МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Южно-Уральский государственный университет (национальный исследовательский университет)» Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО СЕРВИСА ДЛЯ ГЕНЕРАЦИИ АННОТАЦИЙ К АУДИОФАЙЛАМ

Научные руководители:
ст. преподаватель кафедры СП,
Силкина Н.С.

м.н.с. кафедры СП,
Старков А.Е.

Автор:
студент группы КЭ-402
Беляков М.С.

Челябинск, 2024 г.

Актуальность

- Рост распространения разговорных аудиозаписей
- Тенденция к широкомасштабному внедрению automatic speech recognition (ASR) систем
- Рост производительности домашних персональных компьютеров

Цель и задачи исследования

Цель:

Разработать интеллектуальный сервис для генерации аннотаций к аудиофайлам.

Задачи:

1. Выполнить анализ предметной области
2. Выбрать технологии реализации
3. Обучить акустическую модель
4. Создать языковую модель
5. Создать веб-сервис для демонстрации ASR

Обзор аналогов

1. **Deepgram**
   1. WER ~ 10%
   2. Обработка в реальном времени
   3. Поддержка более 30 языков и акцентов
2. **Google Cloud Text to Speech**
   1. WER ~ 8-12%
   2. Обработка в облаке
   3. Поддержка более 120 языков и диалектов
3. **Whisper AI**
   1. WER ~ 5-10%
   2. Поддержка более 50 языков
   3. Возможность работы на локальных устройствах
   4. Хорошая работа с шумом и низкокачественными записями.

