МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра системного анализа и автоматического управления

СОЗДАНИЕ ПРИЛОЖЕНИЯ НА ОСНОВЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ ДЛЯ РАСПОЗНАВАНИЯ ГОЛОСОВЫХ КОМАНД

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета Компьютерных наук и информационных технологий
Одарченко Владимира Олеговича

Научный руководитель
доцент, к. ф.-м. н. _____ М. В. Корнилов

Заведующий кафедрой
к. ф.-м. н., доцент _____ И. Е. Тананко

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы. Объем аудиоданных – лекций, интервью, подкастов, совещаний – растет с каждым годом. Ручная расшифровка и анализ отнимают много времени. Особенно если нужно не только получить текст, но и понять, кто из участников и в какой момент говорил. Существующие локальные решения решают эти задачи по отдельности или в ограниченном формате, однако на рынке отсутствует единый офлайн-инструмент, который бы объединял полный цикл – распознавание речи, диаризацию, голосовое управление с верификацией диктора и удобную навигацию по расшифровке (полнотекстовый поиск и фильтрацию по говорящим).

Цель бакалаврской работы – создать комплексную офлайн-систему анализа аудиозаписей: она должна объединять диаризацию, транскрибацию, голосовое управление с верификацией диктора и навигацию по расшифровке.

Поставленная цель определила **следующие задачи**:

1. Проанализировать существующие методы и программные продукты для диаризации, распознавания речи и голосового управления.
2. Спроектировать клиент-серверную архитектуру, обеспечивающую локальное взаимодействие вычислительного модуля и интерфейса.
3. Разработать модуль диаризации и распознавания речи, выдающий структурированные данные о сегментах аудио.
4. Реализовать модуль голосового управления с верификацией диктора.
5. Создать клиентское приложение с интерфейсом для загрузки, визуализации, поиска, фильтрации и истории.
6. Протестировать систему и оценить ее характеристики.

Методологические основы работы в области обработки аудиосигналов и распознавания речи представлены в трудах таких ученых, как H. Bredin (pyannote.audio), D. Snyder (x-vectors), A. Radford (Whisper), M. Ravanelli (SincNet), а также российских исследователей: К.Л. Капуста, И.С. Кипяткова, И.А. Кагиров, В.Х. Багманов, А.Х. Султанов, Е.С. Шамакова, В.С. Коломойцев.

Теоретическая значимость заключается в обобщении и систематизации современных нейросетевых методов диаризации и автоматического распознавания речи, а также в демонстрации возможности их эффективной интеграции в рамках клиент-серверного приложения, полностью функциониру-

ющего офлайн.

Практическая значимость состоит в создании готового к использованию программного продукта, позволяющего выполнять транскрибацию и диаризацию аудиозаписей полностью офлайн, что гарантирует конфиденциальность данных. Приложение не требует от пользователя специальных технических навыков.

Структура и объем работы. Бакалаврская работа состоит из введения, четырех разделов, заключения, списка использованных источников и двух приложений. Общий объем работы – 67 страниц, из них 40 страниц – основное содержание, включая 8 рисунков и 5 таблиц. К работе прилагается флеш-накопитель с программной реализацией. Список использованных источников информации содержит 38 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Анализ предметной области и существующих решений» посвящен обзору ключевых задач обработки аудиоданных – диаризации дикторов, автоматическому распознаванию речи и голосовому управлению – а также анализу существующих методов и программных продуктов.

В разделе рассмотрены традиционные подходы к диаризации, основанные на мел-частотных кепстральных коэффициентах (MFCC) и кластеризации (k-средних, агломеративная кластеризация). Показаны их ограничения: чувствительность к шумам, необходимость задания числа дикторов, нестабильность на коротких репликах. В качестве современной альтернативы представлены нейросетевые методы с использованием дикторских вложений (d-векторы, x-векторы), которые формируются глубокими сетями и обеспечивают более высокую устойчивость к шумам и вариативности голоса. Выполнен обзор открытых библиотек диаризации – Pyannote.audio, SpeechBrain, Kaldi – и обоснован выбор Pyannote.audio (предобученная модель speaker-diarization-3.1, простота интеграции, работа офлайн).

Для задачи распознавания речи проведен сравнительный анализ современных нейросетевых моделей. Рассмотрена архитектура Whisper (кодер-декодер на базе трансформера) и ее оптимизированная реализация faster-whisper. Приведены характеристики моделей разного размера (tiny, small,

medium, large-v3-turbo) в сравнении со специализированными российскими системами GigaAM v3 и T-One. Отмечено, что, несмотря на более высокую точность российских моделей на русском языке, выбор сделан в пользу Whisper благодаря многоязычной поддержке (включая русский, английский, немецкий и др.) и открытой лицензии.

