0.1 INTRO

Уважаемый председатель государственной экзаменационной комиссии, уважаемые члены государственной экзаменационной комиссии и уважаемые присутствующие! Позвольте представить Вашему вниманию мою выпускную квалификационную работу

0.2 Актуальность

Актуальность моего приложения обусловлена множеством аспектов, но ключевые: возрастающее количество пользователей языковых моделей для бизнесом (мне было интересно собственноручно создать такой проект), в свою очередь актуальность для ВУЗа — создание универсального ассистента, способного отвечать на заданые вопросы без хардкодинга и с возможностью неявного поиска.

0.3 Цели и задачи

Передо мной была поставлена следующая цель: спроектировать и разработать приложение для взаимодействия с структурой предприятия посредством какого-либо интерфейса. Был выбран интерфейс в виде телеграмм бота, но мою систему можно без особых сложностей подключить как к ВКонтакте или Яндекс.Алисе. В настоящий момент работает только в Telegram.

Поставленные задачи отображены на слайде.

0.4 Сравнительный анализ

Во время начала разработки моей системы непосредственных конкурентов у моей системы не было, поэтому я сравнивал свою систему с похожими решениями. Недавно начало разрабатываться подобное решение от компании Сбербанк.

0.5 Swot

Для выявления сильных строн и потенциальных угроз был проведен SWOT анализ.

0.6 Временные оценки

На данном сладе расположена оценка временных затрат, а так же затрат на разработку.

0.7 Гант

На данном сладе расположена диаграмма ганта с визуализацией этапов разработки

0.8 Функциональное моделирование

На данном слайде схематично отображены функциональные требования. Таким образом система должна уметь: приветствовать пользователя, уметь объяснить пользователю как управлять системой, а так же предоставлять информацию о учреждении.

0.9 Схема работы приложения

Пользователь отправляет запрос сервису, после чего запрос преобразуется в вектор. Для преобразования вектора испольуется малослойная нейронная сеть. Далее полученный вектор сравнивается при помощи вычисления косинусного и Евклидового растояний. Вектора с наибольшим коэфициентом подобия достаются из базы знаний и подставляются в промт. Далее промт с данными из базы знаний и запросом от пользователя отправляется на вход языковой модели, после чего происходит генерация ответа, основываясь на предоставленном контексте. После окончания генерации ответа происходит отправка ответа пользователю.

0.10 Используемые технологии

В основном использовалось три технологии: Python для разработки сервиса и RAG, PyTorch для глубокого обучения, в виду того, что многие методы удобно написаны на языке C++, который во много раз быстрее Python, а так же Docker для упаковки приложения в контейнер для сборки всех необходимых зависмостей, что позволяет запустить приложение на любом устройстве с любой операционной системой.

0.11 Векторизация

Векторизация происходит при помощи малослойной нейронной сети. На вход подается текстовый корпус. Для оптимизации используется градиентный спуск.

0.12 Сопоставление векторов

Сопоставление вектор происходит при помощи расчета Евклидового расстояния между векторами. На слайде можно наблюдать визуализацию данного метода для K=5. В моей системе используется K=3, чтобы уместиться в контекстное окно языковой модели.

0.13 Расчет экономической эффективности

Для расчета экономической эффективности использовалась предоставленная на слайде формула. Процент окупаемости составил 300 процентов, что я считаю довольно хорошим результатом.

0.14 Реализация ИС

На слайдах можно наблюдать реализацию системы. Работа приложения отвечает описанным функциональным требованиям. Так же планировался блок для сурдо-перевода, но у меня не было такого количества данных для тренировки модели + это очень трудозатратно. Посмотреть работы приложе-

ния можно отсканировав QR-код. Приложение работает на моем персональном компьютере и возможно у меня кончится видеопамять)