

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»
(Университет ИТМО)

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и облачные технологии**

Направление подготовки **09.03.03 Прикладная информатика**

О Т Ч Е Т

по курсовой работе

на тему: “Разработка телеграм бота для анализа YouTube видео при
помощи ИИ”

Выполнила: Сигаева Ксения Леонидовна

Дата 30.12.2024

Санкт-Петербург 2024

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	2
ВВЕДЕНИЕ.....	3
1 Телеграм бот: структура и логика.....	5
2 Мой вклад в разработку телеграм бота.....	8
3 Работа в команде.....	14
ЗАКЛЮЧЕНИЕ.....	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	18
ПРИЛОЖЕНИЕ.....	19

ВВЕДЕНИЕ

В наши дни невозможно не заметить растущую популярность и значимость видеоконтента. На протяжении уже достаточного количества лет YouTube как основная платформа для размещения длинных роликов является одним из самых используемых приложений, а сами видео прочно вошли в нашу жизнь. Этот формат подачи информации наиболее предпочтителен для людей, поскольку большинство из них является визуалами или аудиалами.

Сейчас снимать и выкладывать контент может кто угодно, так как современные технологии позволяют с легкостью это делать. С одной стороны, можно порадоваться такому прогрессу, но с другой – каждый день создаются миллионы роликов, и ориентироваться в них становится все сложнее и сложнее. Именно поэтому анализ видеоконтента является востребованной задачей в нашем мире. Очевидно, что делать это вручную не имеет никакого смысла: ни у кого нет столько времени. На помощь приходит автоматизация и, как следствие, искусственный интеллект. Его использование открывает совершенно новые возможности для обработки, анализа, структурирования материала и выделения главного.

Когда мы порассуждали о YouTube видео и ИИ, возникает вопрос: почему мы выбрали создание телеграм бота для этой цели, а не веб-сайта или приложения:

- доступность и простота использования: не требует установки дополнительного приложения, поскольку большинство пользователей уже имеют Telegram на своем устройстве,
- кроссплатформенность: работает одинаково на всех устройствах, поддерживающих Telegram, что позволяет охватить широкую аудиторию без адаптации под различные операционные системы и браузеры,
- интуитивный интерфейс: Telegram предоставляет пользователю привычный и знакомый интерфейс, что минимизирует время на

проектирование как в сайте, тем более что функционал бота узок и не требует каких-то сложностей в дизайне и представлении.

Суммируя все вышеизложенное, можно выделить, что актуальность проекта заключается в необходимости автоматизации анализа и обработки YouTube видео и удобстве и простоте использования Telegram для этого процесса.

Хотя изначально задумывалось, что бот будет в основном использоваться в учебных целях для поиска наиболее подходящего видео при изучении и разборе темы, он может быть полезен не только студентам, но и таким категориям людей, как:

- создатели контента и маркетологи: они смогут анализировать успешные ролики и выявлять темы, интересующие аудиторию,
- бизнес-аналитики: бот поможет анализировать видео-отзывы о продуктах и брендах,
- журналисты: у них будет возможность получить анализ интервью, документальных фильмов и шоу, не тратя час на просмотр.

Таким образом, целью этого проекта является создание телеграм бота, который сможет анализировать содержимое YouTube-видео, используя возможности искусственного интеллекта.

Для более структурированной работы над проектом были выделены следующие задачи:

- исследование возможностей и технологий ИИ для анализа видеоконтента,
- разработка серверной части телеграм бота,
- создание пользовательского интерфейса,
- тестирование.

1 Телеграм бот: структура и логика

1.1 Принцип работы телеграм бота

Результатом работы над данным проектом является телеграм бот, позволяющий пользователю получать краткий анализ видео по предоставленной ссылке. Даже из названия проекта очевидно, что для этого используется ИИ, однако как же это происходит?

Последовательность действий для пользователя довольно проста. Он открывает Telegram, заходит в чат бота и начинает с ним работать (нажимает на кнопку /start). Ему не нужно авторизоваться или вносить какую-то информацию о себе, что очень удобно. Всю нужную информацию о доступных действиях можно получить в /help. Если есть необходимость сменить язык, это легко делается в настройках (/settings). Для того чтобы получить обработку видео, ему надо отправить сообщение с ссылкой на него.

