

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»
(Самарский университет)

Институт информатики и кибернетики
Кафедра технической кибернетики

Отчет по лабораторной работе №2

Дисциплина: «Инженерия данных»

**Тема: «Применение n8n для реализации пайплайна обработки видео с
использованием LLM моделей»**

Выполнил: Казаркина Д.Д.

Группа: 6233-010402D

ЗАДАНИЕ

1. Telegram Bot принимает либо ссылку на видео, либо сам файл видео.
2. Если пришла ссылка — скачать видео; если пришёл файл — использовать его напрямую. В обоих случаях извлечь аудиодорожку с помощью ffmpeg.
3. Сгенерировать субтитры (EN), используя auto_subtitle.
4. Выполнить перевод EN→RU с помощью LLM из HuggingFace (допустим API; предпочтительно — собственный сервер vLLM/LLama).
5. Добавить полученные (RU) субтитры в исходное видео.
6. Отправить результат обратно пользователю в Telegram.

ЛАБОРАТОРНАЯ РАБОТА

1. Техническое описание пайплайна

Архитектура. Система реализует n8n-workflow, реагируя на отправку сообщения пользователем в Telegram-бота. Сообщение проверяется на то, файл это или ссылка, если файл – используется как бинарный, если ссылка – скачивается во временное хранилище. Далее идет ряд http-запросов к эндпоинтам развернутого в docker-контейнере сервиса video-translation. Поочередно выполняются следующие этапы: извлечение из видео субтитров при помощи модели openai/whisper в файле формата .srt, перевод полученного файла на русский язык при помощи LLM (deepseek-ai/DeepSeek-V3.2), инкрустация полученных субтитров в исходное видео. Далее файл отправляется обратно пользователю в чат в Telegram. В самом конце, а также в случае возникновения ошибок, в сервис отправляется запрос на очистку временных файлов.

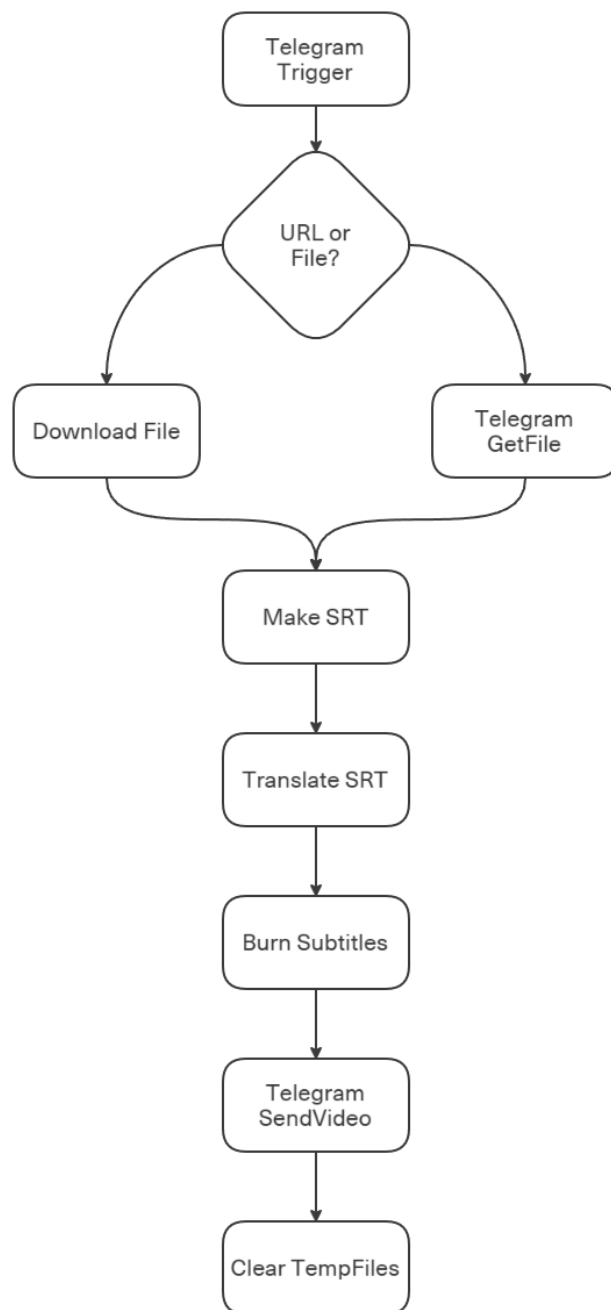


Рисунок 1 – Схема пайплайна

Telegram Trigger. Для работы с telegram-webhook необходимо обзавестись https. Для этого организуем туннель при помощи утилиты tuna (аналог ngrok, работающий в РФ), она выдает нам https-ссылку, ее вставляем в переменную окружения WEBHOOK_URL при запуске контейнера n8n. Далее настраиваем триггер на сообщения от пользователя, внося Credentials нашего telegram-бота в ноду.

