

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

Факультет **Прикладной информатики**

Направление подготовки **09.03.03 Прикладная информатика**

Образовательная программа **Мобильные и сетевые технологии**

КУРСОВОЙ ПРОЕКТ

Тема: «Разработка телеграмм бота для анализа Youtube видео при помощи ИИ»

Обучающийся: (Афанасьев Дмитрий Михайлович, К3140)

Санкт-Петербург 2024

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
ВВЕДЕНИЕ	4
Цель проекта	4
Задачи проекта	4
Актуальность использования Telegram-ботов для анализа видео	5
Обзор существующих решений и их недостатков	6
1 Разработка проекта	8
1.1 Разделение обязанностей между участниками команды	8
1.2 Выбор технологий и инструментов	8
2 Реализация отдельных компонентов проекта	10
2.1 Разработка локального сервера	10
2.2 Получение субтитров	12
2.3 Работа с запросами на сервер	13
2.4 Telegram-бот	14
2.5 Анализ видео с использованием нейросети	14
3 Дополнительная функциональность	15
3.1 Распределение задач	15
3.2 Возможность загрузки MP4-файлов	15
3.3 Получение метаданных из YouTube видео	16
3.4 Перспективы расширения	16
4 Анализ проделанной работы	18
4.1 Достижения и успехи проекта	18
4.2 Сложности, с которыми столкнулась команда	18
ЗАКЛЮЧЕНИЕ	20
Выводы о проделанной работе	20
Влияние проекта на развитие навыков	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	21
ПРИЛОЖЕНИЕ	22

ВВЕДЕНИЕ

Просмотр видео на YouTube очень часто отнимает у нас много свободного времени. В этом нет ничего плохого, если целью просмотра является отдохнуть и расслабиться, однако если целью просмотра является получение информации, то зачастую у нас просто нет времени смотреть длинное видео. Так же иногда после просмотра видео мы понимаем, что оно было не совсем о том, о чем говорилось в названии, и, если бы мы заранее хотя бы примерно знали бы о содержании видеоролика, то точно бы не стали тратить на него свое время.

Или же другая ситуация: вам нужно узнать о том, какой именно процессор в конфигурации собираемого вами компьютера лучше всего подойдет для той или иной видеокарты. Вы нашли нужное видео, но оно идет почти час, где автор разбирает каждую мелочь в подробностях, и, перемотав в конец, вы понимаете, что вывода у видео нет. Чтобы в этой ситуации получить необходимые ответы, нужно полностью посмотреть видео. Тут и поможет наш бот.

С этими проблемами поможет разобраться именно наш бот.

Цель проекта

Перед нами стояла цель, заключающаяся в том, чтобы написать работающего телеграмм бота, который бы помогал пользователю анализировать видео с YouTube.

Задачи проекта

Перед командой были поставлены следующие задачи, указанные в техническом задании к проекту (см. приложение Техническое задание): создание сервера, реализовать работу с сервером, создание оболочки бота, реализовать работу с LLM, реализовать основные методы для получения

необходимой информации. Задачи были распределены следующим образом: я должен был написать сервер, Алиса реализовать работу с сервером, Анастасия работу с LLM, Ксения оболочку бота, Дмитрий должен был написать основные методы.

В ходе работы над проектом появились еще несколько задач: реализовать загрузку видео, реализовать получение субтитров с загруженного файла, реализовать получение метаданных с YouTube видео. Я должен был реализовать последние две задачи.

Актуальность использования Telegram-ботов для анализа видео

Перед тем как создавать приложение мы изучили наличие аналогов на рынке, однако ничего в открытом доступе в телеграмме мы не нашли. Значит наш проект уже будет являться уникальным в своей нише, такой как телеграмм боты.

