**1 СЛАЙД**

Добрый день, уважаемые председатели и члены комиссии. Вашему вниманию представляется выпускная квалификационная работа на тему "программный комплекс интеллектуального поиска в корпоративных документах".

За последние несколько десятилетий произошел значительный рост объемов информации, который требует тщательного анализа поступающих данных, что существенно замедляет рабочий процесс в корпорациях. В этих условиях традиционные методы работы с информацией становятся неэффективными и нуждаются в значительных ресурсах для поддержания актуальности и доступности информации.

**2 СЛАЙД**

В связи с этим были сформулированы цель и задачи

Но сперва проведем краткий экскурс в предметную область.

**3 СЛАЙД**

Как и было сказано из цели необходимо обрабатывать запрос на естественном языке, на данный момент для этого используются языковые модели, которые были использованы при обработке.

при создании промпта, LLM ищут информацию среди тех данных, которые были предоставлены для обучения, но при поставленной цели необходимо искать среди тех данных, которые были подгружены в модель. Такой метод называется Retrieval Augmented Generation (RAG). Исходный промпт превращается в эммбединг, который ищет релевантные данные в векторной БД, что на конечном шаге вместе с исходным промптом подается языковой модели и она ищет релевантную информацию именно из предоставленной информации

**4 СЛАЙД**

Для того, чтобы языковая модель могла понимать естественный язык, т.е. обрабатывать запрос, его необходимо преобразовать в векторное представление. Для этого существует модель эмбеддинга, что позволяет сравнивать вектор запроса с вектором данных через косинусное сходство, если полученное значение достаточно мало, то данная информация является релевантной и, следовательно, будет включена в ответ языковой модели

**5 СЛАЙД**

Переходя к видам языковых моделей, было выделено 2 вида, трансформер и mamba, которые обрабатывают большие последовательности быстро и эффективно, с их описанием можно ознакомиться в пояснительной записке

**6 СЛАЙД**

Далее был проведен обзор аналогов. Следует отметить, что несмотря на наличие обширного списка аналогов их использование невозможно в связи с санкциями российским пользователям, несоответствием требований и необходимостью мощного ПО

**7 СЛАЙД**

Связи с этим были выбраны следующие модели. Llama и GigaChat. GigaChat мощная модель, но к ней можно подключиться и использовать API. Llama имеет несколько моделей, для которых не требуется мощное ПО

**8 СЛАЙД**

В качестве стека технологий был выбран набор: streamlit, ollama, langchain. Вся разработка производилась на языке python.

**9 СЛАЙД**

Проведя анализ, были сформулированы функциональные требования, на основе которых происходили этапы проектирования. На 1м этапе была спроектирована диаграмма прецендентов. Сотруднику необходимо сформировать свой запрос используя дополнительные настройки: выбрать языковую модель, определить область поиска. Так же сотрудник может загружать свои документы в хранилища и при необходимости удалять

**10 СЛАЙД**

На 2м этапе была спроектирована диаграммы последовательности.

На 1м рисунке показано, как сотрудник попадает в систему, происходит получение токена доступа для взаимодействия c API GigaChat и получение документов как из корпоративной базы, так и из личного хранилища сотрудника. После инициализации отображается чат с необходимыми настройками, и сотрудник может вводить свой запрос.

На 2м рисунке показано, как сотрудник работает с документами, ему необходимо их загрузить в хранилище. После он может посмотреть их наличие и удалить при необходимости актуализации документа или устаревания

**11 СЛАЙД**

Завершая этап проектирования, были составлены макеты интерфейса. При входе в комплекс сотрудник попадает на главную страницу, где будет размещена информацию о данном комплексе. На станице Чат сотрудник будет задавать запрос. В разделе Документы сотрудник будет загружать, просматривать и удалять документы

**12 СЛАЙД**

Переходя к этапу реализации, была составлена диаграмма пакетов, которая отображает зависимость пакетов иллюстрации функциональности программного обеспечения программного комплекса

**13 СЛАЙД**

На этапе тестирования были проверены ответы языковых моделей в зоне поиска определенного документа и среди базы документов. В качестве примера был использован документ ОС ТУСУР. На 1м рисунке показано, как данный файл выбирается и в нем ищется ответ. На 2м рисунке, как система по аналогичному запросу находит данный файл, а потом получает из него ответ.

**14 СЛАЙД**

llama3.1:8b

**15 СЛАЙД**

GigaChat

**16 СЛАЙД**

Всего было сделано 30 запросов к моделям. Корректность ответа определялась вручную

**17 СЛАЙД**

В заключении, был разработан программный комплекс интеллектуального поиска в корпоративных документах. За период обучения была опубликована 1 статья, которая написана по теме ВКР.

Продолжение работы планируется в рамках магистерской диссертации.

Спасибо за внимание.

(Векторная база данных - это специализированный тип базы данных, который хранит данные в виде многомерных векторов, каждый из которых представляет определенные характеристики или качества. Основное преимущество векторной базы данных заключается в ее способности эффективно и точно извлекать данные на основе близости или сходства векторов. Это позволяет выполнять поиски на основе семантической и контекстуальной значимости, а не только полагаться на точные совпадения или заранее определенные критерии, как в традиционных базах данных)