URL or File? Match-нода, проверяет содержимое входных данных из предыдущей ноды, если там есть video_id – это файл, если сообщение содержит в себе «http» – это ссылка на видео.

Download File. Загружает файл по ссылке во временное хранилище.

Telegram GetFile. Загружает файл напрямую из Telegram.

Make SRT. Эндпоинт extract-audio принимает видеофайл, далее при помощи ffmpeg вырезается аудиодорожка, на следующем шаге ии-модель «openai/whisper» составляет субтитры из полученной аудиодорожки. Модель сама умеет распознавать отдельные фразы, на выходе получаем список с таймкодами и текст. Далее это упаковывается в .srt-файл и отправляется обратно.

Translate SRT. Эндпоинт translate принимает .srt-файл, далее упаковывает в http-запрос для инференса ии-модели «deepseek-ai/DeepSeek-V3.2». В тело запроса помещаются инструкции по типу «переведи только текст на русский язык, сохрани форматирование, не комментируй». В ответе получаем переведенное содержимое .srt-файла, упаковываем сырой текст в файл и отправляем обратно.

Burn Subtitles. Эндпоинт burn-subtitles принимает .srt-файл субтитров и исходный видеофайл, далее при помощи ffmpeg субтитры инкрустируются в видео. Полученный видеофайл отправляется обратно.

Telegram SendVideo. Отправляет полученное видео обратно пользователю в чат с ботом.

Clear TempFiles. Эндпоинт clear при получении запроса очищает папку /tmp сервиса.

2. Развертывание системы

Создание и запуск контейнеров. Все компоненты развертываются через Docker Compose. Файл `docker-compose.yml` описывает 2 сервиса: `n8n` (сам `nodemation`) и `video-translation` (сервис с эндпоинтами обработки). После команды `docker-compose up -d` сервисы запускаются в изолированных контейнерах. Также создан файл `.env.example` с примерами переменных окружения.

Настройка Telegram бота. Через BotFather в Telegram создан новый бот командой `/newbot`. После указания имени и `username` бота система генерирует уникальный API-токен.



Рисунок 2 – Создание Telegram бота

Запуск пайплайна. Workflow в `n8n` начинает работу по триггеру, триггером служит отправка пользователем сообщения нашего боту в Telegram.