Площадка в виде телеграмма была выбрана изначально, но всё же стоит затронуть тему ее актуальности. Для начала нужно сказать о том, что такое телеграмм. Telegram — это облачное многоплатформенное приложение для обмена сообщениями, которое позволяет пользователям отправлять текстовые сообщения, мультимедийные файлы, создавать группы и каналы, а также использовать встроенных ботов, что для нашего проекта является ключевым параметром. Оно отличается высокой скоростью работы, удобным интерфейсом и повышенной безопасностью благодаря функции шифрования данных. Telegram активно используется как для личного общения, так и для деловых и образовательных целей, предоставляя широкие возможности для коммуникации и организации контента. В данный момент телеграмм является самой популярной социальной сетью среди молодежи, абсолютно каждый имеет в ней аккаунт, каждый состоит в множестве бесед, читает множество каналов, и в целом телеграмм занимает существенную часть нашей жизни. Так

же в данный момент телеграмм работает наиболее стабильно, среди всех оставшихся социальных сетей. Именно поэтому бот, который будет находится в этой социальной сети, будет всегда доступен и удобен для использования. Его можно закрепить для быстрого доступа или быстро найти благодаря строке поиска.

Таким образом, можно заключить, что наш телеграмм бот будет востребован, удобен для использования, а значит актуален в наше время.

Обзор существующих решений и их недостатков

Из существующих решений наша команда нашла только аналог анализа видео от Яндекса. Это нейросеть, которая позволяет извлекать основные идеи из видеоматериалов и генерировать текстовый пересказ. Она использует мультимодальный анализ видео, аудио и текстовых данных для создания лаконичного и понятного конспекта. Однако она работает только в Яндекс браузере, по информации из документации [4], что делает её неудобной в использовании, как минимум из-за того, что большая часть YouTube просмотров происходит с мобильных устройств. Основное различие между инструментом Яндекса и нашим телеграмм ботом в том, что для работы нашего телеграмм бота не требуется устанавливать никакие плагины, а если у вас уже есть ссылка на видео, например вам учитель скинул видеоматериал лекции, то не нужно даже заходить на платформу YouTube. И, разумеется, наш бот работает на мобильных устройствах, что отдает ему приоритет, поскольку в наше время у всех пользователей телеграмма и YouTube имеется под рукой телефон, в котором наш бот работает. Так же стоит заметить, что наш бот работает не только на телефоне, а на абсолютно любом устройстве, где есть Телеграмм, вне зависимости от вычислительных мощностей машины, поскольку все вычисления происходят на сервере и не нагружают систему.

Из существующих решение еще можно выделить сами нейросети, которые мы используем. Возможно, покажется странным использование телеграмм бота, когда можно просто написать нейросети запрос и получить такой же ответ. Но тут возникает серьезная проблема. А какой именно нужен запрос? Ведь если просто отправить нейросети ссылку на видео, то нейросеть не перейдет по ней и не станет смотреть его и анализировать. А значит для анализа видео простая нейросеть никак не подойдет.

Таким образом, наш телеграмм бот на данный момент является наиболее приоритетным вариантом из всех существующих.

1 Разработка проекта

1.1 Разделение обязанностей между участниками команды

Как было уже сказано ранее, каждый получил по своей задаче с четким дедлайном. Наш тимлид строго соблюдал временные рамки и требовал выполнение каждой задачи не позднее положенного, поскольку без некоторых выполненных задач не было возможности начать следующие. Так задача Дмитрия напрямую зависела от моей, поскольку я должен был написать сервер, а Дмитрий на этом сервере работу с ссылками и получение субтитров. Задача Алисы зависела от моей, поскольку в её обязанности входило написать работу с сервером, который должен был быть написан мною.

1.2 Выбор технологий и инструментов

Важная часть в создании любого проекта – это выбрать технологии и инструменты, которые будут использоваться в процессе.

Основной инструмент, который мы использовали – это телеграмм. В нем проводились созвоны с командой и создавался телеграмм бот. Выбору телеграмм было много времени уделено во введении.

Для организации выдачи задач и учета прогрессии их выполнения была использована учебная платформа от Odoо. В этой платформе нам выдали задания, мы могли разбивать их на подзадачи и смотреть за выполнением проекта в реальном времени.

Для написания сервера была выбрана библиотека Flask. Это удобная и простая библиотека, которая позволяет создавать серверы. Она простая в использовании, одна из самых популярных на open source рынке. У нас был выбор между Flask и Django. Django тоже библиотека, которая выполняет ту же задачу, однако Django более сложен и многофункционален, однако для

нашего проекта было достаточно использовать Flask и его функционала нам более чем хватало. Была изучена документации к этой библиотеки [1].