Теперь рассмотрим, что происходит со стороны бота. Он получает сообщение и выделяет из него ссылку, если находит ее. Из ссылки извлекается идентификатор видео и по нему с помощью специальной библиотеки запрашивается транскрипция. Далее текст форматируется, посредством “склеивания” бессвязных фраз в предложение. Перед передачей в ИИ транскрипция обрезается, чтобы избежать проблем с лимитом на количество токенов. После того, как мы удостоверились, что текст не превысит ограничения и точно поступит на анализ, его отправляют в языковую модель с сообщением о том, что с ним надо сделать. Когда приходит результат, его выводят пользователю в чат.

На протяжении всего процесса работы бота реализована обработка исключений, что делает его более устойчивым при возникновении неожиданных ошибок. Если какие-либо этапы, такие как анализ видео, взаимодействие с языковой моделью или обработка пользовательского

ввода, сталкиваются с проблемами, бот не падает, а информирует о возникшей ситуации соответствующим сообщением.

Более наглядно весь процесс работы бота представлен на рисунке 1. На нем показаны все основные связи.

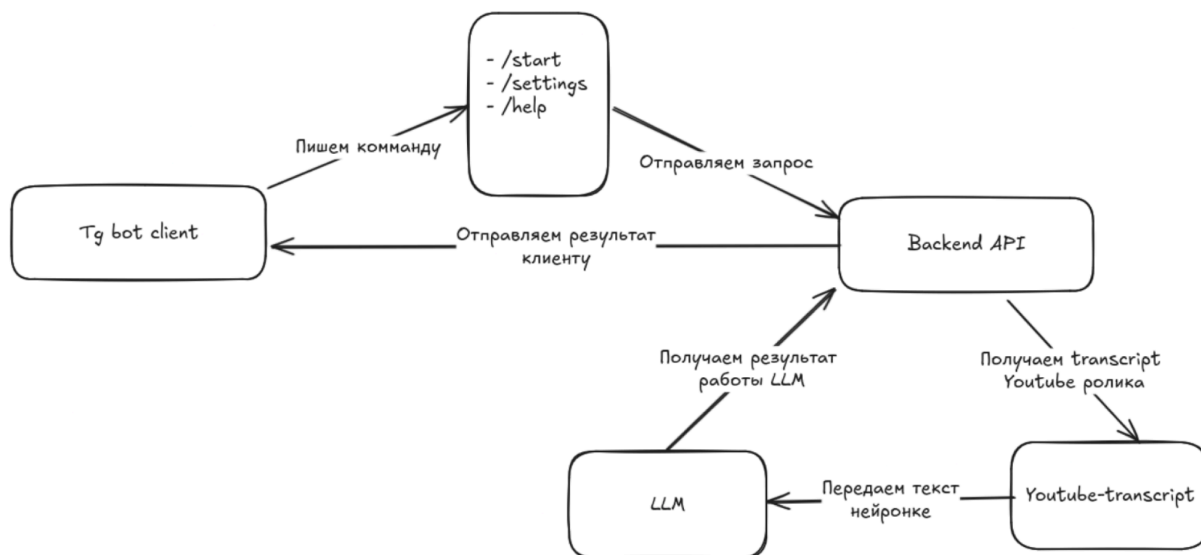


Рисунок 1 - Схема работы телеграм бота

1.2 Основные этапы реализации

Для того, чтобы разработка проходила равномерно и у каждого участника проекта была одинаковая нагрузка, были поставлены следующие задачи:

- создание телеграм бота с помощью библиотеки aiogram: добавление интерфейса с кнопками, принятие команд и сообщений от пользователей, их обработка,

- создание web-server с использованием технологий Flask: сервер отвечает за анализ данных и взаимодействие с языковой моделью, он обрабатывает данные, извлекает информацию из сообщения пользователя и отправляет результат обратно боту,

