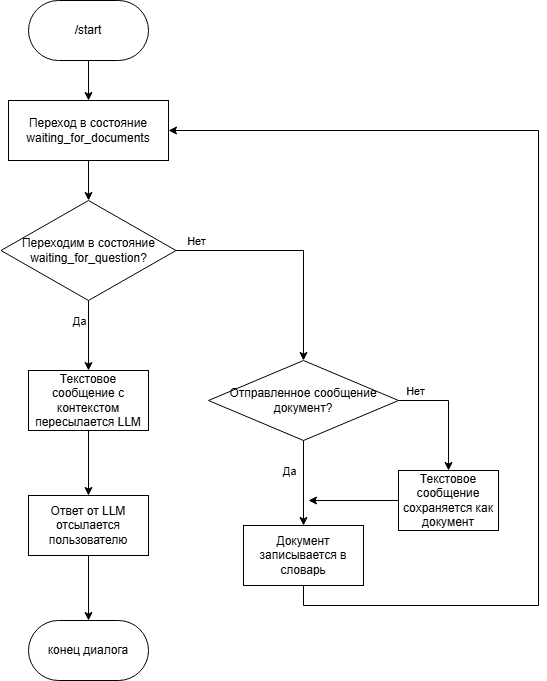
1. При получении сообщения с документом проверяется расширение файла (только .docx).
2. Файл скачивается через Telegram API.
3. С помощью библиотеки python-docx извлекается текст из параграфов.
4. Текст и метаданные добавляются в состояние FSM.
5. Пользователю отправляется подтверждение успешной загрузки или сообщение об ошибке.

***Обработка текстового сообщения***

1. Текст добавляется в список документов в состоянии FSM с метаданными о том, что это пользовательский ввод.

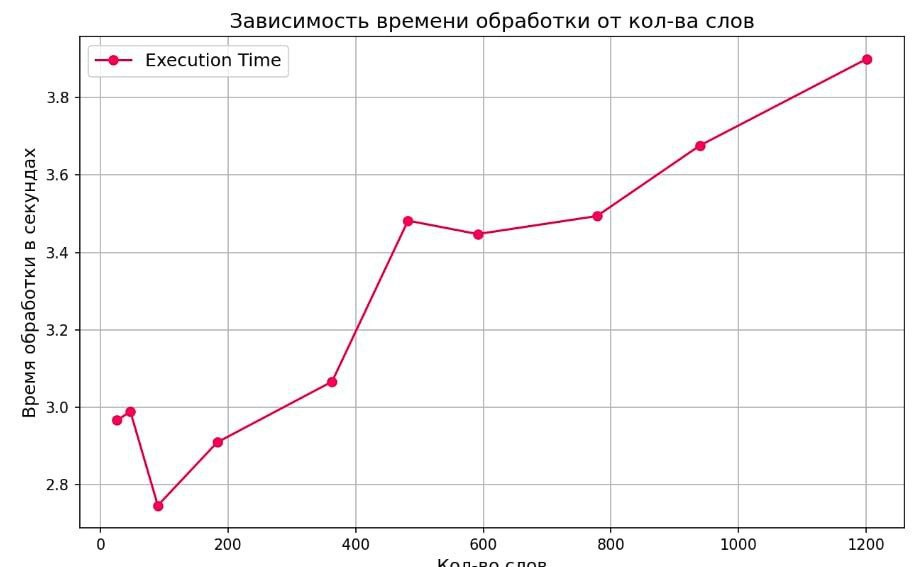
***Обработка вопроса***

1. Извлекаются все документы из состояния.
2. Документы конвертируются в объекты Document LangChain.
3. Создаются эмбеддинги с помощью модели HuggingFace sentence-transformers/all-MiniLM-L6-v2.
4. Создаётся векторное хранилище FAISS для поиска по документам.
5. Инициализируется LLM через OpenRouter с кастомным промптом, ограничивающим ответы кулинарной тематикой.
6. Запускается цепочка RetrievalQA, которая ищет релевантные документы и формирует ответ. LLM модель для генерации ответа и ключ инициализируется в настройках. Используется бесплатная модель с Open Router qwen/qwen3-4b:free.
7. Ответ и источники отправляются пользователю.

****

**График производительности приложения**

Сначала идет просадка по скорости из за времени загрузки модели для эмбеддингов с жесткого диска в оперативную память. Время задержки ответа от выбранной llm qwen/qwen3-4b:free меньше секунды, далее время растет от количества переданных токенов.



**Вывод:**

В ходе выполнения лабораторной работы были принципы разработки диалоговых систем с поддержкой естественного языка. Были разработаны: структуры хранения данных, алгоритмы их обработки.

Таким образом, результаты лабораторной работы могут быть использованы для создания эффективных диалоговых систем, способных вести интерактивный обмен информацией с пользователями.