Московский государственный университет имени М.В. Ломоносова

Университет МГУ-ППИ в Шэньчжэне

Факультет вычислительной математики и кибернетики

Направление подготовки 01.03.02 «Прикладная математика и информатика»

Выпускная квалификационная работа

«Разработка мобильного приложения для изучения иностранного языка с использованием адаптивного алгоритма оценки речи»

Выполнил:

Воронцов Георгий Юрьевич

Научный руководитель:

к.т.н., доцент А.А. Демин

Шэньчжэнь

2024

Оглавление

Введение

Развитие международных отношений и ускорение темпов глобализации стоят в ряду приоритетных задач общества в 21 веке. Одним из основополагающих факторов в их решении является изучение иностранных языков. Изучение иностранных языков включает в себя не только изучение грамматики языка и оборотов письменной речи, но и совершенствование навыков устной речи, главным образом выражающихся в произношении. Наличие высокого уровня произношения необходимо, поскольку только при условии его наличия можно быть уверенным в полном понимании со стороны собеседника. Такой навык, как правило, может быть приобретен лишь посредством упорной работы и практики, и занятий со специалистом в области преподавании языка как иностранного.

Также, по данным исследования 2009 года, представленного в журнале “Language Learning”, нахождение в языковой среде и мотивация являются одними из основополагающих факторов эффективности изучения иностранного языка. Если человек, изучающий иностранный язык не имеет возможности приехать в страны носителей языка, то процесс изучения может замедлиться. Во избежание замедления необходимо уделять больше времени и сил занятиям и практике.

Главным предметом исследования в данной выпускной квалификационной работе и выступает произношение, а точнее индивидуальные его аспекты и ошибки речи, в ходе изучения иностранного языка.

Научная новизна представленной работы сводится к разработке и реализации алгоритма оценки речи людей, изучающих русский язык как иностранный. Адаптивный алгоритм оценки строится на выявлении ошибок речи с учетом особенностей голоса при помощи преобразования исходных звукового потока с дальнейшим анализом при помощи языковой модели.

Цель работы заключается в разработке мобильного приложения для изучения иностранного языка с использованием адаптивного алгоритма оценки речи. Данное приложение помогает частично автоматизировать процесс тренировки произношения. При помощи разработанного инструмента люди, нацеленные на изучение иностранного языка, смогут самостоятельно совершенствовать навыки устной речи, не прибегая к помощи со стороны преподавателя.

В данном приложении реализован контроль качества произношения русского языка. В целях повышения качества выносимой оценки, разработка ведется совместно с Центром русского языка Университета МГУ-ППИ в Шэньчжэне. Первыми пользователями приложения будут китайские студенты Университета МГУ-ППИ.

В рамках данной работы, в соответствии с целью, решаются следующие задачи:

- Разработка алгоритма для оценки речи
  - Исследование существующих подходов к анализу речи,
  - Реализация алгоритма выявления и визуализации ошибок произношения,
- Разработка мобильного приложения для платформы WeChat
  - Изучения средств разработки мини-приложений,
  - Разработка клиентской части мини-приложения,
  - Разработка серверной части.

В данном приложении реализован контроль качества произношения русского языка. В целях повышения качества выносимой оценки, разработка ведется совместно с Центром русского языка Университета МГУ-ППИ в Шэньчжэне. Первыми пользователями приложения будут китайские студенты Университета МГУ-ППИ.

Основная часть текста данной выпускной квалификационной работы делится на главы:

- Общий обзор аналогов, в которой содержится обзор существующих приложений-аналогов и анализ проведенных исследований в данной предметной области,
- Общий обзор, в которой содержится описание решений задачи оценки речи, рассмотренных в рамках выполнения данной работы,
- Проектная реализация, в которой содержится описание процесса разработки алгоритма оценки речи и мобильного приложения, также в этой главе содержится описание используемых сред разработки.

Глава 1. Общий обзор аналогов

1.1 Обзор приложения Duolingo

Duolingo - приложение для изучения иностранных языков. В нем представлено множество языков, доступных для изучения, включая русский, китайский, английский и другие языки. В приложении представлены полные курсы по языкам, позволяющие развивать навыки ученика в 4 направлениях:

- устная речь,
- чтение,
- восприятие информации на слух,
- письмо.

Приложение дает пользователю возможность подобрать оптимальную для него структуру курса, основываясь на тесте, состоящем из 4 вопросов:

- интересующий иностранный язык или языки,
- текущий уровень владения иностранным языком по субъективной оценке пользователя,
- цель изучения иностранного языка,
- предполагаемое время, затрачиваемое на изучение в день.

Также приложение позволяет оценить текущий уровень владения языком перед началом прохождения курса.

После всех вышеперечисленных подготовительных операций пользователь может приступить к прохождению курса. Сам курс разделен на тематические блоки с заданиями, направленными на развитие перечисленных выше навыков.

В рамках данного проекта, наибольший интерес представляет способ развития навыков говорения, представленный в приложении Duolingo, а конкретно - механика проверки контроля качества произношения. Рисунки 1-2 – скриншоты интерфейса задания на отработку произношения в Duolingo. Интерфейс обладает классической, для заданий данного типа, структурой:

- текст, который необходимо произнести,
- кнопка для проигрыша речи диктора
- кнопка начала и остановки записи,
- кнопка проверки записанной речи.

Рисунок 1. Интерфейс Duolingo

Рисунок 2. Интерфейс Duolingo

Также, в связи со спецификой самого приложения – оно предоставляет возможность развивать все ключевые навыки в ходе изучения иностранного языка, есть кнопка для пропуска задания при условии, что ученик не может говорить в момент прохождения курса.

Перейдем к рассмотрению техники проверки речи на предмет ошибок в произношении. Исходя из информации, представленной на официальном сайте сервиса, разработчики считают, что произношение не должно быть “идеально правильным”. Иностранный язык – средство коммуникации и передачи смыслов. Разработчики считают, что у ученика должен быть определенный уровень произношения, для этого сервис предусматривает множество упражнений на аудирование, а также уроки по самостоятельному развитию речевого аппарата. Следовательно, сервис не ставит перед собой задачи достижения пользователем высокого уровня произношения.

С технологической точки зрения, проверка произношения реализована с помощью технологии распознавания речи с использованием искусственного интеллекта.