- написание запроса к серверу со стороны телеграм бота: эта задача отвечает за связь бота и сервера, она передает данные, полученные от пользователя “внутренностям” программы, т.е. если в бот отправляют ссылку на видео, она проверяется на принадлежность к YouTube и передается серверу, а ответ от него обрабатывается и выводится обратно клиенту,

- обработка полученных ссылок: по ссылке получается транскрипция с помощью модуля `youtube_transcript_api`, текст из нее форматируется, поскольку он представляет из себя разрозненные и несвязанные фразы,

- добавление запроса к LLM (Large Language Model): на этом этапе происходит отправка форматированного текста из транскрипции в языковую модель (YandexGPT) с соответствующим запросом, которая проводит анализ и возвращает интеллектуальный ответ,

- добавление возможности пользователю отправлять видеофайл и получать транскрипцию с видео: дополнительная задача, которая возникла в связи с проблемой автоматических запросов к YouTube, требует переписи некоторых функций, ответственных за получение метаданных из видео и субтитров (транскрипции),

- тестирование: заключительный этап разработки, в котором принимали участие все исполнители, он необходим для отлова багов и ошибок.

2 Мой вклад в разработку телеграм бота

2.1 Описание моей задачи

Моей задачей в этой работе было создание телеграм бота с помощью библиотеки aiogram. Таким образом, я отвечала за “интерфейс”, который будет доступен пользователю. Проще говоря, писала код для создания всех кнопочек, на которые нажимают при взаимодействии с ботом и которые впоследствии появляются как реакция.

В процессе реализации мною были добавлены следующие кнопки-команды бота:

- /start — команда для начала взаимодействия пользователя с ботом, которая выдает приветствие: "Добро пожаловать! Бот активирован"

- /help — команда, которая отображает список всех доступных действий с кратким описанием их функционала, а также предоставляет возможность перейти к ним,

- /settings — команда настроек, позволяющая выбрать русский или английский язык для дальнейшего взаимодействия путем нажатия на одну из предоставленных кнопок.

Также я работала над распознаванием ссылки из сообщения пользователя.

2.2 Реализация

Для создания бота была выбрана библиотека aiogram [1]. У нее есть несколько преимуществ, важных для этого проекта [2]. Она поддерживает асинхронность, что позволяет одновременно управлять большим количеством запросов и упрощает работу с API. Также она поддерживает последние версии Telegram Bot API, поэтому можно работать с новейшими и наиболее удобными технологиями. Кроме того, все взаимодействие с ботом поделено на модули, каждый из которых ответственен за свою

задачу. Это делает написание кода более легким, что особенно важно для начинающих разработчиков.

Каждая команда реализована в виде отдельной асинхронной функции [3]. Она принимает в качестве параметра данные, полученные от пользователя и возвращает нужное сообщение.

Помимо этого, для команды выбора настроек была создана инлайн-клавиатура с двумя кнопками русского и английского языка. После нажатия на одну из них срабатывает функция обработки обратной связи. Внутри нее происходит запись выбранного языка в словарь с ключом пользователем. В дальнейшем (при расширении проекта) эту информацию можно будет заносить в базу данных. По завершении на экран выводится сообщение, уведомляющее о выборе.

Отдельно стоит сказать о функции распознавания любого сообщения пользователя. Бот проверяет текст на наличие YouTube-ссылки. Это делается с помощью регулярного выражения [4], представляющего из себя шаблон ссылки. По нему ищется совпадение с имеющимися данными.

2.3 Ход работы

В целом, я думаю, что могу считать свою работу над проектом успешной. У меня получилось самостоятельно выполнить свою задачу по созданию телеграм бота с нужными функциями на основе библиотеки aiogram.

Поскольку это был мой первый опыт в создании ботов, поначалу было немного трудно привыкнуть к структуре кода, так как она отличается от того, что я писала до этого. Казалось, что это не известный мне за много лет python, а другой язык программирования. Однако прочитав документацию библиотеки, я смогла разобраться в новой логике и поняла, чем такой “необычный” вид кода объясняется. Это связано с использованием архитектуры, которая поддерживает асинхронное

выполнение задач и взаимодействие с внешними API, а также с событийно-ориентированным подходом, который лежит в основе работы ботов. Модуль aiogram предлагает оптимальный и лаконичный способ разработки, убирая излишнюю сложность, которая могла бы возникнуть при работе с API напрямую.

