Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Кубанский государственный университет»

Факультет компьютерных технологий и прикладной математики

Кафедра прикладной математики

**ОТЧЕТ О ПРОХОЖДЕНИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ (технологическая (проектно-технологическая) практика)**

период с 29.06.2024 г. по 26.07.2024 г.

Айрапетов Тигран Эдуардович

*(Ф.И.О. студента полностью)*

студента 35 группы 3 курса ОФО

02.03.03 Математическое обеспечение и администрирование информационных систем

Руководитель практики от факультета

канд. физ.-мат. наук, доцент Колотий А.Д.

*ученое звание, должность (подпись) (Ф.И.О)*

Оценка по итогам защиты практики: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«26» июля 2024 г.  *(дата)*

Руководитель практики от производства

завед. лаб. робототех. и механотрон. КубГУ Прутский А. С.

*должность (подпись) (Ф.И.О)*

Краснодар 2024 г.

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc172858249)

[1 Схема проекта 4](#_Toc172858250)

[2 Описание хода работы 5](#_Toc172858251)

[2.1 Рассмотрение моделей диалоговой суммаризации 5](#_Toc172858252)

[2.2 Реализация Телеграм бота 5](#_Toc172858253)

[2.3 Генерация отчётов 8](#_Toc172858254)

[Заключение 12](#_Toc172858255)

# ВВЕДЕНИЕ

В условиях современных рабочих процессов, особенно в IT-индустрии, команды часто взаимодействуют через мессенджеры, такие как Telegram. Среди большого потока информации нелегко потеряться или забыть про встречи. Для решения этих проблем можно реализовать помощника, который бы собирал важную информацию в чатах проектов и выдавал бы краткий обзор по текущему прогрессу.

Целью данного проекта является создание Telegram-бота, который способен резюмировать обсуждения в чате проекта. Это позволит участникам проекта быстрее и эффективнее получать основную информацию из большого объема сообщений, что повысит продуктивность работы команды и упростит управление проектом.

Проект также представляет собой исследование эффективности реализации технологий машинного обучения в сфере управления проектами. Реализация проекта даст хороший фундамент для создания и развития более комплексных решений.

# 1 Схема проекта

Архитектура проекта состоит из двух основных компонентов: Telegram-бота и FastAPI сервера, которые взаимодействуют между собой для выполнения задач суммаризации сообщений и генерации отчетов.

* Telegram-бот. Этот компонент отвечает за взаимодействие с пользователями. Он получает сообщения из чата, отправляет их на сервер для суммаризации, а затем возвращает результаты обратно в чат.
* FastAPI сервер. Этот компонент обрабатывает запросы от Telegram-бота, выполняет суммаризацию сообщений с использованием модели из библиотеки transformers от HuggingFace и взаимодействует с генеративной моделью GigaChat через соответствующий API для создания детализированных отчетов.

При реализации модулей были использованы следующие технологии:

Библиотека aiogram – используется для создания Telegram-бота. Aiogram является асинхронной библиотекой Python, которая обеспечивает удобный и гибкий способ взаимодействия с Telegram API. Она позволяет легко обрабатывать входящие сообщения, отправлять ответы и управлять состоянием бота.

FastAPI – современный веб-фреймворк для создания API на Python. Он обеспечивает высокую производительность, легкость использования и автоматическую генерацию документации.

Transformers от HuggingFace – библиотека, предоставляющая доступ к множеству предобученных моделей для обработки естественного языка (NLP). В проекте используется модель диалоговой суммаризации из этой библиотеки, что позволяет эффективно и качественно обрабатывать большие объемы текстовой информации и создавать сжатые резюме.

GigaChat – генеративная модель, используемая для создания детализированных отчетов по запросу. Интеграция с GigaChat позволяет получать более глубокий анализ и структурированные отчеты по проекту.

# 2 Описание хода работы

## 2.1 Рассмотрение моделей диалоговой суммаризации

В процессе выбора модели диалоговой суммаризации была сделана попытка обучить модели BERT и T5, используя открытые датасеты диалоговой суммаризации, однако хороших результатов в краткие сроки не удалось достичь. Так как обучение представляет собой длительный и сложный процесс, был сделан выбор в пользу уже обученной на нескольких датасетах модель d0rj/rut5-base-summ. Модель основана на T5 и дает неплохие результаты, однако иногда допускает орфографические ошибки в словах и упускать детали. На рисунке 1 представлен результат тестирования модели на примере чата проекта.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 1 – Тестирование модели rut5-base-summ

## 2.2 Реализация Телеграм бота

Далее был реализован Телеграм бот, для реализации взаимодействия с моделью.

