

Для разработки системы выдачи рекомендаций выбирались язык программирования из двух представителей: Python [4] и C# [5] (табл.1).

Таблица 1

Результаты анализов языков программирования

Показатели	Python	C#
Скорость выполнения	3	4
Наличие библиотек, связанных с обработкой текста на русском языке	Да	Нет
Наличие библиотек для создания программного интерфейса по архитектуре REST API	Да	Да

Оценка проводилась по следующим параметрам:

- скорость выполнения – как быстро выполняется исполняемый программный код. Оценка в пределах [0; 5] – где 0 – самая минимальная скорость исполнения, а 5 – самая высокая;
- наличие библиотек, связанных с обработкой текста на русском языке;
- наличие библиотек для создание программного интерфейса по архитектуре REST API.

По итогам, было принято решение использовать язык программирования Python. Несмотря на то, что C# обеспечивает хорошую работу с архитектурой REST API ни чуть ни меньшую чем Python, последний обладает большим количеством библиотек, созданных для работы с русскоязычным текстом.

1. Анатомия рекомендательных систем. Часть первая. – URL: <https://habr.com/ru/companies/lanit/articles/420499/>;
2. Кратко о типах архитектур программного обеспечения, и какую из них выбрали мы для IaaS-провайдера. – URL: <https://habr.com/ru/companies/1cloud/articles/424911/>;
3. REST API: для чего нужен и как работает. – URL: <https://yandex.cloud/ru/docs/glossary/rest-api/>;
4. Python. – URL: <https://ru.wikipedia.org/wiki/Python>;
5. Краткий обзор языка C#. – URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/overview>.

УДК 004

**РАЗРАБОТКА ПРОГРАММНОЙ СИСТЕМЫ
ДЛЯ ФОРМИРОВАНИЯ СТЕНОГРАММ ЗАСЕДАНИЙ ДУМЫ
ГОРОДА ВЛАДИВОСТОКА**

Д.О. Сергеев, бакалавр

*Владивостокский государственный университет
Владивосток. Россия*

Аннотация. В современном государственном управлении оперативная подготовка стенограмм заседаний имеет большое значение. Ранее ручная транскрипция требовала до трех рабочих дней. Разработанная программная система сокращает это время до 2,5–3 часов, сохраняя точность $\geq 90\%$, используя нейронные сети, в частности модель Whisper от OpenAI.

Ключевые слова: распознавание речи, стенограмма, нейронные сети, Whisper, веб-сервис, автоматизация документооборота, фронтенд и бэкенд.

DEVELOPMENT OF A SOFTWARE SYSTEM FOR GENERATING TRANSCRIPTS OF THE VLADIVOSTOK CITY DUMA MEETINGS

Abstract. *In modern public administration, the operational preparation of meeting transcripts is highly important. Previously, manual transcription required up to three working days. The developed software system reduces this time to 2.5–3 hours while maintaining ≥90% accuracy, using neural networks, particularly OpenAI's Whisper model.*

Keywords: *speech recognition, transcript, neural networks, Whisper, web service, document flow automation, frontend and backend.*

В современных органах власти важную роль играет оперативная подготовка протоколов заседаний. В Думе города Владивостока расшифровка аудиозаписей ранее осуществлялась вручную, что требовало от одного до трёх рабочих дней для обработки одной записи продолжительностью от одного до полутора часов. Это существенно замедляло процессы документооборота. Целью данного проекта стала разработка программной системы, позволяющей автоматизировать процесс расшифровки аудиозаписей заседаний, сократить время обработки до 2,5–3 часов при обеспечении точности расшифровки не менее 90 %.

Ручная расшифровка аудиозаписей является трудоёмким и ресурсозатратным процессом. Она требует внимательности, высокой концентрации и значительных затрат времени даже со стороны опытных специалистов. В условиях современной цифровизации государственное управление требует быстрого доступа к актуальной информации, что невозможно обеспечить при ручной обработке данных. Автоматизация процесса расшифровки на основе нейросетевых технологий позволяет значительно повысить производительность труда, уменьшить вероятность ошибок, связанных с человеческим фактором, и ускорить передачу данных между различными уровнями управления [1].