В процессе работы у меня возникла маленькая проблема, которой я хотела бы поделиться. Когда я создавала клавиатуру (две кнопки) для выбора языка в настройках, она оставалась активной на протяжении всего дальнейшего взаимодействия с ботом. То есть пользователь мог нажать на неё и получить сообщение о выборе, что будет совсем не актуально. Так, создание большого количества клавиатур во всем чате могло привести к неприятным последствиям. Интерфейс бота стал бы перегруженным и не таким интуитивным, как задумывалось. Кроме того нажатие и объявление события могло привести к конфликтам логики обработки и непредсказуемому поведению бота. Для ликвидации этой проблемы было решено удалять клавиатуру сразу после выбора, дабы предотвратить возможные проблемы.

Анализируя весь процесс работы над проектом, могу сказать, что она протекала планомерно. Я последовательно и постепенно выполняла свою задачу: сначала читала документацию и знакомилась с aiogram и его принципами работы, а потом начинала писать. Когда возникали вопросы и уточнения, консультировалась с командой и продолжала. Для меня особенно важно то, что я не делала все “в последний момент”, как это часто бывает. Я думаю, это потому что я работала не одна и чувствовала больше ответственности, чем обычно. Работа над проектом вошла в мой еженедельный график. В итоге, практически полностью моя задача были закончена к моменту написания промежуточного отчета.

2.4 Тестирование

Тестирование является важным этапом разработки любого проекта, не важно состоит ли он из нескольких простых функций или представляет из себя полноценное приложение. По окончании работы над своей задачей я решила провести свою собственную небольшую проверку кода, чтобы точно быть уверенной в корректности реализации.

Чтобы протестировать код, я создала свой телеграм бот, который называется trybot [5]. Для начала я попробовала все основные команды: /start, /help, /settings. Как мы видим на рисунке 2, они работают корректно.