Наименьшим элементом речи пользователя при использовании такой технологии является слово. Соответственно, оценка произношения сводится к ответу на вопрос: “произнесены ли все слова?”. Это также подтверждает цель команды разработчиков повысить качество речи пользователя до уровня, при котором другой человек сможет понять, что было сказано, но не более того. То есть, действительно, язык изучается как средство коммуникации.

В итоге, приложение Duolingo предоставляет пользователям возможность улучшения качества произношения путем выполнения заданий на аудирование и говорение, следования материалам и практикам от команды Duolingo. Однако, контроль качества произношения не строгий, а результат оценки недостаточно информативный.

1.2 Обзор приложения Hello Chinese

Hello Chinese - приложение для изучения исключительно китайского языка. Так же, как и приложение Duolingo направлено на всецелое развитие на пути изучения иностранного языка. То есть, в приложении присутствуют все четыре типа упражнений в тех или иных проявлениях.

Пользовательский путь в приложении начинается с вопроса об уровне знания китайского языка. Далее открывается доступ к контенту в виде тематических уроков, содержащих различные упражнения, в том числе и упражнение на развитие навыков устной речи.

Следует упомянуть, что во введении к курсу в приложении Hello Chinese присутствует краткое введение в фонетические аспекты языка (Рис. 3), что важно в начале изучения китайского языка.

В рамках данной работы, интерес представляют упражнения, направленные на развитие навыка устной речи. Разберем технику проверки произношения на примере из вводного урока. На рисунках 1 и 2 представлены скриншоты упражнения на произношение. Интерфейс, так же, как и в приложении, рассмотренном ранее, включает все те же базовые элементы:

- текст, который необходимо произнести,
- кнопка для проигрыша речи диктора
- кнопка начала и остановки записи,

- кнопка проверки записанной речи,
- кнопка пропуска задания.

Реализация данного типа заданий в Hello Chinese, имеет два преимущества перед реализацией в Duolingo. Первое – наличие латинской транслитерации (пиньинь). Однако, это обусловлено спецификой языка, иероглификой, так что абсолютным преимуществом назвать этот факт нельзя. Второе – более строгая проверка качества произношения, и это уже является значительным преимуществом. При произношении иероглифа с неправильным тоном, Hello Chinese отмечает этот факт и засчитывает голосовой ответ неверным. Реакцию приложения на неверный ответ можно увидеть на Рисунке 2.

Рисунок 3. Скриншот Hello Chinese

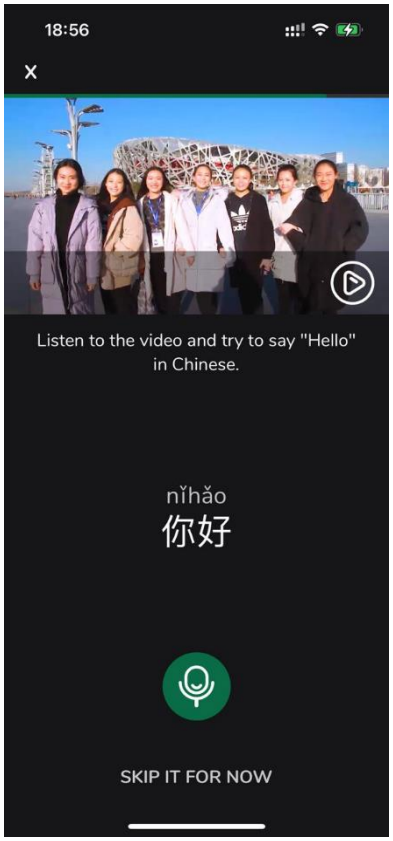

Рисунок 4. Скриншот интерфейса Hello Chinese

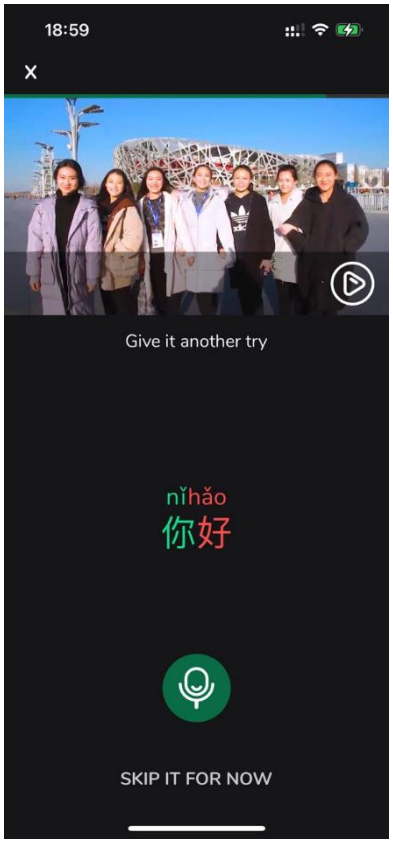

Рисунок 5. Скриншот интерфейса Hello Chinese

Несмотря на преимущества приложения в развитии навыка говорения, главной проблемой остается недостаточная информативность. Указано наличие ошибки при произношении определенного иероглифа, но не указано, в чем конкретно была ошибка.

В итоге, приложение развивает все 4 главных направления в изучении иностранного языка, но оценка качества произношения, все еще, не точна, а результат проверки не указывает на конкретную проблему.

1.3 Обзор приложения Babbel

Приложение Babbel – приложение для изучения множества языков. Среди них можно выбрать и русский язык. Основным языком в приложении является английский.