В качестве нейросети была выбрана версия “Yandex GPT 4”. Перед нами стоял выбор Яндекс или Open AI, и мы решили выбрать отечественный продукт, который работает без VPN. В итоге мы выбрали версию “Yandex GPT 4”, самую актуальную на момент написания проекта. Была изучена документация по использованию API этой нейросети [4].

Созвоны с командой были важной частью нашего взаимодействия. Они вытеснили работу со специальной платформой, так как они проводились дважды в неделю и их было действительно много. Все всегда были в курсе проекта, понимали, что именно происходит и что нужно сделать, благодаря им все всё делали вовремя, так как при возникновении каких-то проблем вся команда быстро помогала их решать, таким образом, продуктивность работы сильно возросла.

2 Реализация отдельных компонентов проекта

2.1 Разработка локального сервера

Разработкой локального сервера занимался я. Как было написано ранее, для разработки была выбрана библиотека Flask. Она была выбрана из-за своей простоты и удобства использования. Она полностью подходила нам по своему функционалу. В ходе изучения этой библиотеки возникла проблема с версиями, в файле со всеми зависимостями в проекте requirements.txt была указана не самая новая версия этой библиотеки. Было решено изменить requirements.txt и обновить до новейшей версии, в последствии версия была обновлена до модифицированной, поддерживающей асинхронность. Для того, чтобы корректно интегрировать асинхронность в проект была изучена соответствующая библиотека [4].

Была получена ссылка на рабочий репозиторий, в котором была прописана файловая структура для сервера. Было принято решение не изменять структуру проекта. В файловой структуре была проблема с настройкой зависимостей, поскольку файл server.py находится в корне, а работа с сервером происходит на уровень ниже. Было изучено взаимодействие с файлом __init__.py, который является специальным файлом в Python, используемым для обозначения директорий как пакетов. Этот файл выполняет несколько функций: инициализирует пакет, управляет взаимодействием с ним, а также позволяет создавать пространство имён для модулей. В ходе работы с __init__.py были устранены все возникавшие проблемы, и была начата работа с сервером.

Была создана html страница для главной страницы сайта, чтобы было видно, что сервер работает. В качестве html страницы был взят документ из прошлого независимого проекта. Она помещена в templates/. Для нее был написан код для перехода на страницу /summarize. Переход осуществляется в

виде кнопки со скриптом. Была инициализирована страница /summarize, на которую и должен отправляться запрос.

Были получены все функции от нашей команды.

Для понимания работы с сервером необходимо было изучить модуль “youtube_transcript_api”, который в свою очередь и принимает субтитры с видео.

Возникла проблема с этим модулем: по умолчанию этот модуль ищет в видео английские субтитры, а если субтитров нет вообще, то выдается ошибка.

Было принято решение установить стандартный приоритет языков, по которым ищутся субтитры и была написана функция, которая показывает отсутствие субтитров. Таким образом, я немного залез в выполнение задачи Дмитрия.

Все функции были импортированы и началась работа с их взаимодействием.

Требовалось понять логику взаимодействия сервера с запросами. Как должен работать сервер: сервер получает в качестве запроса ссылку на видео -> сервер выводит json-ответ с транскрипцией видео.

Для того, чтобы корректно вызвать выполнение функции получения субтитров требуется корректное получение ID видео. Для этого существует функция, которая получает строку и получает из нее id. В качестве нее была написана пустая функция, которая в последствии была заменена на корректную Дмитрием в рамках своей задачи. Затем выполняется функция, которая по ID получает субтитры. При отладке этой функции возникла проблема: бесконечное ожидание ответа. Вскоре была предпринята попытка запустить на сервере VPN. В итоге получение ответа стало возможным. Таким образом, возникла глобальная проблема: VPN на сервере. Было принято решение, что на сервере должен стоять VPN, поскольку иначе дальнейшая разработка не представлялась возможной.

