

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

ЛАБОРАТОРНАЯ РАБОТА № 2  
Применение `p8n` для реализации пайплайна обработки видео с  
использованием LLM моделей.

по курсу  
Инженерия данных

Группа 6232

Студент \_\_\_\_\_ Р.А. Зотов  
(подпись)

Преподаватель \_\_\_\_\_ Р.А. Парингер  
(подпись)

## АРХИТЕКТУРА

В качестве основного инструмента оркестровки был выбран n8n, развернутый в Docker-контейнере. Весь процесс построен в виде визуального графа (workflow), где данные передаются от узла к узлу.

В ходе выполнения задания реализовывалась архитектура и гита. Архитектуру можно увидеть на рисунке 1

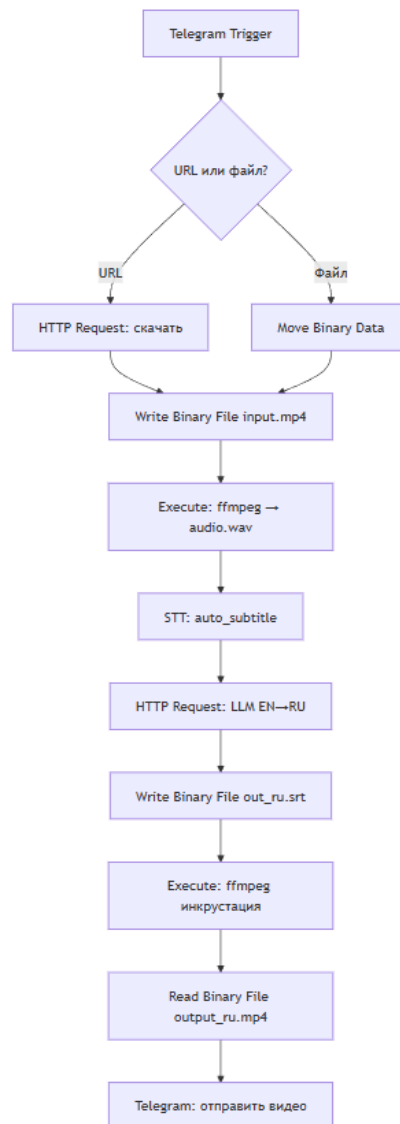


Рисунок 1 – Архитектура n8n

Система построена на принципах микросервисной архитектуры, где каждый модуль выполняется в изолированном Docker-контейнере.

Список основных компонентов и их назначение:

1. Оркестратор процессов (n8n) – управляет логикой: принимает данные, вызывает внешние сервисы, принимает решения (ветвления) и обрабатывает ошибки.
2. Локальный сервер Telegram Bot API – нужен для того, чтобы обрабатывать большие файлы (>20МБ). Сохраняет файлы в локальную директорию и n8n получает к ним доступ.
3. Сетевой туннель (ngrok) – чтобы связать сервера Telegram и наш локальный. Без него телеграм просто не сможет отправить webhook и прочее.
4. Модуль обработки медиа – модулями служат **yt-dlp** и **FFmpeg**. **yt-dlp** отвечает за скачивание файлов, FFmpeg используется для получения аудиодорожки, конвертации и наложения субтитров.
5. Сервис распознавания речи - контейнер с нейросетью Whisper (модель large-v3), предоставляющий API-интерфейс. Он принимает аудиофайлы и отдает текст с таймкодами.
6. Сервис перевода – тут особенность изначально использовалась Llama, потом qwen, а потом я вообще изменил подход на «судья». Проблема с тем, что бывает такое, что модель начинает галлюцинировать и вести себя не как переводчик, а как агент, поэтому добавил google translate, чтобы отлавливать такие ошибки.
7. Модуль логики и валидации – python скрипты, нужны для конвертации из json и обратно, очистки Whisper и когда qwen галлюцинирует.

## РЕЗУЛЬТАТЫ

На рисунке 2 представлен скриншот реализованного воркфлоу из n8n.