Архитектура системы

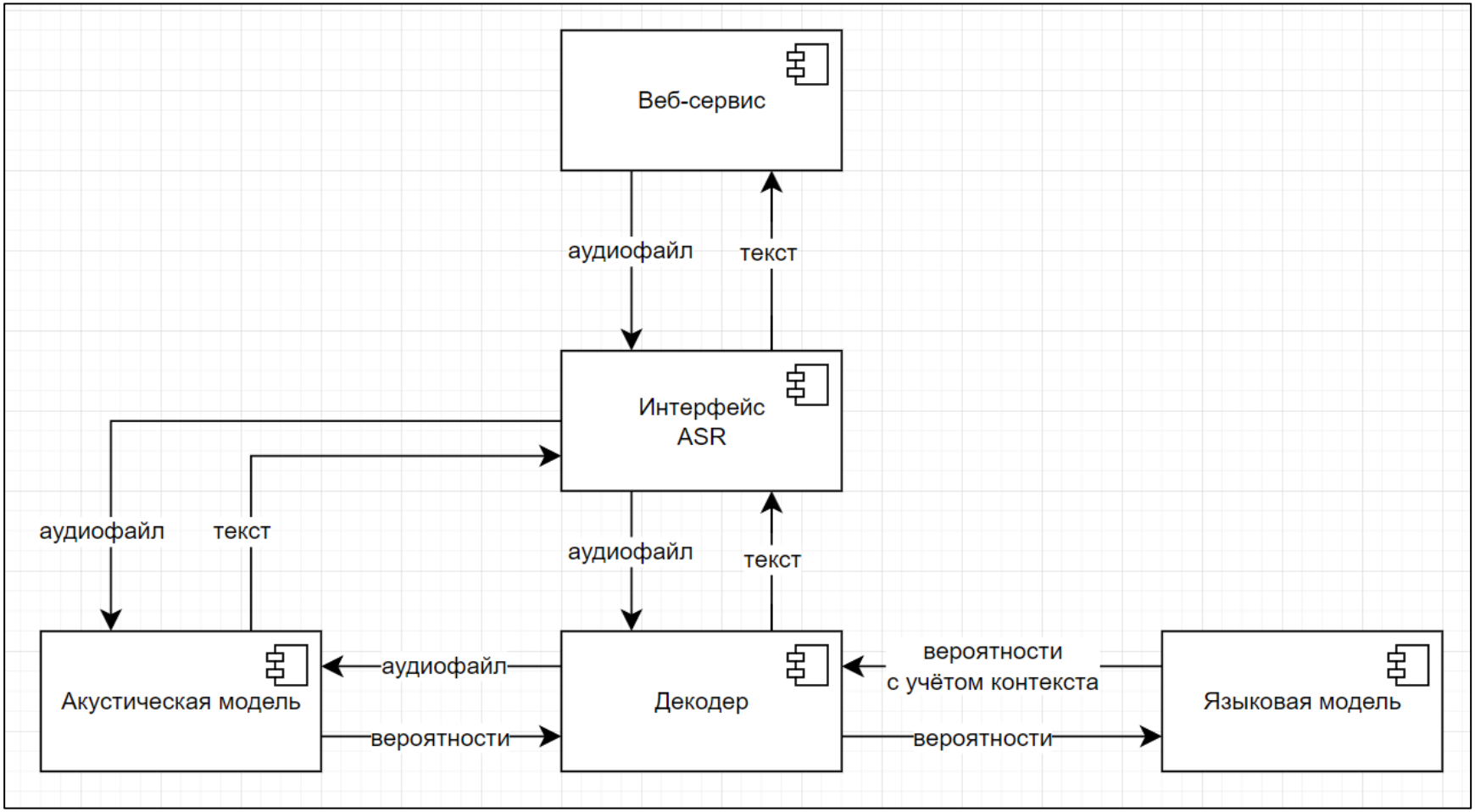
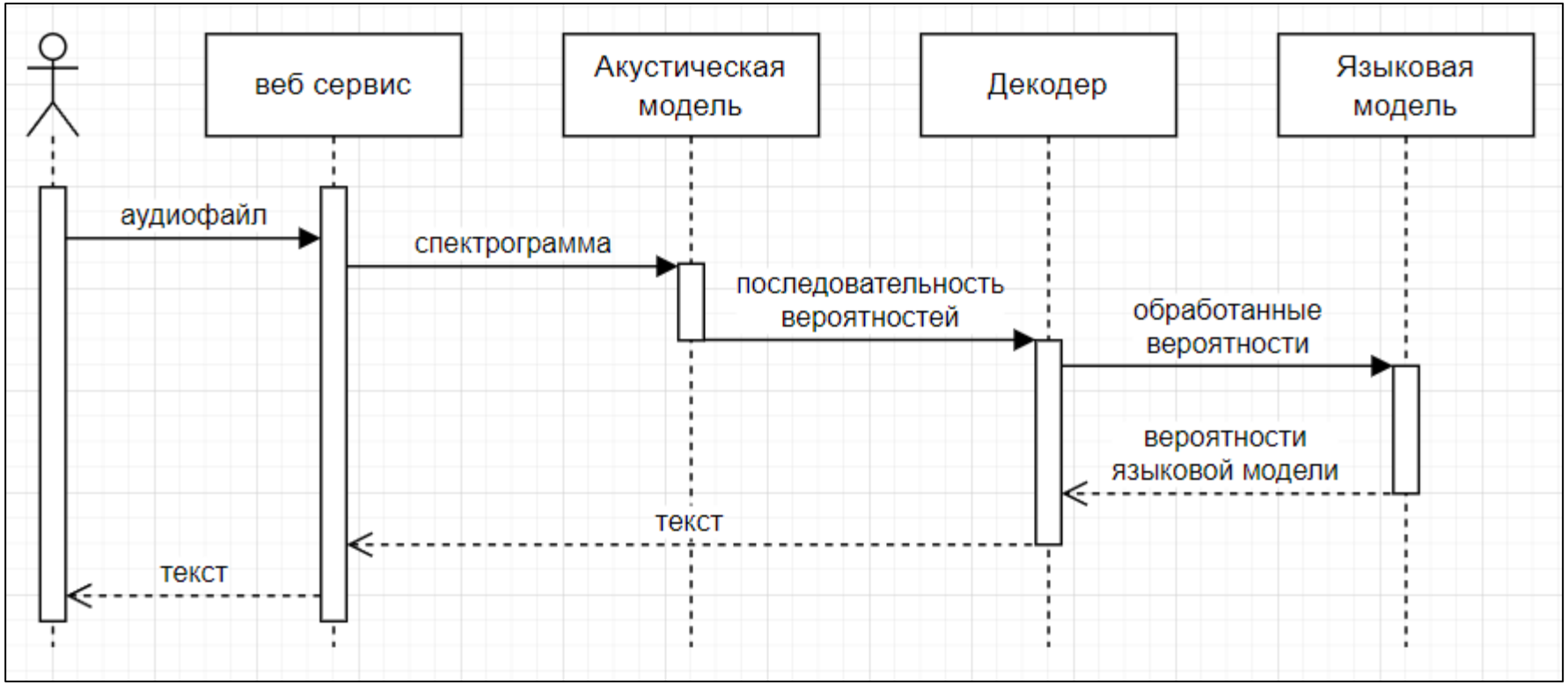