После получения функцией текста требуется отправить ответ сервера на запрос.

Для отладки сервера была написана программа, которая создает запросы. Она имеет вид: “введите ссылку: ... - субтитры”.

Таким образом начался процесс отладки. Были протестированы несколько типов видео:

- короткие,
- длинные,
- очень длинные,
- без субтитров,
- с большим количеством субтитров,
- русские,
- иностранные.

Все тесты пройдены, везде при тестировании был получен корректный ответ.

Все шаги выполнения сопровождались коммитами, которые требовались от нашего тимлида. Все коммиты были оформлены по госту, благодаря чему команде будет с ними проще работать.

Был переписан Readme файл. В нашем проекте Readme файл нужен для того, чтобы каждый член нашей команды смог правильно запустить и протестировать проект на своей машине. В нем был указан путь для запуска сервера и запуска отладки. Были прописаны следующие пункты: активация виртуального окружения, установка зависимостей из requirements.txt, запуск сервера, запуск отладки. Были написаны инструкции для win и unix-систем, так как команды у разных операционных систем отличаются.

2.2 Получение субтитров

Получением субтитров из видео занимался Дмитрий. Для получения субтитров использовался модуль “youtube_transcript_api”. Проблемы с использованием этого модуля и их решение были описаны выше. Требовалось

написать функцию, которая анализирует ссылку и извлекает из нее уникальный ID видео.

Пустая функция, написанная ранее была изменена на рабочую. Для реализации этой функции использовалась библиотека «re», которая позволяет использовать шаблоны. Таким образом, была написана функция извлечения уникального ID из любого типа YouTube URL адресов.

Далее для получения субтитров использовалась библиотека “youtube_transcript_api”, с помощью которой, имея уникальный ID видео, можно получить субтитры видеоролика. Возникла проблема с приоритетом языков, ее решение описывалось выше.

После получения субтитров ответ преобразовывается в необходимый формат, чтобы сервер корректно мог отправить обратно ответ на запрос.

Для функции получения субтитров была добавлена асинхронность.

Таким образом, была сделана работающая функция для получения транскрипции видеоролика и можно было начинать выполнение следующих задач.

2.3 Работа с запросами на сервер

Работой с запросами на сервер в нашем проекте занималась Алиса. Требовалось написать программу, которая отправляет запросы на запущенный сервер и получает ответ. В качестве запроса должна подаваться ссылка на видео, а в качестве ответа сервер должен предоставлять транскрипцию видеоролика.

Все поставленные задачи были выполнены, запросы корректно отправлялись на сервер и принимались с корректной информацией.

2.4 Telegram-бот

Созданием телеграмм бота занималась Ксения. Требовалось написать телеграмм бота с помощью библиотеки Aiogram. Стояла задача продумать функционал бота и создать интерфейс для удобной работы с ним. Были добавлены такие функции как смена языка и запуск анализа видео.

Все поставленные задачи были достигнуты, телеграмм бот был написан и протестирован. Интерфейсом команда осталась довольна.

2.5 Анализ видео с использованием нейросети

Анализом видео и работой с нейросетью занималась Анастасия. Стояла задача с использованием Api нейросети от Яндекса получить корректный анализ видео с помощью полученных субтитров. Стояла задача определиться с температурой запроса и конфигурацией запроса.

На созвоне команда решила, что температура ответа должна быть 0, что означает, что нейросеть использует только полученную информацию и не добавляет ничего от себя. Это гарантирует точность анализа видео и минимизирует отдаление от изначального смысла видеоматериала.

Был написан запрос, состоящих из множества параметров. Запрос включал в себя правильное оформление вывода, чтобы в телеграмме ответ выглядел естественно. Полученный ответ направляется к боту и сразу без обработки к пользователю.

Все поставленные подзадачи были выполнены в установленный срок.

3 Дополнительная функциональность

3.1 Распределение задач

Я и Грачев Дмитрий Александрович получили задачу реализовать получение субтитров не напрямую через YouTube, а через загрузку видео и обработки файла локально. Обязанности не были распределены лидером проекта, так что разбиение на задачи проводили мы сами. Всего было поставлено 3 задачи.