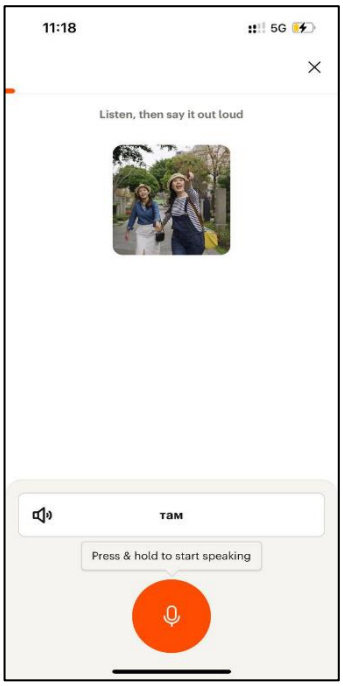

Рисунок 6. Скриншот интерфейса Babbel

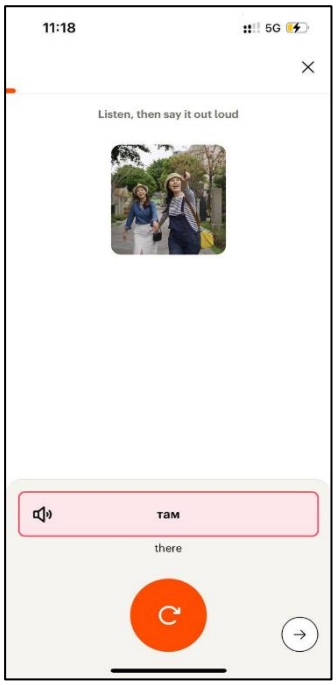

Рисунок 7. Скриншот интерфейса Babbel

Изучение русского языка в данном приложении начинается со знакомства с русским алфавитом. Приводятся базовые представления об алфавите и строится соответствие между некоторыми буквами латиницы и кириллицы. Далее через разбор слов по буквам, а фраз по словам происходит изучение фраз первой необходимости. Эти фразы состоят из букв, которые поставлены в соответствие с аналогами из латинского алфавита.

Развитие навыка говорения не является обязательным, однако упражнения на устную речь, все же, присутствуют в приложении (см. рис. 6-7). Интерфейс для данного упражнения содержит все необходимое для функционирования:

- Кнопка для прослушивания речи диктора,
- Кнопка для записи речи,
- Текст, который необходимо прочесть в рамках задания.

Проверка произношения реализована с помощью технологии распознавания речи, по аналогии с тем, как реализована проверка в приложении Duolingo. Такой способ проверки не является информативным. Причина была подробно раскрыта в пункте 1.1.

Касаемо развития других навыков, невозможно точно ответить о качестве сервиса, так как бесплатно предоставляется для изучения только один урок, состоящий из двух слов. Упражнения, действительно, направлены на развитие и письма, и говорения, и навыка аудирования, но невозможно оценить их объективно, так как бесплатная версия слишком малый список упражнений.

1.4 Сравнительная характеристика

Итак, перейдем к сравнению приложений-аналогов с приложением, разработанным в рамках данного проекта. Для удобства восприятия сравнительная характеристика представлена в виде таблицы, а выполнение критериев отмечено оценкой от 1 до 5. Оценка выполнения критериев субъективна.

Ниже приведены пояснения к таблице 1.

Точность определения места ошибки.

Как было упомянуто выше, Duolingo и Babbel используют технологию распознавания речи, которая направлена не на поиск ошибок в речи, а наоборот на подбор более вероятного слова по звукозаписи. Hello Chinese определяет локацию ошибки речи более точно - до отдельного иероглифа (может быть и словом, зависит от контекста), однако это обусловлено спецификой китайского языка. Оба приложения указывают на большую часть речи - слово, что снижает точность контроля качества произношения.

Разработанное приложение, напротив, направлено на определение местоположения ошибки с точностью до фонем, являющихся наименьшими единицами речи. Это позволяет значительно повысить точность определения местоположения ошибки в произнесенной фразе.

Определение характера ошибки.

Разработанное приложение выдает пользователю фонемный разбор его речи и речи диктора, исходя из их звукозаписей, соответствующих заданию, в виде выравнивания. Благодаря этому, пользователь может наглядно выяснить, в чем конкретно заключается ошибка: пропуск фонемы, замена фонемы или группы фонем, постановка ударения. В приложениях-аналогах такая возможность не предусмотрена.

Таблица 1. Сравнение приложений аналогов с текущим проектом

	Duolingo	Hello Chinese	Babbel	**Текущий проект**
Точность определения места ошибки	2	3	2	**5**
Определение характера ошибки	0	0	0	**5**
Наличие фонетического разбора дикторской звукозаписи	0	4	0	**5**
Наличие фонетического разбора пользовательской звукозаписи	0	3	0	**5**

Наличие фонетического разбора дикторской звукозаписи.

Также стоит учитывать, что тренировка произношения не равнозначна тренировке чтения, то есть не всегда произносится каждая буква и не всегда очевидно, на какой слог падает ударение. Эту проблему решает наличие фонетического представления. Duolingo и Babbel не демонстрируют пользователю такого представления. Hello Chinese выдает пользователю пиньинь для соответствующих иероглифов, на котором указаны тона, что подобно фонетическому разбору. Приложение, разработанное в рамках ВКР, предоставляет пользователю фонетический разбор дикторской звукозаписи, с помощью которой пользователь может более подробно разобрать фразу, приведенную в упражнении.

Наличие фонетического разбора пользовательской звукозаписи.

В процессе тренировки навыка говорения необходимо не только видеть каноническое произношение предложения в задании, но и свои ошибки, чтобы понимать, на что нужно обратить внимание при следующей попытке. В приложениях Duolingo и Babbel фонетический разбор отсутствует. В приложении Hello Chinese подобие такого разбора присутствует за счет наличия пиньиня. В разработанном приложении присутствуют фонетические разборы и дикторской звукозаписи и пользовательской. Также между ними построено выравнивание с разметкой ошибок, чтобы пользователь имел возможность сравнить свое произношение с каноническим.

Стоит упомянуть, что приложения Duolingo, Babbel и Hello Chinese – коммерческие проекты, их курсы с определенной ступени могут быть получены только на платной основе. Приложение, разрабатываемое в рамках данной выпускной квалификационной работы, является некоммерческим, весь процесс разработки стимулирован заинтересованностью в предмете исследования.

Приведем также несколько критериев, не имеющих прямое отношение к разработанному приложению.

Развитие других навыков.

Так как приложения-аналоги направлены на изучение иностранных языков в целом, подходят для пользователей с любым уровнем, они обладают широким спектром заданий. Задания направлены на развитие чтения, аудирования, письма и устной речи.

Данный проект является, в большей степени, приложением-помощником для развития произношения. В рамках данной выпускной квалификационной работы не стоит задача разработки приложения, позволяющее всецело изучить иностранный язык.

Возможность изучения разных языков.

Duolingo – всемирно известное приложение для изучения множества языков. Babbel также не малоизвестен, но менее популярен, чем Duolingo. Hello Chinese направлен на изучение исключительно китайского языка, об этом можно судить по названию приложения. Приложение, разработанное в рамках данного проекта, позволяет повысить уровень произношения только для русского языка. Однако, используемая технология дает возможность расширить список доступных языков. Подробнее об алгоритме будет рассказано во второй главе.