Голосовое управление рассмотрено как задача распознавания ограниченного набора команд с критическим требованием малой задержки. Описаны два подхода – обнаружение ключевых слов и классификация коротких аудиофрагментов. Предложено использовать для команд модель Whisper tiny в связке с детектором голосовой активности WebRTC VAD, что позволяет отсекать тишину и шум. Опциональная верификация диктора реализуется через сравнение дикторских вложений (модель pyannote/embedding) с эталоном по косинусному сходству.

Выполнен анализ существующих локальных программных продуктов (SpeechPulse, aTrain, VoxInput/Scribe). Показано, что ни один из них не объединяет одновременно транскрибацию, диаризацию, голосовое управление с верификацией диктора и средства навигации по расшифровке (полнотекстовый поиск, фильтрацию по дикторам, историю обработки). Сделан вывод о необходимости создания комплексной офлайн-системы.

Второй раздел «Теоретические основы используемых методов» посвящен фундаментальным принципам цифровой обработки аудиосигналов, а также нейросетевым архитектурам, которые легли в основу разработанной системы. В разделе представлен анализ методов, выбранных для реализации диаризации, распознавания речи, верификации диктора и обнаружения голосовой активности.

Рассмотрены базовые процедуры цифровой обработки: дискретизация (с частотой 16 кГц, согласно теореме Котельникова), квантование (16 бит), спектральный анализ через быстрое преобразование Фурье и построение спектрограмм. Эти операции являются необходимым этапом подготовки аудиосигнала для работы нейросетевых моделей.

Подробно описан метод дикторских эмбеддингов – компактных векторных представлений голоса фиксированной размерности (512 чисел), инвариантных к содержанию речи и устойчивых к шумам. В работе используется модель pyannote/embedding с архитектурой x-vector, дополненной обучаемы-

ми SincNet-фильтрами. SincNet заменяет классические банки фильтров параметризованными sinc-функциями, что повышает устойчивость признаков. Для сравнения эмбеддингов применяется косинусное сходство, а порог верификации (0,4) выбран экспериментально как компромисс между долей ложных принятий и ложных отклонений. Модель используется одновременно для двух задач – диаризации (через библиотеку pyannote.audio) и верификации диктора (через отдельный эндпоинт сервера), что экономит память.

Рассмотрены нейросетевые архитектуры для аудио: сверточные сети (CNN) для извлечения локальных признаков из спектрограмм, рекуррентные сети (LSTM) для учета временной динамики, а также трансформеры с механизмом самовнимания. Трансформеры не имеют ограничений на длину последовательности и хорошо параллелизуются, что делает их предпочтительными для распознавания речи. На этой архитектуре построена модель Whisper (кодер-декодер), которая преобразует спектрограмму в текст. Для ускорения инференса использована библиотека faster-whisper на движке CTranslate2 с поддержкой квантования (`int8_float16` для модели tiny, `float16` для large-v3-turbo). Модель large-v3-turbo имеет сокращенное число слоев декодера, что дает высокую скорость при минимальной потере точности.

Представлен метод обнаружения голосовой активности (VAD) на основе WebRTC VAD. Сигнал нарезается на кадры по 20 мс, для каждого вычисляются логарифмические энергии в шести частотных поддиапазонах, после чего гауссова смесь из двух компонент (речь / шум) принимает решение. Выбран самый строгий режим (Aggressive), что минимизирует ложные срабатывания. При накоплении тишины более 0,9 секунды речевой фрагмент отправляется на распознавание команд. Такой подход снижает нагрузку на сервер и исключает передачу фонового шума.

Теоретический анализ показал, что дикторские эмбеддинги (x-vector с SincNet) являются универсальным инструментом для решения задач диаризации и верификации. Трансформерная архитектура Whisper обеспечивает высокое качество многоязычного распознавания речи, а ее оптимизированная версия faster-whisper позволяет работать на ограниченных ресурсах. WebRTC VAD эффективно выделяет речевые фрагменты в непрерывном потоке. Выбранные методы образуют теоретический базис для проектирования и реали-

зации системы.

Третий раздел «Проектирование системы анализа аудиозаписей» посвящен разработке архитектуры, требований, серверной и клиентской частей, а также базы данных.

Сформулированы функциональные требования: загрузка аудио, диаризация и транскрибация с привязкой ко времени и диктору, визуализация (цветовая шкала и текстовая расшифровка), полнотекстовый поиск, фильтрация по дикторам, история обработки, экспорт (TXT, CSV, SRT), голосовое управление с опциональной верификацией. Нефункциональные требования: полная локальность, производительность, масштабируемость (выбор модели Whisper), надежность, удобство интерфейса.