Для решения поставленной задачи в ходе проекта была выбрана нейронная сеть Whisper, разработанная OpenAI [2]. Данная модель представляет собой современный алгоритм автоматического распознавания речи (ASR), обученный на многомиллионных наборах данных, включающих в себя разнообразные языки, акценты, фоновые шумы и стили речи. Whisper обеспечивает высокую устойчивость к шумовым помехам и способно точно распознавать даже сложные аудиозаписи, что особенно важно в условиях проведения заседаний, где возможны перекрёстные разговоры и неблагоприятные акустические условия.

Научная новизна данного проекта заключается в интеграции нейросетевого алгоритма распознавания речи в программную систему электронного документооборота муниципального уровня. Были разработаны архитектурные решения, обеспечивающие параллельную обработку больших объёмов аудиоданных с возможностью масштабирования вычислительных ресурсов. Система сочетает в себе технологии backend- и frontend-разработки, машинного обучения и баз данных, что позволяет обеспечивать не только высокую точность распознавания, но и удобство работы конечного пользователя [3].

Целью исследования являлась разработка удобной, надёжной и масштабируемой системы автоматической расшифровки заседаний Думы города Владивостока с использованием современных нейросетевых технологий. Для достижения цели были определены задачи:

- 1) анализ требований к качеству и скорости обработки данных;
- 2) проектирование архитектуры приложения;
- 3) интеграция модели Whisper для распознавания речи;
- 4) создание интуитивно понятного пользовательского интерфейса;
- 5) проведение всестороннего тестирования системы;
- 6) выполнение оптимизации кода для повышения производительности и устойчивости системы при высоких нагрузках.

Методологической основой проекта послужили методы системного анализа, объектно-ориентированного проектирования информационных систем, алгоритмы обработки

естественного языка (NLP) и методы машинного обучения [4]. Для реализации серверной части использовался фреймворк Flask на языке программирования Python, что позволило обеспечить высокую скорость обработки запросов и масштабируемость. Клиентская часть разработана с использованием стандартных веб-технологий – HTML, CSS и JavaScript, что обеспечило простоту интерфейса, его высокую адаптивность и совместимость с большинством современных браузеров без необходимости установки дополнительных компонентов.

Ключевым этапом обработки аудиофайлов является не только преобразование аудиосигнала в текст, но и последующая диаризация речи, которая позволяет определить, какие фразы принадлежат какому участнику заседания. Диаризация выполняется после первичного распознавания текста с помощью алгоритмов кластеризации аудиосегментов по акустическим признакам. Затем к каждой реплике добавляются временные метки начала и окончания, что позволяет точно синхронизировать текст с оригинальной записью. Процесс обработки аудиофайла можно посмотреть на рис. 1.



Рис. 1. Процесс обработки аудиофайла

Физическим сервером для работы системы служит вычислительная платформа с установленной графической картой NVIDIA GeForce GTX 1080, обладающей 11 ГБ видеопамяти. Эта модель обеспечивает необходимую производительность для параллельной обработки чанков аудиофайлов без значительных задержек. Использование GTX 1080 позволило достичь компромисса между стоимостью решения и его вычислительными возможностями, обеспечив быструю обработку записей длительностью до 60 минут за 12–15 минут реального времени.

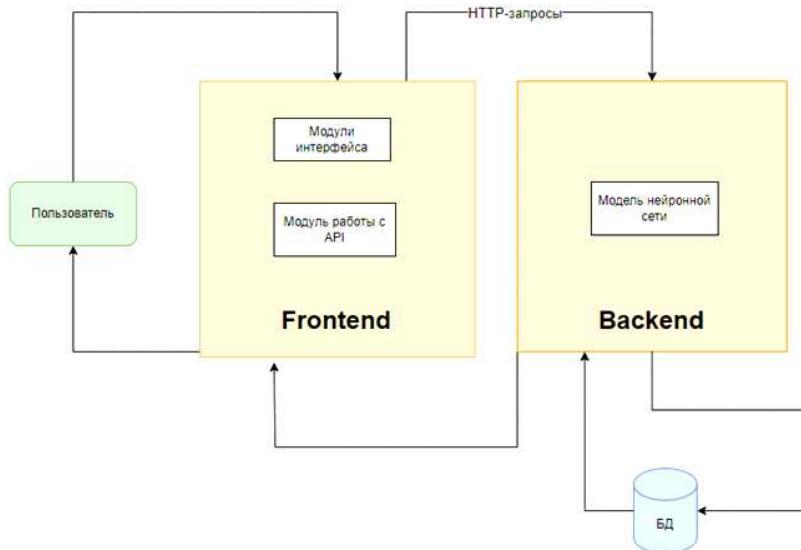


Рис. 2. Программная архитектура системы

Программная архитектура построена по модульному принципу: отдельные компоненты – загрузка файлов, распознавание речи, диаризация, хранение данных – функцио-

нируют как независимые сервисы, взаимодействующие через RESTful API. Это обеспечивает гибкость масштабирования системы, упрощает обновление отдельных компонентов и повышает отказоустойчивость (рис. 2).