- Реализовать загрузку видео с YouTube,
- Реализовать получение субтитров с загруженного файла,
- Реализовать получение данных о видео: Заголовок, Описание, Канал и т.д.

Мы договорились распределить следующим образом: Дмитрий Александрович выполняет первое задание, а я все остальные.

3.2 Возможность загрузки MP4-файлов

Возникла проблема, заключающаяся в том, что, для того чтобы получать субтитры из загруженного файла, нужно иметь загруженный файл. Однако, посоветовавшись, мы пришли к решению, что для реализации этой части не нужно иметь видео именно с YouTube, а можно тестировать на обычном MP4 файле. Все проблемы были решены и стало возможно работать параллельно.

В качестве примера, на котором будет тестироваться модуль, был выбран 15-секундный видеоролик с одной фразой: “А я всё это вместе”.

Для получения субтитров из файла нужно было либо полностью написать свою программу, либо изучить имеющиеся варианты. Было принято решение остановиться на втором пункте. С помощью тимлида был найден Репозиторий “yt-dlp” — многофункциональный командный загрузчик аудио/видео.

Для нашей задачи yt-dlp – слишком большой модуль, поэтому было принято решение углубленно его изучить и выделить для себя лишь необходимый функционал.

На GitHub был изучен исходный код проекта. Была найдена необходимая функция.

Возникла проблема – у этой функции было слишком много зависимостей в проекте. Из-за этого было потрачено время на отрезание лишней ветвей и на максимальное сокращение кода.

В результате была написана функция, которая получает субтитры из MP4 файла.

Написанный модуль был перенесен в корень проекта в отдельную директорию.

3.3 Получение метаданных из YouTube видео

Для получения метаданных использовался модуль yt-dlp. Этот инструмент позволяет извлекать данные о видео, такие как заголовок, описание и дата загрузки, с использованием команды '--dump-json', которые были взяты из документации по использованию yt-dlp [2].

В этом модуле была найдена команда, которая принимает на вход уникальный идентификатор видео, и выводит всю информацию о видео в виде Json. Для запуска функции была использована библиотека subprocess. Для запуска функции были использованы параметры '--dump-json', '--no-playlist', которые отвечают за формат вывода и работу с плейлистом, в котором находится искомый видеоролик. Далее нужно было обработать полученный Json-материал. Было принято решение исключить работу с тегами видео и оставить только заголовок, название канала, дату загрузки видео, длительность видео, количество просмотров на текущий момент, описание.

Так же был добавлена возможность ловить ошибки при получении данных. Эта функция была добавлена в общий файл работы с ссылками.

3.4 Перспективы расширения

Данные модули на момент релиза проекта не были использованы, так как возникли проблемы с проху. Однако они позволяют добавить огромный функционал боту.

Благодаря этим решениям теперь возможна работа не только с YouTube, но и с обычными видео и даже кружочками. Это на самом деле большой функционал, поскольку на данный момент телеграмм кружочки нельзя расшифровывать без платной подписки, а зачастую нет времени или возможностей посмотреть срочный кружочек. Реализации получения субтитров с разных типов видео уже может выполняться в один шаг, поскольку все необходимые модули для этого есть.

Стоит заметить, что получение метаданных видео тоже очень важно для анализа видео, поскольку данные о канале, просмотрах, названию дают больше понимания нейросети о смысле видео, об аудитории, они помогают составить более точную картину. Для улучшения анализа видео сейчас требуется лишь изменить запрос, добавив в него полученные метаданные, тем самым сильно повысив качество ответа.

Метаданные так же могут помочь в дальнейшей реализации RAQ-системы. Была изучена статья [7], посвященная этому, для анализа дальнейшей интеграции этой системы. Поскольку сейчас стало возможным получить информацию о длительности видео, тем самым можно заранее подготовить разбиение по блокам, по участкам видео, тем самым сократив время анализа видео.

4 Анализ проделанной работы

4.1 Достижения и успехи проекта

Все поставленные задачи были выполнены в срок. Цель нашего проекта выполнена. Был написан телеграмм бот, который справляется со своей задачей. Добавлены возможности расширения проекта, которые указаны в главе с дополнительным функционалом.