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 2 – Код обработки команды /summarize

На рисунке 2 можно увидеть код обработки команды /summarize, которая отвечает за суммаризацию последних сообщений в чате. При отправке команды /summarize, бот компонует последние собранные сообщения, и отправляет на сервер для обработки сообщений моделью rut5-base-summ. При успешном выводе, в чате выводится сообщение с полученным резюме по текущему обсуждению. К сообщению также прикрепляются кнопки «Сохранить» и «Отмена» для ситуаций, когда в сообщениях была ложная информация или модель плохо справилась с суммаризацией. При нажатии на кнопку «Сохранить» текст суммаризации сохраняется для дальнейших взаимодействий, а текущий набор сообщений стирается из хранилища бота. В дальнейшем можно будет создать более продвинутый набор команд для работы с ботом (сохранение отдельных сообщений без изменений или изменение существующих суммаризаций).

Для просмотра текущего набора сообщений в боте реализована команда /mes, а для просмотра списка сгенерированных суммаризаций – команда /sums. Во время работы приложения данные хранятся в виде словарей – для каждого чата свой набор значений, а при генерации резюме по обсуждению, происходит сохранение данных в json-файл для предотвращения потери данных при отказе работы частей системы. Для еще большей надежности и удобства можно будет использовать БД, подходящие для работы с сообщениями, а также вести сохранение через промежутки времени. На рисунке 3 представлен код обработчиков команд вывода хранимых данных.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – Код обработчиков команд /mes и /sum, а также функция сохранения данных

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4 – Пример работы бота в Телеграмме

Как можно заметить на рисунке 4, бот довольно быстро может обобщить несколько сообщений, а также выделить суть, исключив поздравление с днем рождения. Однако на данном этапе модель все ещё может давать плохие результаты при несвязной речи в чате, а также при больших обсуждениях есть риск потерять слишком много полезной информации. Для решения проблемы с потерей информации можно поэкспериментировать с длинной окна вывода модели и выбрать оптимальное значение.

## 2.3 Генерация отчётов

После того, как в чате набралось достаточное количество обсуждений и суммаризаций, участники могут захотеть получить отчёт по проекту. Подобный функционал был реализован на рисунке 5.

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 5 – Код обработки команды /report

При отправке команды /report, бот отправит сообщение для уточнения, стоит ли учитывать последние сообщения в обсуждении или использовать только сгенерированные суммаризации для генерации отчёта. После получения ответа бот просто компонует данные и посылает на сервер. Далее при успешной генерации отчёта он выводится в чат группы.

Отчёт может включать в себя самую разную информацию, так что его формат может варьироваться от проекта к проекту. Гибким и удобным решением в данном случае будет использование генеративных моделей.

Тем не менее, такие модели являются довольно большими и требуют больших вычислительных мощностей для их работы, так что для подключения к проекту генеративных моделей, необходимо воспользоваться онлайн-сервисом. Одним из таких является GigaChat – разработанный Сбером сервис, способный решать разные задачи – от генерации текста, кода или изображений, до общения с пользователем в режиме диалога.

Для реализации взаимодействия с API GigaChat нужно зарегистрироваться с помощью Сбер ID, а также использовать библиотеку gigachat для python.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 – Код обработки запросов на стороне сервера FastAPI

Для успешного применения модели GigaChat необходимо указать специальный препромпт перед данными, отправляемыми ботом. Этот промпт содержит подробное предписание задачи, которую предстоит решить генеративной модели. На данном этапе препромпт выглядит так: «Необходимо создать отчёт по проекту, используя данные из нижеперечисленных текстов. В отчёте нужно указать участников проекта, планы по проекту(если они есть). Если нужной информации нет, то не придумывай что-то новое. Вот тексты:». При обработке запроса данные чатов вставляются сразу после этого препромпта, после чего модель может сгенерировать отчёт некоторого формата по своему усмотрению.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 7 – Пример работы генерации отчета

Как можно заметить на рисунке 7, исходя из одних и тех же данных можно получить разный результат отчёта. Улучшение качества отчёта напрямую зависит от детализации препромпта, а также правильной работы с ботом посредством периодической суммаризации коротких обсуждений, чтобы фиксировать ключевые моменты в проекте.

# ЗАКЛЮЧЕНИЕ

Проект создания Telegram-бота для суммаризации сообщений в чате был успешным и продемонстрировал значительные улучшения в управлении информацией и повышении продуктивности команды.

Несмотря на то, что на данном этапе, бот уже может работать с чатами и группами, функционал его весьма ограничен, так как помимо текстовых сообщений существуют другие виды информации (изображения, видео, голосовые сообщения, голосования и т.д.). Для более комплексной суммаризации обсуждений по проекту и более комфортной работы для участников проекта, потребуется реализовать много дополнительного функционала, а также улучшить качество работы моделей.

Также для успешного развертывания проекта, потребуется выполнить ещё много шагов, таких как: оптимизация контейнеров, масштабирование и безопасность.

Во время прохождения практики были изучены основные аспекты работы с моделями машинного обучения в продакшене, а также намечены планы развития проекта. Поставленные задачи послужат фундаментом для дальнейшего развития.