Спроектирована двухзвенная клиент-серверная архитектура. Сервер на Python (FastAPI) выполняет ресурсоемкие вычисления (нейросети). Клиент на .NET (WPF) реализует интерфейс и логику. Компоненты общаются через локальный REST API, данные не покидают устройство – обеспечена конфиденциальность. Клиент построен по паттерну MVVM.

Серверная часть включает модули для маршрутов, управления моделями и утилит. Реализована динамическая загрузка моделей Whisper: в памяти хранится только одна модель, при смене старая выгружается из видеопамяти. Это позволяет эффективно использовать ресурсы GPU, в том числе на устройствах с ограниченной памятью. Для голосовых команд используется модель tiny с квантованием, а для полной транскрибации по умолчанию применяется наиболее точная large-v3-turbo, при этом пользователь в любой момент может выбрать любую другую модель из линейки.

Клиентская часть построена на .NET 8 с WPF. MainViewModel – центральное звено, содержит коллекции сегментов, свойства для привязки к интерфейсу и команды. Сервисы инкапсулируют задачи: HTTP-клиент, воспроизведение аудио, построение временной шкалы (ScottPlot), голосовое управление с VAD, работа с историей. Интерфейс содержит вкладки «Анализ» и «История».

База данных – встраиваемая SQLite. Таблица History содержит поля: идентификатор, имя файла, путь, дату, длительность, число дикторов и сегменты в формате JSON. Это упрощает чтение/запись и избавляет от лишних таблиц.

Спроектирована модульная клиент-серверная архитектура с динамической загрузкой моделей, сервисной организацией клиента и хранением истории на SQLite. Все решения соответствуют требованиям локальности, масштабируемости и удобства сопровождения.

Четвертый раздел «Практическая реализация и тестирование» посвящен описанию разработки, ключевых инженерных решений и экспериментальной оценке созданной системы. В разделе отражены результаты самостоятельной работы по реализации серверной и клиентской частей, а также тестирование на реальных аудиозаписях.

Сервер реализован на Python 3.12 с использованием виртуального окружения venv. Ключевые библиотеки: FastAPI + Uvicorn (веб-сервер), faster-whisper и pyannote.audio (распознавание и диаризация), librosa, soundfile (обработка аудио), PyTorch с CUDA 12.6 (вычисления на GPU). Клиент разработан в Visual Studio 2022 на .NET 8 с WPF. Использованы пакеты: ScottPlot.WPF (визуализация), NAudio (аудиозахват и воспроизведение), WebRtcVadSharp (VAD), Newtonsoft.Json (сериализация), Microsoft.Data.Sqlite (база данных). Предобученные модели: pyannote/speaker-diarization-3.1, pyannote/embedding, а также все размеры faster-whisper (tiny, small, medium, large-v3-turbo). Тестирование проводилось на ПК с Intel Core i5, 16 ГБ ОЗУ, NVIDIA GeForce RTX 2070 (8 ГБ видеопамяти).

Разработана функция `get_or_load_whisper` (модуль `models.py`), реализующая динамическую загрузку моделей с использованием словаря-кэша. При запросе модели, отличной от текущей загруженной, старая модель удаляется из памяти, вызывается `torch.cuda.empty_cache()`, затем загружается новая. Для tiny использовано квантование `int8_float16`, для large-v3-turbo – `float16`. Это позволяет работать на GPU с 8 ГБ памяти. В модуле `utils.py` создан список `HALLUCINATION_PATTERNS` и функция `clean_transcript`, отсекающая «галлюцинации» Whisper (осмысленные фразы на тишине) и строки короче трех символов. Для транскрибации подобраны параметры инференса: `beam_size=10`, `temperature=0.0`, `no_speech_threshold=0.5`, `vad_filter=True` с порогом 0.6 и минимальной речью 250 мс. Для эндпоинта `/command` используется модель tiny с теми же параметрами без склеивания сегментов.

Реализован `VoiceService`, который непрерывно захватывает аудио с микрофона (NAudio), использует WebRTC VAD в строгом режиме (Aggressive).

Кадры по 20 мс проверяются на наличие речи; при накоплении тишины более 0,9 секунды буфер превращается в WAV и отправляется на сервер. Это исключает передачу шума и снижает ложные срабатывания. Реализован `TimelinePlotService` на ScottPlot: для каждого сегмента строится прямоугольник от start до end, цвет назначается по хешу диктора, ось X форматируется в минуты:секунды, оси Y скрыты для компактности. `HistoryService` на SQLite сохраняет результаты анализа сразу после завершения: в таблицу `History` записываются имя файла, путь, дата, длительность, количество дикторов и JSON сегментов. Реализованы методы `SaveAsync`, `LoadAllAsync`, `GetByIdAsync`, `DeleteAsync`, `UpdateSegmentsAsync`. После каждого анализа запись автоматически появляется на вкладке «История», возможны повторный просмотр, удаление, обновление имен дикторов.