Для удобства пользователей разработан лаконичный и интуитивно понятный интерфейс (рис. 3). Страница загрузки аудиофайлов позволяет в несколько кликов отправить запись на обработку, после чего пользователь может просматривать результат в структурированном виде: с разбивкой по спикерам и точными временными метками. Все элементы интерфейса оптимизированы под различные устройства.

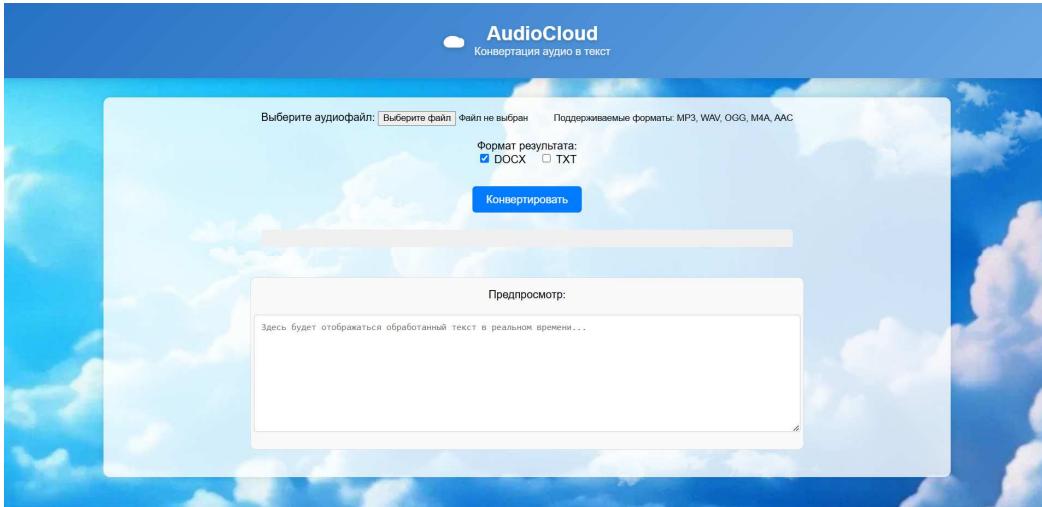


Рис. 3. Главная страница системы

В рамках тестирования проведено использование набора из 20 аудиофайлов заседаний с общей продолжительностью более 15 часов. Система продемонстрировала среднюю точность распознавания 94–96 %, а время обработки одного файла составило 12–15 минут. Основные проблемы, выявленные в процессе тестирования, касались распознавания одновременной речи нескольких участников, а также сильных шумовых помех.

Особое внимание в проекте уделено вопросам дальнейшего развития системы. В ближайших планах – внедрение системы личных кабинетов пользователей. Реализация данной функции позволит обеспечить индивидуальный доступ к загруженным и обработанным файлам, создавать историю работы с системой для каждого пользователя, управлять заданиями в очереди обработки и предоставлять персонализированные отчёты.

Таким образом, разработанная программная система демонстрирует высокую эффективность применения нейросетевых технологий для автоматизации стенографирования заседаний органов власти. Система позволяет существенно сократить временные затраты на подготовку стенограмм, повысить точность данных и упростить работу сотрудников органов власти с текстовыми материалами.

-
1. Гришко В.В. Технологии автоматизации документооборота. – Москва: Инфра-М, 2019. – 224 с.
 2. OpenAI. Whisper ASR System. – URL: <https://openai.com/research/whisper>
 3. Радомский А. Архитектура веб-приложений: принципы и практика. – Москва: ДМК Пресс, 2020. – 304 с.
 4. Поляков С.А. Нейронные сети и машинное обучение. – Санкт-Петербург: Питер, 2021. – 512 с.