Глава 2. Общий обзор технологий

2.1 Подходы к решению задачи вынесения оценки речи

Поскольку, главным образом, задача вынесения оценки речи решается с помощью сравнения двух звукозаписей: дикторской и пользовательской, в рамках данной выпускной квалификационной работы, были рассмотрены два подхода к решению задачи вынесения оценки произношения: подход через разбор звукозаписи по фонемам с дальнейшим анализом с использованием алгоритмов обработки текстовой информации и подход через вычисление мел-частотных кепстральных коэффициентов (Mel-frequency cepstral coefficients), МЧКК (MFCC).

Оба подхода можно условно разделить на два этапа:

- Первый этап - преобразование звуковой информации в текстовую или числовую,
- Второй этап - дальнейшая обработка полученных данных с целью вынесения оценки.

Первый подход реализован с помощью нейросети Wav2Vec2Phoneme, которая переводит звуковую информацию в последовательность фонем, что по сути является текстовым представлением. После преобразования происходит поиск наиболее оптимального выравнивания этих последовательностей для оценки произношения с помощью алгоритма Нидлмана-Вунша.

Второй подход основан на использовании технологии вычисления МЧКК, обладающих полезными свойствами для задачи оценки произношения, такими, как отсечение частот, не содержащих человеческий голос. После получения наборов МЧКК происходит, как и в первом подходе, поиск оптимального выравнивания, однако уже с помощью другого алгоритма – DTW (Data Time Warping).

Несмотря на то, что оба алгоритма осуществляют поиск наиболее оптимального выравнивания, они различаются, так как входными данными для алгоритма Нидлмана-Вунша являются символьные последовательности, а для алгоритма DTW – числовые последовательности. Поскольку не у всех символов есть численные эквиваленты, которые можно было бы использовать для поиска выравнивания, реализация алгоритмов различается, но идейно они похожи.

Следует отметить, что, помимо двух упомянутых выше подходов, также в начале исследования поставленной задачи оценки произношения экспериментально были исследованы способы сравнения звукозаписей, по-отдельности использовавших прямое вычитание амплитуд и алгоритм DTW для поиска оптимального выравнивания временных рядов. Эти способы показали неудовлетворительные результаты, так как оценка схожести дикторского и пользовательского файла явно не соответствовала действительности.

2.2 Подход через фонемный разбор

2.1.1 Нейронная сеть Wav2Vec2Phoneme

Для подробного рассмотрения и описания нейросети была взята статья от разработчиков Wav2Vec2Phoneme. Дальнейшая информация изложена в соответствии с источником.

Для распознавания речи в рассматриваемом проекте используется метод нулевого переноса для кросс-языкового распознавания фонем на основе модели wav2vec 2.0, предварительно обученной на множестве языков. Эта модель позволяет распознавать фонемы ранее не рассматривавшихся языков.

Методология включает два ключевых компонента: предварительное самообучение модели и сопоставление фонем. Предварительное самообучение использует предобученную модель wav2vec 2.0 (XLSR-53), обученную на данных из 53 языков. Эта модель включает сверточный кодировщик признаков, который преобразует сырые звуковые данные в латентные речевые представления, и трансформер, который выводит контекстные представления. Кодировщик работает по схеме BERT, в рамках которой на этапе обучения скрытые представления маскируются, а задача модели заключается в правильной идентификации маскированных латентных представлений.

Для отображения фонем используются символы международного фонетического алфавита (IPA). Поскольку словарь целевого языка может не включать все необходимые фонемы, применяются артикуляционные признаки. Для каждой фонемы вычисляется расстояние по Хэммингу между артикуляционными векторами признаков, и создаются

два типа отображений: из фонем обучающего языка в целевой и наоборот. На практике используется первое, если не указано иное.