Проведена оценка транскрибации на десятиминутном аудиофайле с диалогом двух русскоговорящих дикторов. Для четырех моделей Whisper измерено полное время обработки (диаризация + транскрибация) и ошибка на уровне слов (WER). Результаты: tiny – 1 мин 15 с, WER 20,1 процентов; small – 2 мин 10 с, WER 11,8 процентов; medium – 4 мин 5 с, WER 8,4 процентов; large_v3_turbo – 5 мин 40 с, WER 7,9 процентов. Подтверждена масштабируемость: выбор модели позволяет балансировать между скоростью и точностью. Тестирование голосовых команд (10 команд, по 20 попыток на каждую) показало точность 87 процентов в тихой обстановке и 74 процента при фоновом шуме около 50 дБ без верификации; с верификацией – 85 процентов и 70 процентов соответственно. Верификация при пороге косинусного сходства 0,4 дала долю ложных отклонений (FRR) около 10 процентов и долю ложных принятий (FAR) около 8 процентов. Проверка истории и экспорта: записи корректно сохраняются в SQLite, открываются, фильтруются, удаляются; экспорт в TXT, CSV, SRT формирует файлы с правильной структурой (временные метки, дикторы, текст).

Реализована полностью работоспособная офлайн-система анализа аудиозаписей. Разработаны и внедрены ключевые инженерные решения: динамическая загрузка моделей Whisper, фильтрация «галлюцинаций», голосовое управление с VAD, визуализация временной шкалы, история на SQLite. Экспериментальное тестирование подтвердило выполнение всех функциональных требований, приемлемую производительность (обработка 10 минут диа-

лога за 5–6 минут на флагманской модели) и хорошее качество транскрибации (WER 7,9 процентов для large-v3-turbo). Точность голосовых команд достаточна для практического использования.

ЗАКЛЮЧЕНИЕ

В ходе выпускной квалификационной работы спроектирована и реализована комплексная офлайн-система анализа аудиозаписей. Она объединяет диаризацию дикторов, автоматическое распознавание речи и голосовое управление с верификацией диктора. Поставленные задачи решены. Проведен анализ существующих методов и программных продуктов в области диаризации, распознавания речи и голосового управления, и на его основе выбраны технологии. Спроектирована клиент-серверная архитектура: Python-сервер и WPF-клиент взаимодействуют локально. Разработан серверный модуль диаризации и автоматического распознавания – на выходе структурированные данные о речевых сегментах с временем, диктором и текстом. Реализован модуль голосового управления с функцией верификации диктора. Создано клиентское приложение с временной шкалой, текстовой расшифровкой, полнотекстовым поиском, фильтрацией по дикторам и историей обработанных файлов. Тестирование подтвердило работоспособность всех основных компонентов. Цель работы – создание офлайн-системы анализа аудиозаписей – достигнута. Приложение может быть полезно студентам, журналистам, преподавателям и другим специалистам, работающим с записями лекций, интервью и совещаний. Система не требует подключения к сети и доступна пользователям без специальной технической подготовки.

Основные источники информации:

1. Bredin H., Yin R., Coria J.M. et al. pyannote.audio: neural building blocks for speaker diarization // IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). – 2020. – С. 7124–7128.
2. Snyder D., Garcia-Romero D., Sell G., Povey D., Khudanpur S. X-Vectors: Robust DNN Embeddings for Speaker Recognition // 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2018. – С. 5329–5333.
3. Radford A. et al. Robust Speech Recognition via Large-Scale Weak Supervision // arXiv:2212.04356. – 2022.

4. Ravanelli M., Bengio Y. Speaker Recognition from Raw Waveform with SincNet // 2018 IEEE Spoken Language Technology Workshop (SLT). – 2018. – С. 1021–1028.
5. Капуста К. Л., Кипяткова И. С., Кагиров И. А. Аналитический обзор интегральных моделей и стратегий распознавания речи на основе архитектуры трансформер // Информационно-управляющие системы. – 2024. – № 5. – С. 2–15.
6. Багманов В. Х., Султанов А. Х. Цифровая обработка сигналов : учебное пособие. – Уфа : УГАТУ, 2012. – 133 с.
7. Шамакова Е. С., Коломойцев В. С. Метод голосовой идентификации диктора // Научный журнал. – 2024. – № 7. – С. 72–78.
8. WebRTC Voice Activity Detection // webrtcvad: Python interface to the WebRTC VAD. – URL: https://pypi.org/project/webrtcvad/ (дата обращения: 08.05.2026).