Диаграмма последовательности



Функциональные требования

- ASR модель должна уметь преобразовывать аудиофайл в текстовый формат
- Пользователь должен иметь возможность загружать свой аудиофайл
- У пользователя должна быть возможность взаимодействия с моделью через веб UI
- Система должна уметь конвертировать аудио в нужный формат

Архитектура акустической модели

Название	Основана	Особенности	Минусы
DeepSpeech	RNN	Простота реализации, хорошая работа с последовательными данными	Высокая вычислительная сложность, медленное обучение
Wave2Vec	CNN	Эффективное извлечение признаков из аудио, высокая точность при большом количестве данных	Требует больших объемов данных для обучения
ContextNet	CNN + RNN	Использование контекстной информации для улучшения качества распознавания речи	Сложность архитектуры, требует больше ресурсов для обучения
QuartzNet15x5	CNN	Компактная модель, низкие задержки, высокая скорость обучения благодаря сверточным сетям	Может быть менее точной при малом количестве данных

Акустическая модель

Отвечает за преобразование аудиоданных в последовательность вероятностей фонем

Для обучения использует библиотеку NeMo (PyTorch)

Вычислительные мощности:

- NVIDIA RTX 3060 12 ГБ
- Оперативная память 16 ГБ
- Intel(R) Core(TM) i5-12400f CPU @ 2.5GHz 4.5GHz

Время обучения модели ~90 часов с использованием обучения с переносом.

Языковая модель и декодер

Языковая модель помогает определить наиболее вероятные последовательности слов и фраз, учитывая контекст и грамматические правила языка.

Создана с помощью библиотеки KenLM в виде n-gram модели.

Для декодера используется библиотека CTC-decoders, которая использует алгоритм лучевого поиска.

Датасеты

1. Common Voice – открытый датасет голосовых записей, собранный Mozilla
    1. Количество голосов: 3206
    2. Количество часов 235

1. Golos – открытый русскоязычный датасет, разработанный компанией Sber
    1. Количество голосов: ~12000
    2. Количество часов: ~1240