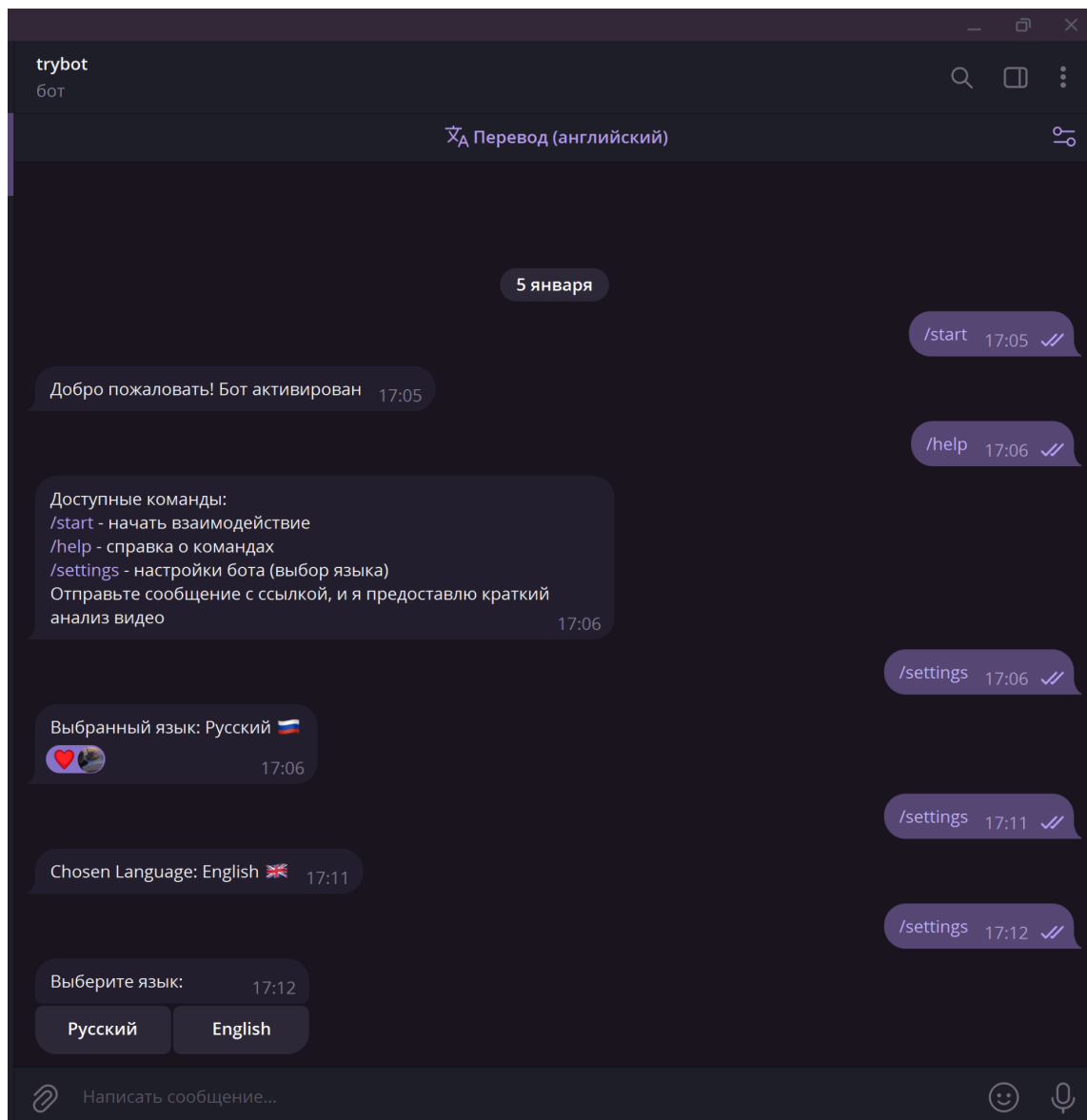


Рисунок 2 - Тестирование основных команд

Следующим шагом была проверка распознавания YouTube ссылки. Сначала я отправила простое сообщение с текстом, а потом – с ссылкой на одну из лекций. Бот понял, где содержится ссылка, а где – нет, что можно видеть на рисунке 3.

Все операции работали так, как это ожидалось, из чего следует вывод о об успешности тестирования и разработки.