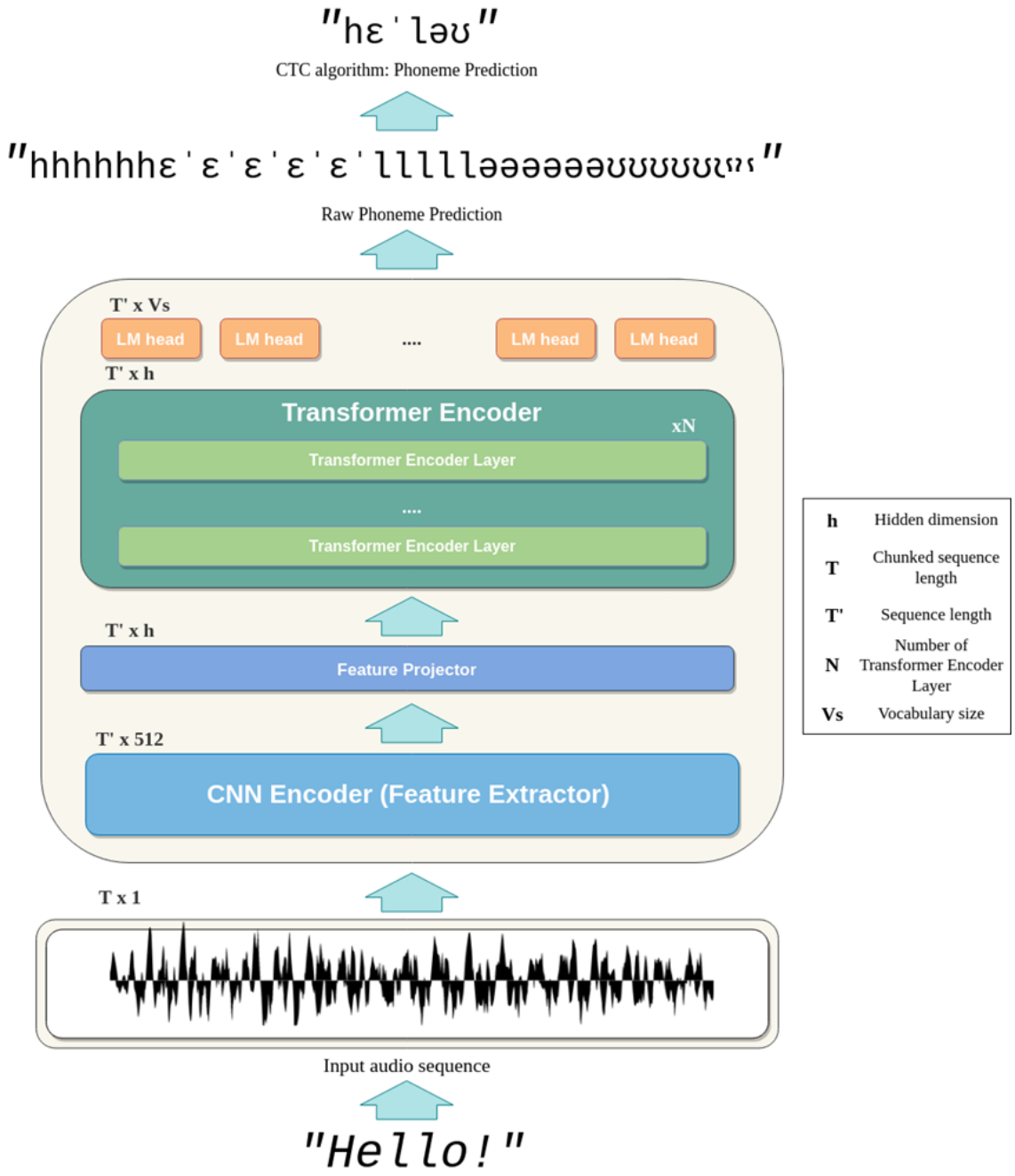

Рисунок 8. Структура нейронной сети

Экспериментальная настройка включает три многоязычных корпуса: CommonVoice, BABEL и Multilingual LibriSpeech (MLS). Эти корпусы содержат данные на множестве языков. Транскрипции нормализуются путем удаления знаков препинания и редких символов, а затем из текстовых транскрипций получают фонемные аннотации с использованием инструментов ESpeak и Phonetisaurus.

Модель обучается на данных, помеченных фонемами обучающих языков, с использованием Connectionist Temporal Classification (CTC). На этапе дообучения веса кодировщика признаков не обновляются, тогда как веса трансформера обновляются. Для генерации окончательных транскрипций используется декодер с поиском по лучу wav2letter.

Результаты исследования показали, что метод нулевого переноса сопоставим по точности с другими методами, но использует более простую схему. Метод, использованный в данной работе, превосходит по эффективности предыдущие работы по нулевому переносу, использующие только извлечение признаков из модели wav2vec 2.0, предварительно обученной на одном языке. Показано, что предварительное обучение на множестве языков значительно улучшает точность. Различные стратегии построения лексикона и фонемизации также влияют на производительность.

Данный подход, основанный на нулевом переносе и дообучении предварительно обученной модели, превосходит предыдущие работы, не использующие помеченные данные родственных языков. Модель способна транскрибировать одновременно несколько языков, что делает возможным преобразование мультиязыковой звукозаписи в фонемную последовательность.

2.1.2 Алгоритм поиска оптимального выравнивания Нидлмана-Вунша

Данный алгоритм представляет собой инструмент для нахождения наиболее оптимального выравнивания двух последовательностей. В большей степени применяется для нахождения выравниваний в биоинформатике между двумя заданными генетическими последовательностями. Однако, его также можно адаптировать под задачу выравнивания фонемных последовательностей с целью вынесения оценки произношения.

Алгоритм Нидлмана-Вунша принимает несколько параметров:

- две последовательности символов, для которых строится выравнивание,
- три параметра, отвечающие за совпадение символов (match), несовпадение (mismatch) и пробел (gap).

Эти три параметра представлены в виде чисел как поощрение и наказание за результат сравнения двух символов. В случае сравнения двух фонемных последовательностей всем трем параметрам присвоена единица.

Итак, алгоритм включает в себя три ступени:

- Ступень инициализации:
  - создается матрица F (двумерный массив) размера $m \times n$, где m – количество символов в первой последовательности, n – количество символов во второй последовательности;
  - первый столбец матрицы F заполняется значениями $[0, -\text{gap}, -2 \times \text{gap}, \dots, -m \times \text{gap}]$;
  - первая строка матрицы F заполняется по аналогии первому столбцу;
  - создается матрица P (двумерный массив) размера $m \times n$, содержащая указатели;
  - первый столбец матрицы P заполняется значениями, соответствующими указателю вверх;
  - первая строка матрицы P заполняется значениями, соответствующими указателю влево.
- Ступень заполнения:
  - для каждой пары символов из последовательностей x и y вычисляются три возможные оценки:
    - Диагональная оценка: если символы совпадают, прибавляется значение mismatch, иначе оно вычитается.
    - Горизонтальная оценка: присваивается значение из ячейки слева и вычитается значение gap.
    - Вертикальная оценка: присваивается значение из ячейки сверху и вычитается значение gap.
  - Вычисляется максимальная из трех оценок и записывается в текущую ячейку массива F.
  - Указатель в текущей ячейке массива P обновляется в зависимости от того, какая из трех оценок была максимальной.

- Обратное следование:
  - Обратное следование начинается с самой нижней правой ячейки массива F и продолжается до тех пор, пока не будет достигнута верхняя левая ячейка.
  - В зависимости от значения указателя в текущей ячейке массива P выбирается направление движения (диагонально, вверх или влево).
  - Символы из последовательностей x и y добавляются к результирующим выровненным последовательностям rx и ry. Если указатель указывает на пробел, в соответствующую последовательность добавляется символ пробела.