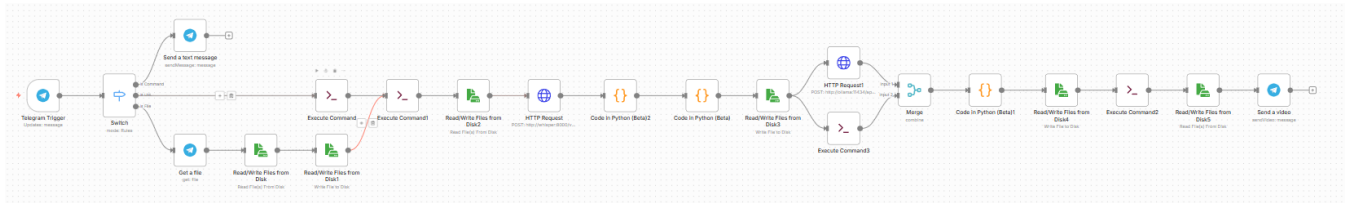


Рисунок 2 – Скриншот воркфлоу из n8n

Триггером служит сообщение в Telegram. Первым делом специальный узел Switch анализирует тип входящих данных. Если пользователь прислал ссылку (например, на YouTube), поток направляется в ветку с командой `yt-dlp` для скачивания ролика. Если прислан файл — он сохраняется напрямую. Для унификации процесса скачанные файлы сохраняются во временную директорию.

Следующим этапом происходит подготовка аудио.

Для распознавания речи используется HTTP-запрос к локально развернутому сервису Whisper. Сервис возвращает готовые субтитры в формате SRT на английском языке. Чтобы избежать проблем с дублированием фраз, был добавлен Python-скрипт для очистки и нормализации полученного текста.

Далее особенность, которая упоминалась выше, логика «судьи» для перевода текста. Из-за того, что LLM добавляет лишние фразы или вообще галлюцинирует и не выдает, что надо добавил гугл переводчик. (в основном проблема в том, что ломаются тайм коды, но при этом перевод лучше, хотя и не всегда)

Дальше происходит вжигание субтитров в виде и отправка пользователю.

На рисунке 3 и 4 представлен пример работы tg бота. В первом случае боту на вход передается видео, которое он сохраняет и обрабатывает, во втором случае мы кидаем ссылку, и он уже скачивает и переводит.