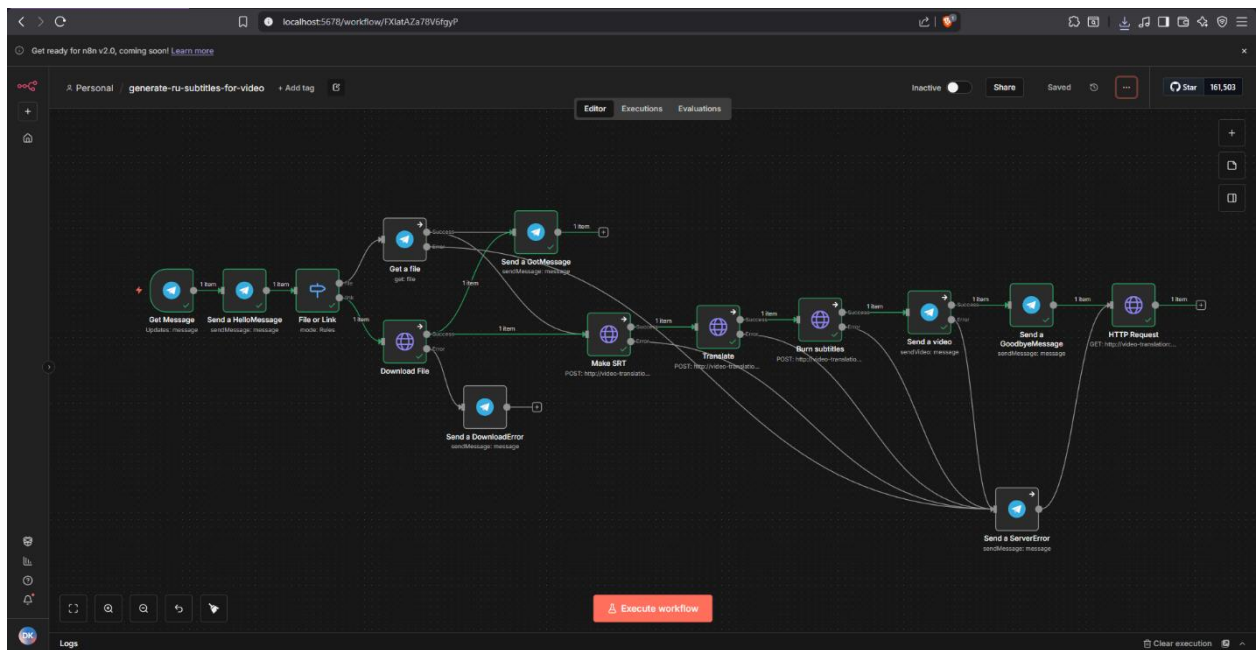


Рисунок 3 – Вид workflow в UI n8n

3. Результаты выполнения

В результате мы получаем бота в Telegram, которому можно скинуть ссылку на видео или само видео, получить несколько сообщений по ходу обработки и в итоге получить видео с наложенными субтитрами, что продемонстрировано на рисунке 4.

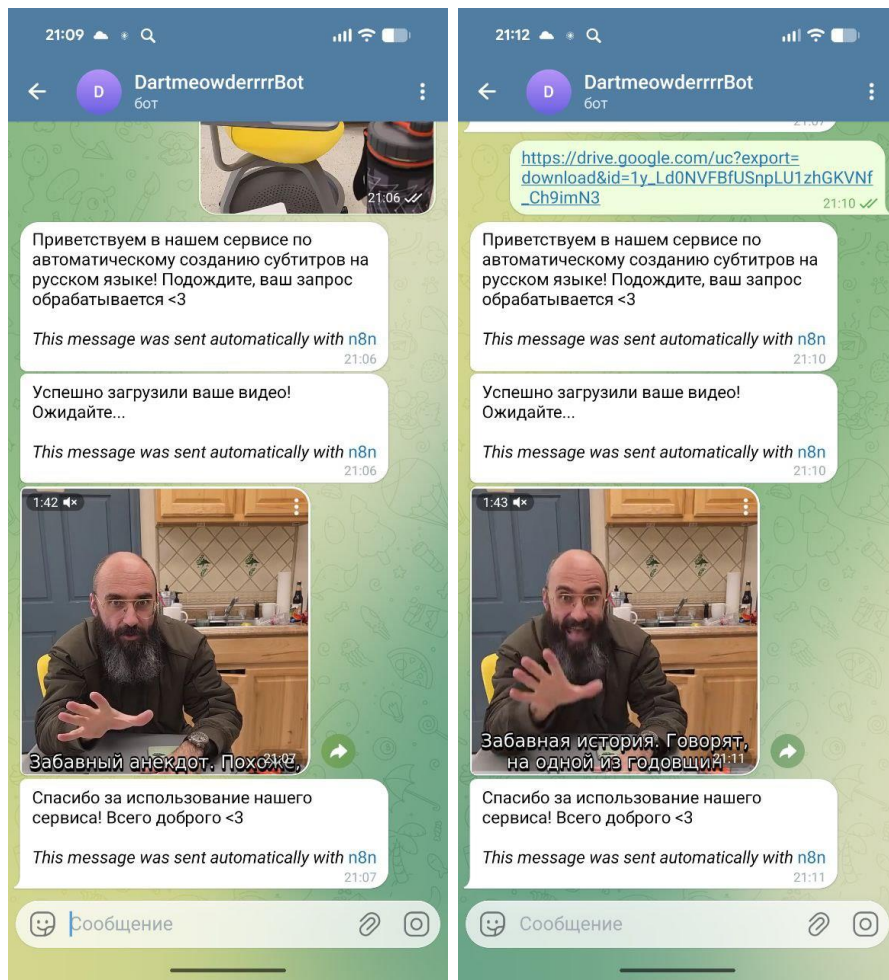


Рисунок 4 – Демонстрация работы в чате с ботом

ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы был успешно реализован n8n-workflow для создания русских субтитров, наложенных на видео на иностранном языке. Был поднят локальный n8n-инстанс и был создан сервис video-translation с несколькими эндпоинтами для создания субтитров на языке оригинала при помощи ии-модели openai/whisper, перевода полученных субтитров и инкрустации их в исходное видео. В качестве интерфейса для работы с API был использован Telegram-bot.

Основные сложности были связаны с изучением нового инструмента n8n, а также с получением https-хука для работы с Telegram API – локально это не работает, везде для тестов советуют ngrok, но он блокирует соединение по IP (возможно только для РФ), было необходимо найти стабильную альтернативу.