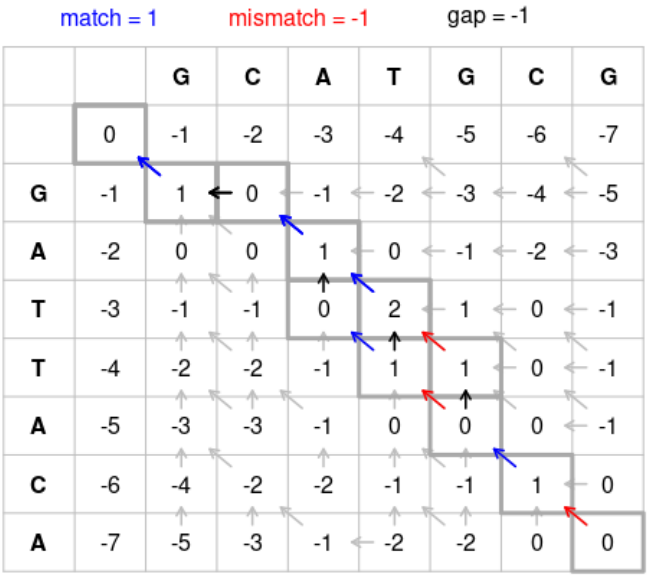

Рисунок 9. Матрица соответствия

Пример работы алгоритма указан на изображении ниже. На нем изображена матрица, включающая в себя оценки и указатели, то есть можно рассматривать представленную матрицу как слияние матриц F и P. На изображении также выделена траектория построения выравнивания. Наиболее оптимальным выравниванием является выравнивание, обладающее наибольшей оценкой. Общая оценка выравнивания – сумма значений, соответствующим направляющим стрелкам в матрице. В данном случае, сумма по траектории равна 1.

2.3 Подход через вычисление частотных коэффициентов

2.3.1 Алгоритм вычисления частотных коэффициентов

Рисунок 10. Алгоритм вычисления

Алгоритм вычисления коэффициентов детерминирован, и может быть расписан пошагово.

Разбиение на фреймы.

Перед выполнением последующих преобразований звуковой файл разбивается на короткие отрезки, называемые фреймами. В зависимости от требуемой точности и постановки задачи длительность фреймов может варьироваться. Наиболее популярным выбором считаются фреймы длительностью 20 - 40 миллисекунд. В связи с тем, что в рамках поставленной задачи объем информации ограничен, для повышения точности было принято решение использовать разбиение на отрезки длиной в 20 миллисекунд. Во избежание некорректного разделения речевого потока разбиение производится с наложением. Наиболее часто встречающимся вариантом длительности наложения является половина фрейма.

Наложение оконной функции Хэмминга.

Окно Хэмминга - функция, используемая для минимизации помех на краях фреймов. Значения фрейма и оконной функции перемножаются. Если окно определено как $W_n(m)$, где $0 \leq m \leq N_m - 1$ и N_m - количество значений в отдельно взятом фрейме, то выходной сигнал после применения оконной функции к сигналу вычисляется следующим образом: $Y(m) = X(m) * W_n(m)$, где $0 \leq m \leq N_m - 1$, и $Y(m)$ представляет собой выходной сигнал после умножения входного сигнала $X(m)$ на окно Хэмминга $W_n(m)$. Существует множество оконных функций, таких как прямоугольное

окно, окно с плоским верхом и окно Хэмминга, но обычно окно Хэмминга представляется следующим образом:

$$W_n(m) = 0.54 - 0.46\cos\frac{2\pi m}{N_m - 1}, \quad 0 \le m \le N_m - 1 \qquad (1)$$

Преобразование Фурье.

При обработке звуковой информации бывает рассмотреть полезно иное представление волны. Изначально данные представляют собой функцию от времени, значениями которой является амплитуда звукового сигнала. Преобразование Фурье позволяет анализировать сигнал в частотной области, результатом процесса преобразования является спектр сигнала. То есть если построить график получившегося дискретного представления функции, то по оси абсцисс будут отложены значения частот, а по оси ординат – значения амплитуд.

Есть множество алгоритмов реализации данного преобразования, при обработке звуковых сигналов чаще всего применяют Дискретное Преобразование Фурье (Discrete Fourier Transform) и Быстрое Преобразование Фурье (Fast Fourier Transform). Эти два алгоритма принципиально ничем не отличаются, то есть результаты их работы совпадают. Тем не менее, с точки зрения временной сложности алгоритма различие прослеживается. Классический алгоритм дискретного преобразования оценивается как $O(N^2)$, в то время как быстрая реализация оценивается как $O(N \log N)$. Такой результат достигнут за счет использования метода «разделяй и властвуй».

Переход на Мел шкалу.

Для выявления особенностей звуковой информации на первоначальном этапе часто используют Мел-частотный кепстр волны. Он является представлением короткого спектра мощности звука, основанным на преобразовании Фурье в нелинейной частотной мел шкале. Мел - единица высоты звука, соответствующая определенной единице на шкале Герца. На основе статистических методов выведена логарифмическая зависимость Герцов от Мелов (см. рис. 7), описанная О'Шонесси в 1987 году.

$$m_f = 2595 \lg\left(\frac{f}{700} + 1\right) \qquad (2)$$

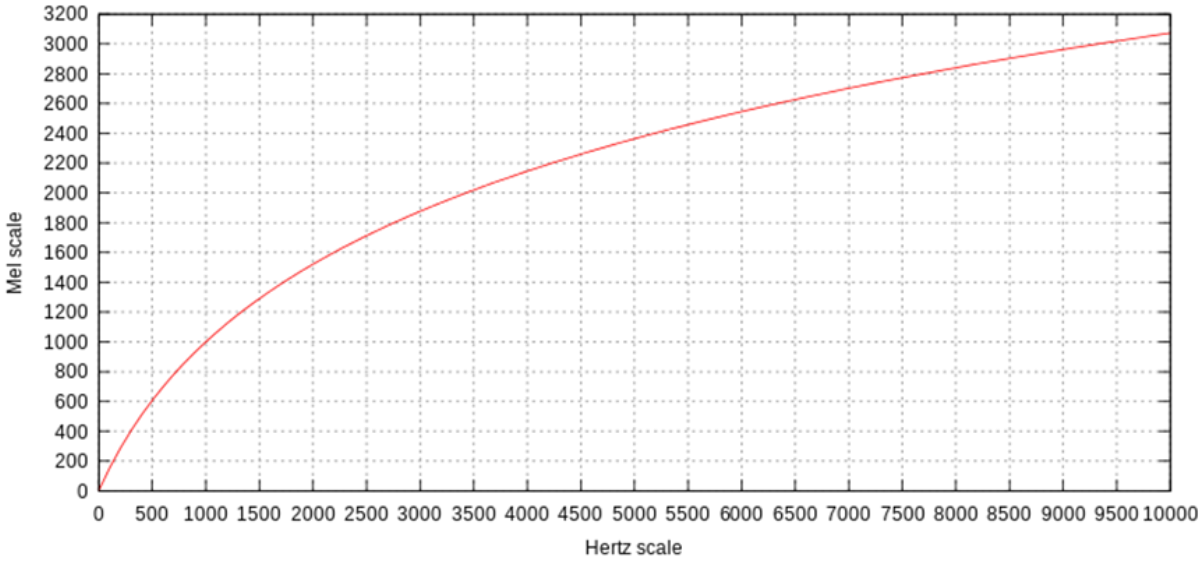

Рисунок 11. Мел шкала

Дискретное косинусное преобразование.