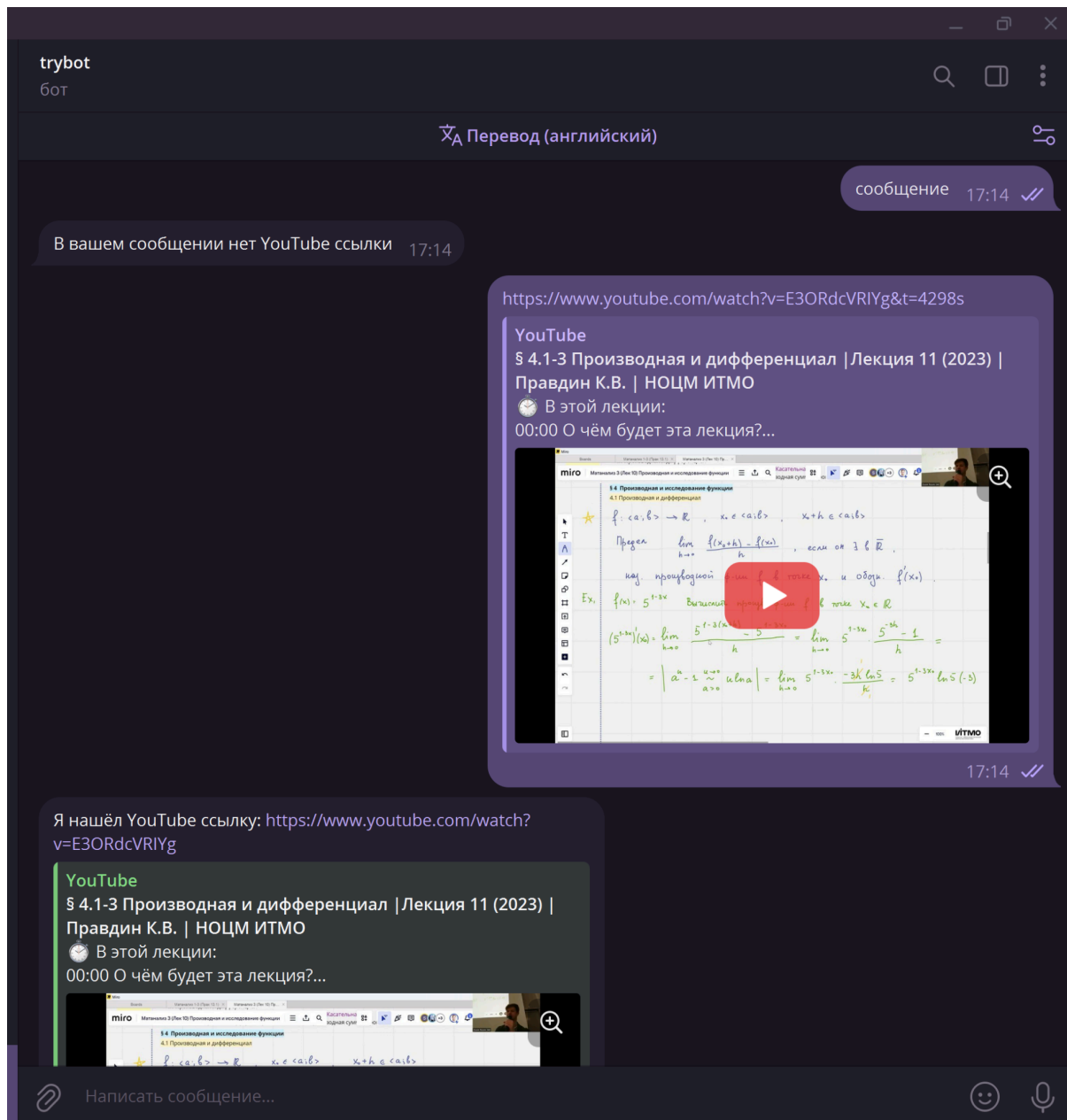


Рисунок 3 - Тестирование распознавания ссылки на видео

2.5 Навыки, полученные в результате работы над проектом

Поскольку это мой первый опыт в разработке ботов, все, что происходило было для меня новым. По окончании работы я получила такие навыки, как:

- создание телеграм-ботов с использованием библиотеки `aiogram`,
- работа с асинхронным программированием (`asyncio`),
- создание пользовательского интерфейса для телеграм бота,
- логирование и отладка.

Также во время работы я следила за тем, что делают другие участники команды, поэтому я освоила некоторые навыки, не входящие в мою задачу. Например:

- создание веб-сервера с использованием `Flask`,
- работа с LLM,
- обработка медиафайлов (извлечение метаданных и транскрипции).

3 Работа в команде

3.1 Взаимодействие с командой

За время работы над этим проектом мы наладили тесную связь в команде. Я считаю, что это стало ключевым фактором успешного завершения разработки телеграм-бота. Благодаря коммуникации наша команда смогла грамотно распределить задачи, что сделало процесс создания эффективным и продуктивным.

Для совместной работы и выгрузки кода были созданы два репозитория на платформе GitHub. Один предназначен для разработки серверной части бота, а другой – для интерфейса, который доступен пользователю, и работы с языковой моделью. Такое разделение позволило четко разграничить зоны ответственности и избежать повторений. Также каждый из нас создавал свою ветку, чтобы не мешать другим. Так у нас получилась слаженная работа, без конфликтов в коде.

Помимо этого, использование GitHub дало возможность следить за прогрессом коллег по проекту. Мы могли видеть, как и над чем работают остальные участники, что мотивировало сделать и свою часть быстрее, чтобы команда не отставала от намеченного графика. И если у кого-то возникали проблемы или вопросы руководителю было легко посмотреть, что произошло и дать обратную связь. Также кто-то из нас мог помочь другому с выполнением задачи, если вдруг нагрузка была распределена неравномерно или возникли непредвиденные сложности.