Рисунок 3 – Пример работы tg бота, при отправке видео

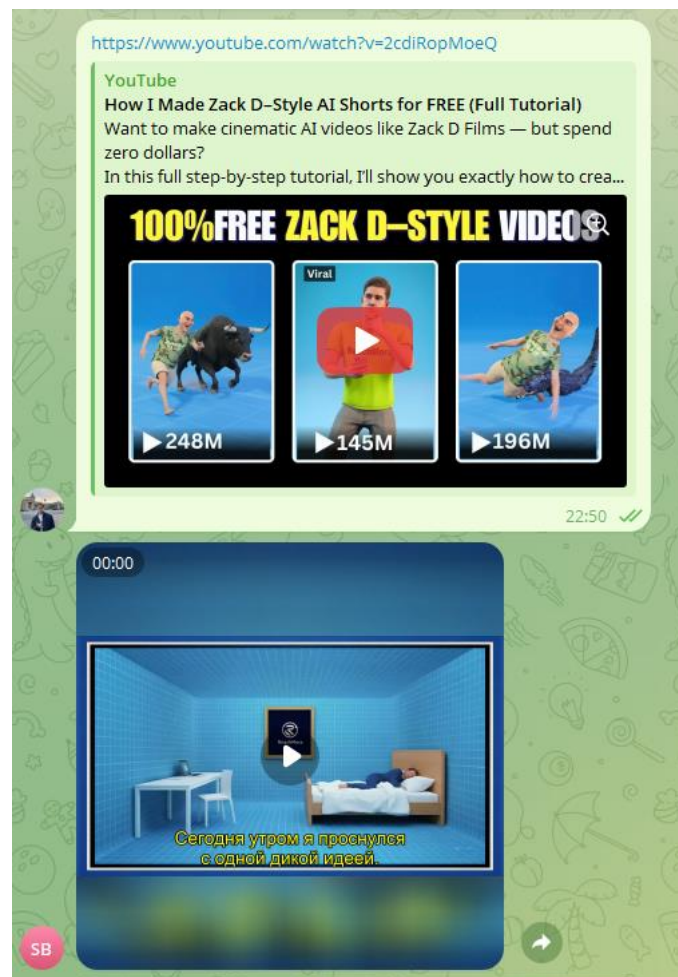


Рисунок 4 – Пример работы tg бота, при отправке ссылки

На рисунке 5 можно увидеть, работу workflow в active режиме (тестили работу боту с друзьями.

Nov 28, 22:57:49  
Succeeded in 21.248s

Nov 28, 22:56:24  
Succeeded in 31.733s

Nov 28, 22:53:26  
Succeeded in 38.716s

Nov 28, 22:50:54  
Succeeded in 4m 40.078s

Nov 28, 22:49:19  
Succeeded in 46.731s

Nov 28, 22:48:09  
Succeeded in 38.547s

Nov 28, 22:47:53  
Succeeded in 102ms

Send a video Success in 1.594s

Input Output

ok

result

message\_id : 179

from

id : 8552368908

is\_bot : true

first\_name : Subscribe AI Lab Bot

username : SubscribeLab\_bot

chat

id : 381860384

first\_name : Roman

last\_name : Zotov

username : Romazotov

type : private

date : 1764358289

Рисунок 5 – Запущенные контейнеры

6

## ВЫВОДЫ

Во время выполнения работы я столкнулся с многими проблемами (которые я, возможно, сам себе накрутил/создал и мог не решать). Для работы нужно было настроить сетевой туннель Ngrok или белый IP (чего у меня), о чем я не знал и пытался сделать через n8n Tunnel (в моменте я удивлялся почему вот сейчас работает Telegram Triger, а минуту назад нет или через минуту). Также одна из первых проблем была в том n8n, когда в скрипте я пытался вызвать установку ffmpeg и пришлось создавать свой образ n8n с уже встроенным ffmpeg.

Дальше проблемой было, что n8n настроен локально и workflow не сохраняется. После реализации первой версии дал затестить друзьям, они скинули большой файл, файл с малым количеством текста и ссылку на youtube. И можно понять все сломалось.

Телеграмм бот не может обрабатывать большие файлы на телеграммных серверах, пришлось настраивать локальный сервер (это не сложно, но из-за этого изменилась логика программы, не нужно стало скачивать видео, нужно было перемешать его в другую директорию, или копировать).

Чтобы скачать через GET запрос, нужно чтобы ссылка прям на видео вела (может такие и есть, но хотелось, чтоб прям с любого сайта качалось), поэтому пришлось прикручивать yt-dlp, и убирать http запрос на Execute Command.

Третий баг связан с видео, где мало текста, но много звуков. Whisper создает субтитры, и там много раз что-то повторяется, когда это попадает в LLM, она начинает вести себя не как переводчик, а как агент и спрашивает, что мне надо делать, хотя логика уже вся задана в промте, что очень раstraивает, так как обработка долгого видео занимает продолжительное количество времени. Так что была реализована логика судьи (LLM не может гугл переводчик поможет).

Еще был баг с отправкой больших файлов ботом. Вроде как p8n реализован на Node.js и не может хранить в одной переменной больше 500мб (то есть из узла чтения в узел отправки данные передать невозможно). В целом решается отправкой прямо из локального хранилища, но я не смог затестить (очень долго работают модели, но в будущем обязательно сделаю).

Для меня работа с p8n была новой, до этого с ним не сталкивался. В целом работа была очень интересной и занимательной. Понравилось, что можно не так сложно/долго реализовать вполне готовый к использованию инструмент (пусть и с проблемами).