На данном этапе происходит обратный переход из частотной области во временную. Такая операция выполняется при помощи дискретного косинусного преобразования:

$$C_n = \sum_{k-1}^{k} (\log D_k) \cos\left[m\left(k - \frac{1}{2}\right)\frac{\pi}{k}\right], \text{ где } m = 0,1, \dots, k-1 \quad (3)$$

Коэффициенты, высчитываемые по этой формуле, и являются Мел-частотными кепстральными коэффициентами.

2.3.2 Алгоритм поиска оптимального выравнивания временных рядов

Алгоритм DTW (Dynamic Time Warping) основан на методах динамического программирования. Этот алгоритм предназначен для измерения схожести между двумя временными рядами, которые могут различаться по времени или скорости. Также эта техника используется для нахождения оптимального выравнивания между двумя временными рядами, если один из них может быть поставлен в соответствие с другим нелинейным образом путем растягивания или сжатия вдоль своей временной оси. Это соответствие между двумя временными рядами затем можно использовать для поиска соответствующих областей между двумя временными рядами или для определения схожести между ними. На рисунке 12 показан пример того, как один временной ряд ставится в соответствие с другим.

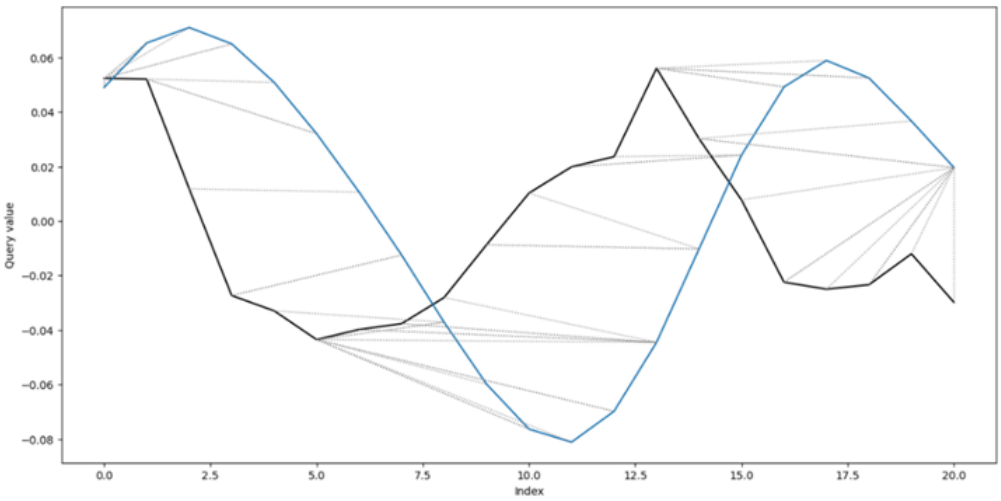

Рисунок 12. Визуализация оптимального выравнивания

В каждой вертикальной линии на рисунке точка в одном временном ряду соединяется с соответствующей похожей точкой в другом временном ряду. Линии имеют сходные значения по оси y, но разделены, чтобы вертикальные линии между ними можно было рассмотреть более легко. Если бы оба временных ряда на рисунке были идентичными, все линии были бы прямыми вертикальными линиями, потому что не требовалось бы никакого искажения для "выравнивания" двух временных рядов. Расстояние по пути искажения – это мера различия между двумя временными рядами после их искажения, которое измеряется суммой расстояний между каждой парой точек, соединенных вертикальными линиями на рисунке. Таким образом, два временных ряда, идентичные за исключением локализованного растягивания временной оси, будут иметь расстояния DTW равные нулю. Принцип DTW заключается в сравнении двух динамических шаблонов и измерении их схожести путем вычисления минимального расстояния между ними.

Классический способ вычисления DTW:

Предположим, у нас есть два временных ряда Q и C, длиной n и m соответственно, где:

$$Q = q_1, q_2, \dots, q_i, \dots, q_n \quad (4)$$

$$C = c_1, c_2, \dots, c_j, \dots, c_m \quad (5)$$

Для выравнивания двух последовательностей с использованием DTW создается матрица размером n на m, где элемент (i, j) матрицы содержит расстояние $d(q_i, c_j)$ между

двумя точкам q_i и c_j . Затем абсолютное расстояние между значениями двух последовательностей вычисляется с использованием вычисления евклидова расстояния:

$$d(q_i, c_j) = (q_i - c_j)^2 \quad (6)$$

Каждый элемент матрицы (i, j) соответствует выравниванию между точками q_i и c_j. Затем накопленное расстояние измеряется как:

$$D(i, j) = \min[D(i-1, j-1), \quad D(i-1, j), \quad D(i, j-1)] + d(i, j) \quad (7)$$

Это показано на рисунке 13, где горизонтальная ось представляет время входного сигнала, а вертикальная ось представляет временную последовательность эталонного шаблона. Показанный путь дает минимальное расстояние между входным и эталонным сигналами. Заштрихованная область представляет собой область поиска для функции отображения времени в шаблонное время. Любой монотонно неубывающий путь в этом пространстве является альтернативой, которую стоит рассмотреть. С использованием методов динамического программирования поиск пути минимального расстояния можно выполнить за полиномиальное время P(t) с использованием уравнения ниже:

$$P(t) = O(NV^2) \quad (8)$$

где N - длина последовательности, а V - количество шаблонов, которые следует рассмотреть.

Итак, данный алгоритм применим при необходимости сравнения двух звукозаписей, так как он решает проблемы, перечисленные в предыдущем блоке, а именно:

- разная длина звукозаписей,
- разное время фактического произношения звуков.