Лично для меня проект был полезен тем, что я давно хотел изучить как работают сервера и зачем они нужны. Так же мне очень интересен полученный продукт, и я планирую его использовать в будущем.

Проект очень гибкий и расширить его функционал не составляет проблем. Я считаю этот проект успешным, так как мы полностью достигли всех целей.

4.2 Сложности, с которыми столкнулась команда

Сложностей, с которыми столкнулась команда, было несколько. Первая сложность – это язык разработки, который нам предлагал наш руководитель. Никто из нас не умел писать на JavaScript, однако Алексей настаивал именно на этом языке. Однако проблема была решена на этапе зарождения, и мы все перешли на всем нам известный Python.

Вторая сложность – это созвоны. Их было действительно много и для первокурсника тратить по 2 часа в неделю это очень дорого. Но эта проблема тоже была решена и созвоны сократились до 15-30 минут от изначального часа. Они стали более продуктивными и уже не отнимали столько времени, и были уже полностью по делу.

Третья сложность – это изучение материала. Лично мне приходилось изучать документацию многих библиотек, чтобы разобраться в их работе. Однако это никак не отразилось на проекте и всё было выполнено в срок.

ЗАКЛЮЧЕНИЕ

Выводы о проделанной работе

Перед нашей командой стояла цель создать телеграмм бота, который может анализировать видео. Цель достигнута. Был разработан телеграмм бот, удовлетворяющий всем поставленным требованиям. Все этапы, прописанные в техническом задании (см. приложение Техническое задание) выполнены в срок, весь указанный функционал внедрен. Все дополнительные задачи обговаривались и были выполнены в срок.

Влияние проекта на развитие навыков

Проект существенно расширил наши знания в области серверных технологий, работы с асинхронными запросами [3], интеграции инструментов для работы с видео [2] и использования современных нейросетей, таких как Yandex GPT [4].

Проект прокачал мои социальные навыки. Я вступал в дискуссии по поводу новых доработок, обсуждал их и выдвигал свои идеи по внедрению. Не слепо следовал инструкциям, а именно работал в команде.

Благодаря работе в команде я прокачал свои навыки в работе с Git. Наш тимлид предоставил нам материал для изучения Git и тот был изучен [5, 6].

Я немного времени уделил работе с LLM, но и это прокачало мои навыки в работе с API нейросетей. Проект дал мне возможность узнать много нового про запросы нейросети, например, что такое температура запроса.

В целом проект прокачал мои социальные и профессиональные навыки, помог мне развиваться в той сфере, которая мне интересна.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Flask. Русскоязычная документация [Электронный ресурс]. – URL: <https://flask-russian-docs.readthedocs.io/ru/0.10.1/> (дата обращения: 18.11.2024).
2. GitHub. yt-dlp: A youtube-dl fork with additional features and fixes [Электронный ресурс]. – URL: <https://github.com/yt-dlp/yt-dlp> (дата обращения: 21.11.2024).
3. Python Software Foundation. asyncio — Asynchronous I/O [Электронный ресурс]. – URL: <https://docs.python.org/3/library/asyncio.html> (дата обращения: 01.12.2024).
4. Яндекс. Yandex GPT. Быстрый старт [Электронный ресурс]. – URL: <https://yandex.cloud/ru/docs/foundation-models/quickstart/yandexgpt> (дата обращения: 01.12.2024).
5. Selectel. Как работать с ветками в Git [Электронный ресурс]. – URL: <https://selectel.ru/blog/tutorials/how-to-work-with-branches-in-git-git-branch/> (дата обращения: 01.12.2024).
6. Habr. Почему Git так сложно понять? [Электронный ресурс]. – URL: <https://habr.com/ru/articles/349860> (дата обращения: 20.11.2024).
7. Habr. Что нового в Python 3.12: все ключевые изменения [Электронный ресурс]. – URL: <https://habr.com/ru/articles/779526> (дата обращения: 28.11.2024).

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
- Мультиязыковая поддержка (русский и английский языки)
- Телеграмм-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