Оценка модели

WER – Word Error Rate

$$WER = 100 * (I + S + D) / N$$

I – вставки лишних слов

S – замены слов на некорректные

D – пропуск слова

N – все слова исходной последовательности

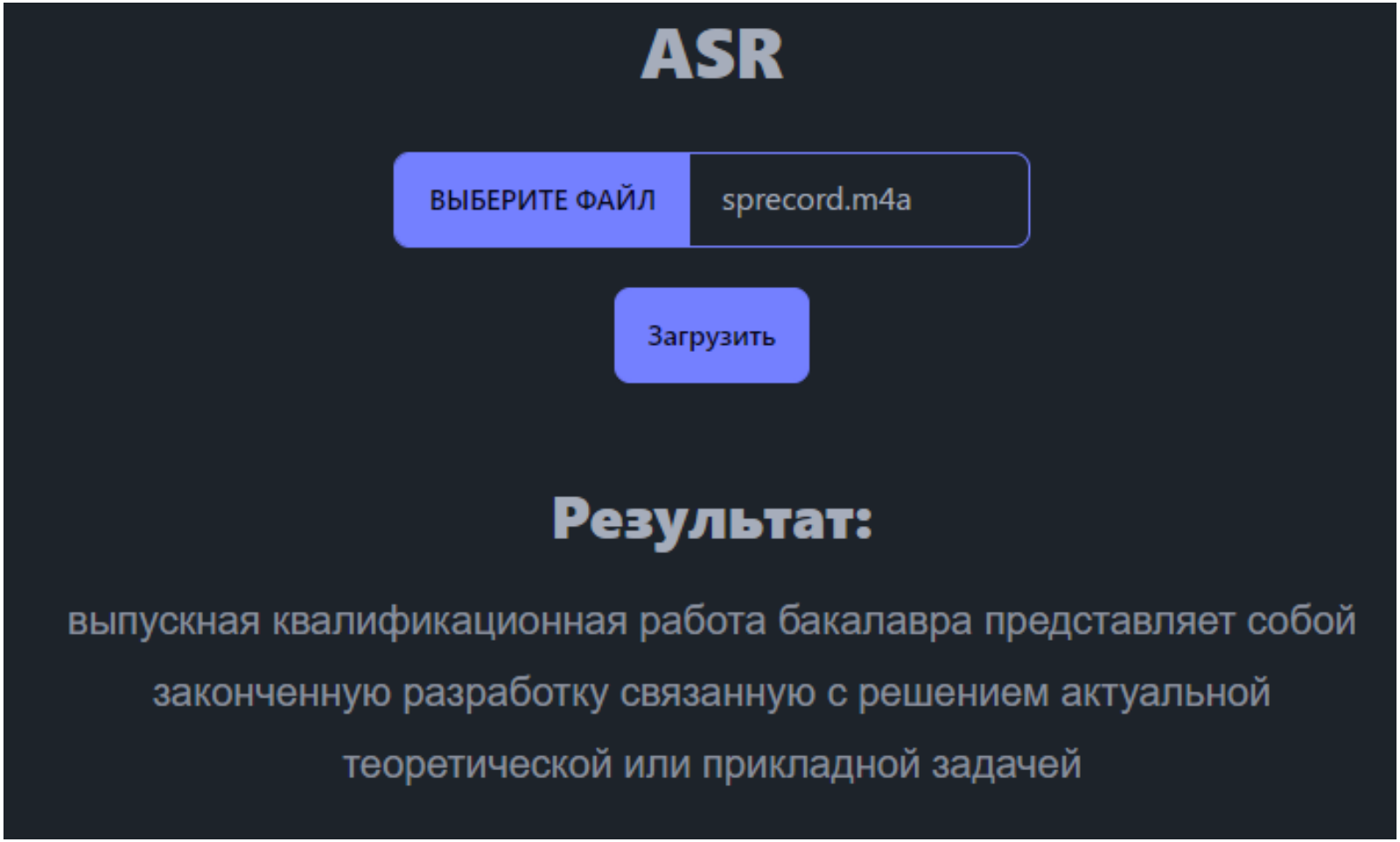
WER без языковой модели: 17%

WER с языковой моделью: 8,5%

WER базовой референсной модели Whisper: 6,7%

Ссылка на пример: https://youtu.be/EyXmczOYsOQ

Frontend


ASR
ВЫБЕРИТЕ ФАЙЛ
sprecord.m4a
Загрузить
Результат:
выпускная квалификационная работа бакалавра представляет собой законченную разработку связанную с решением актуальной теоретической или прикладной задачей

Backend

Есть POST запрос по пути «/asr/upload_audio»

На вход он принимает аудиофайл в любом формате.

По необходимости аудиофайл конвертируется в нужный формат через ffmpeg

Дальше аудиофайл передаётся в модель, и после работы полученный текст возвращает клиенту

Инструменты разработки

Клиент:

- Vite – инструмент сборки для проектов
- React – библиотека для создания UI
- Typescript – расширение JavaScript с поддержкой статической типизации
- Tailwind CSS – фреймворк для быстрой разработки интерфейсов

Сервер:

- Python – высокоуровневый ЯП общего назначения
- Flask – легковесный веб-фреймворк для Python

Основные результаты

1. Проанализирована предметная область
2. Выбраны технологии реализации
3. Обучена акустическая модель
4. Реализована языковая модель
5. Разработан веб-сервис для демонстрации модели