В процессе мы все стали ближе друг к другу, можно даже сказать, что этот проект в каком-то смысле связал нас. Мы, не стесняясь, уточняли мелкие детали в общем чате или коммуницировали лично в университете. Для меня лично этот проект стал первой командной работой, и данный опыт вселил уверенность и в успешность будущей совместной деятельности.

3.2 Взаимодействие с руководителем

На протяжении всей разработки бота наш руководитель организовывал процесс и обеспечивал условия для наиболее эффективной работы команды. Наши постоянные созвоны помогали делать все вовремя и не откладывать задачи на конец.

На начальном этапе наш магистрант организовал несколько вводных встреч. На одной мы с ребятами познакомились друг с другом, рассказали о своем предыдущем опыте в сфере IT и поделились своими ожиданиями от проекта и мотивацией при вступлении в команду. На следующей – наш руководитель представил общий план работы, подробно объяснив, что надо будет сделать, а также обозначил выделенные им задачи. Так, мы постепенно начали вникать в суть того, что нам предстояло. Далее у нас было время на размышление и краткое знакомство со всеми задачами. Это было необходимо для того, чтобы каждый понял, чем хочет заниматься, что для него предпочтительнее и интереснее.

В дальнейшем проводились регулярные встречи два раза в неделю. На них каждый из нас поочередно делился тем, что он сделал и как продвигается его задача. Если что-то не получалось, мы говорили об этом, и все вместе обсуждали, как лучше решить проблему. Наш руководитель давал советы и задавал нужное направление. После того, как мы заканчивали свой небольшой отчет, была возможность спросить о чем-то, не связанном с проектом.

Перед самой защитой мы провели заключительный созвон, на котором подвели итоги работы и поделились своим мнением о том, как все проходило и что нас особенно впечатлило. Наш магистрант подсказал, как лучше сделать презентацию и на чем сделать акцент во время выступления.

3.3 Оценка руководителя

Я считаю, что наш руководитель достойно исполнял свои обязанности во время работы над проектом. Он старался максимально эффективно организовать весь процесс, чтобы мы не только выполнили все, что требуется, но и сохранили мотивацию.

В начале работы магистрант таким образом распределил все задачи, чтобы мы не пересекались и не зависели друг от друга. Это сделало разработку максимально удобной. Также в течении всего времени он контролировал процесс на постоянной основе. К нему всегда можно было обратиться во время созвона, в общем чате или личным сообщением, причем время ответа было вполне приемлемым. А на последнем этапе он подготовил нас к презентации, особенно морально.

Отдельно стоит сказать, что помимо всей формальной части наш руководитель стремился поддерживать командный дух, сделать процесс работы интересным, а еще настаивал на том, чтобы мы больше коммуницировали между собой. Кроме того, он давал полезные советы, например, говорил о том, на какие мероприятия внутри университета стоит обратить внимание, а какие лишь кажутся привлекательными, но в действительности только отнимают время от учебы. И конечно, мы узнали о работе нашего магистранта и получили рекомендации, полезные в будущей карьере.

Подводя итоги, могу отметить, что лидерские качества нашего руководителя сыграли важную роль в проекте. Если надо было бы оценить его работу по десятибалльной шкале, то я бы честно поставила Коровину Александру Игоревичу 10/10. В результате взаимодействия с ним я не только узнала много всего нового по теме телеграм ботов, но и поменяла свое отношение к некоторым вещам. Кроме того, могу признаться, что наших созвонов я ждала с нетерпением, они были самым интересным из всего, что касалось учебы, благодаря лидеру.

ЗАКЛЮЧЕНИЕ

Первое, что хотелось бы написать в заключении, это то, что наш проект был успешно завершен, все задачи выполнены и цель достигнута: телеграм бот, анализирующий видео с YouTube создан. Все участники получили новый опыт и теперь могут самостоятельно написать код для бота.

Каждый завершил свою задачу, пусть и с небольшими подсказками от руководителя. Не все пошло, как ожидалось, поэтому для исправления некоторых функций, ответственных за получение транскрипции и отправку запроса к YouTube API, была создана дополнительная задача, что отражено в разделе, посвященном этапам реализации. Однако в конце бот стал работать как надо.

Моим вкладом в проект стало создание самого бота на основе библиотеки aiogram. Это важная и необходимая часть работы, поскольку именно в ней заключается разработка того, что доступно пользователю. Таким образом, что в итоге сделала я:

- пользовательский интерфейс, включающий команды /start, /help, /settings,
- распознавание ссылки на YouTube-видео,
- тестирование собственной задачи,
- участие в тестировании всего бота.

В завершении моего отчета отдельно хочется сказать про возможный путь развития нашего проекта. Во-первых, можно добавить поддержку других языков, кроме русского и английского. Во-вторых, можно реализовать возможность общения в чате с ботом, задавать ему уточняющие вопросы по видео. В-третьих, можно добавить визуализацию, вставлять графики, диаграммы или хотя бы просто кадры из видео. Работа с ИИ предоставляет огромные возможности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Aiogram – <https://docs.aiogram.dev/en/stable/>
2. Какую библиотеку на Python выбрать для создания телеграм-бота? – <https://habr.com/ru/companies/otus/articles/771110/>
3. Asyncio — Asynchronous I/O – <https://docs.python.org/3/library/asyncio.html>
4. Регулярные выражения в Python от простого к сложному.
Подробности, примеры, картинки, упражнения – <https://habr.com/ru/articles/349860/>
5. Telegram Боты на Aiogram 3.x: Первые Шаги – <https://habr.com/ru/companies/amvera/articles/820527/>

ПРИЛОЖЕНИЕ

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Наименование проекта:

Разработка телеграмм бота для анализа YouTube видео при помощи ИИ

Исполнитель:

Коровин Александр Игоревич

Сроки выполнения:

11.2024 (начало: месяц, год) – 12.2024 (окончание: месяц, год)

Цель проекта:

Разработать телеграмм-бота для анализа YouTube видео с использованием искусственного интеллекта.

Ключевые слова и словосочетания:

- анализ видео YouTube,
- API взаимодействия с нейросетью,
- искусственный интеллект,
- сводка содержания видео,
- таймкоды ключевых моментов видео,
- мультязыковая поддержка (русский и английский языки),
- телеграм-бот (конечный продукт).

Требования к проекту:

Бот должен принимать ссылки на ютуб видео и обрабатывать их. после обработки, юзер при помощи команд должен взаимодействовать с ботом и получать краткую информацию из видео. Также пользователь может прочитать инструкцию по использованию бота.

Бот должен работать во всех официальных приложениях Telegram (мобильных, веб- и десктопных версиях).

Бот должен проверять валидность ссылок и выдавать сообщение об ошибке, если ссылка недействительна.

Пользовательский интерфейс бота должен быть интуитивно понятным.

Содержание работы (этапы по срокам):

Этап работы	Срок выполнения	Документ о выполнении этапа
Получить необходимое оборудование и ПО, API	05.11.2024	Выписка о получении API
Реализовать доступ к YouTube видео	10.11.2024	График подключения к youtube.com
Разработать оболочку бота (Python, YandexGPT)	15.11.2024	Ссылка на бота
Написать алгоритм анализа видео	30.11.2024	Отчет о работе алгоритма
Внедрить алгоритм анализа в бота	02.12.2024	Ссылка на бота
Создать удобный пользовательский интерфейс	10.12.2024	Ссылка на бота
Разместить бота на сервере	12.12.2024	Ссылка на бота

Основной результат работы и форма его предоставления:

Результат: Телеграмм-бот для анализа YouTube видео, предоставляющий краткую сводку материала на языке запроса.

Форма представления: Готовый к использованию бот, доступный в Telegram.

Исполнитель:

Коровин Александр Игоревич

Заказчик:

Горлушкина Наталия Николаевна