Результатом работы алгоритма является матрица расстояний, по которой высчитывается оптимальный путь выравнивания звукозаписей. В идеальном случае совпадения звукозаписей оптимальный путь выглядит, как линейная функция, и чем более искривлен график, тем менее похожи две звукозаписи. Выглядит такой путь выравнивания, на примере сравнения записей диктора и студента, так:

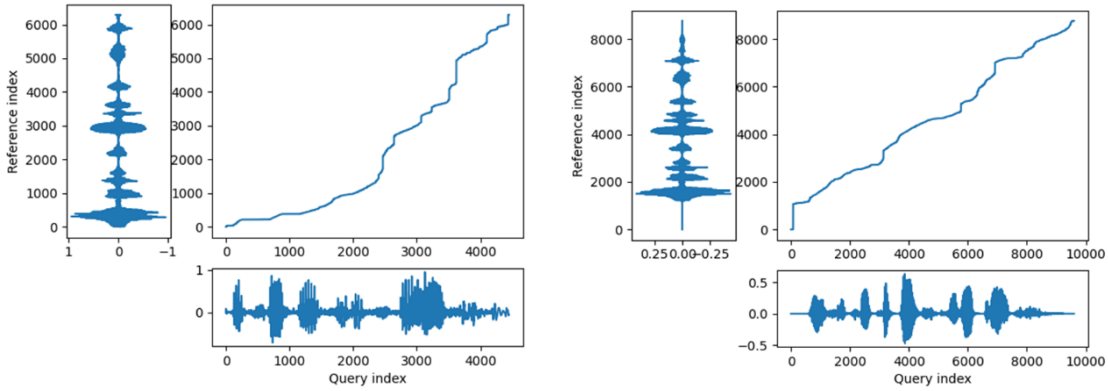

Рисунок 13. Визуализация работы DTW

Однако, главный минус двух вышеописанных методов состоит в том, что они не учитывают, какой конкретно звук или буква была произнесена. Причиной тому служит то, что само по себе значение амплитуды звука в конкретный момент времени неинформативно. Вдобавок к этому данные часто зашумлены, что делает показатели амплитуды еще более неинформативными. Также следует заметить, что скорость сравнения сырых звуковых дорожек существенно зависит от частоты дискретизации.

2.2 Общий обзор подхода к разработке

В рамках данной выпускной работы было разработано мобильное приложение для проверки качества произношения. В течение первых недель исследования поставленной задачи был также рассмотрен вопрос о выборе платформы для разработки приложения. Главным образом, рассматривались два направления: мобильные приложения для IOS и Android и веб-приложения для мобильных устройств. Поскольку, согласно статистике за ближайшую декаду лет, мобильные приложения на операционных системах IOS и Android составляют более 99% рынка, другие операционные системы не рассматривались.

Если рассматривать вариант разработки мобильных приложений с точки зрения общедоступности для пользователей двух вышеупомянутых операционных систем, то

целесообразной становится разработка для обоих вариантов. Зачастую это можно осуществить тремя путями:

- Создание отдельных нативных приложений для Android и IOS,
- Создание прогрессивного веб-приложения,
- Одно кроссплатформенное приложение для двух систем.

Нативное решение предполагает разработку приложения на языке программирования, который создан или который принимается платформой. Для операционной системы на базе Android в роли родных языков выступают Java и Kotlin. Для платформы IOS чаще всего разработка ведется на специально созданном Swift. С одной стороны, такое решение обеспечивает приложению доступ к полному спектру функционала данной системы. С другой стороны, в рамках такого подхода придется создать два приложения, чтобы охватить оба типа пользователей, что потребует значительно больше времени и усилий.

Прогрессивное веб-приложение – технология в веб-разработке, преобразующая сайт в мобильное приложение. В России данная технология получила особое распространение после вступления в силу ограничений для мобильных приложений российских банков. Ярким примером реализации такого решения является веб-приложение от Тинькофф банка. Приложение не требует скачивания и может быть установлено на прямую из браузера. Однако, данный подход также обладает своим недостатком – отсутствие доступа к большей части функционала, в том числе и к микрофону, который необходим для осуществления оценки произношения.

Решение задачи разработки приложения через создание одного кроссплатформенного приложения обретает популярность в последнее время. Кроссплатформенность – это способность программного обеспечения работать на нескольких платформах. Такой подход позволяет исключить недостаток прогрессивного веб-приложения с доступом к функционалу устройства, также не требует удвоения нагрузки на разработчика, так как в этом случае необходимо разработать одно приложение для обеих систем. Существует несколько инструментов для реализации такого решения, самым популярным на данный момент является фреймворк Flutter.

Несмотря на преимущества кроссплатформенного подхода, как и у остальных решений, есть и недостатки. Из числа самых значительных недостатков: большой вес

приложения по итогу разработки, отсутствие возможности встраивания нативных модулей из-за их отсутствия в стандартном наборе.

В результате обзора решений для разработки был выбран подход, совмещающий в себе преимущества каждого из перечисленных. Приложение было разработано на платформе WeChat Mini Programs. WeChat Mini Programs (далее мини приложения) – платформа для разработки и публикации приложений от самого популярного в Китае мессенджера WeChat. По статистике на 2023 год, WeChat насчитывает более 1.3 миллиардов пользователей, более 60% которых пользуются мини приложениями.

Преимущества такого подхода перечислены ниже:

- Полный доступ к функционалу системы, поскольку мини приложения работают в рамках мессенджера;
- Отсутствие необходимости скачивания приложения на устройство, так как мини приложения реализованы в виде веб-приложений, хранящих только кэш в памяти пользовательского устройства, которые при необходимости можно удалить;
- Кроссплатформенность – мини приложение работает в рамках мессенджера, приложение для которого уже доступно для устройств с любой операционной системой;
- Большое сообщество разработчиков.

Есть и недостатки в такой реализации, а именно тот факт, что скорость работы приложения зависит от скорости интернет-соединения пользователя. Также в некотором роде субъективным недостатком является отсутствие полного объема документации на русском или английском языке.

Подводя итог вышесказанному, подходом для решения задачи разработки мобильного приложения в рамках данной выпускной квалификационной работы был выбран подход с использованием платформы для разработки мини приложений WeChat Mini Programs.

Глава 3. Практическая реализация. Мобильное приложение

3.1 Обзор сред разработки приложения

3.1.1 Среда разработки серверной составляющей решения

Алгоритм, реализованный для оценки произношения, и веб-сервер был разработан в среде программирования PyCharm Community Edition версии 2023.2.1. Такой выбор обусловлен удобством данной среды и многолетнего опыта работы в ней.

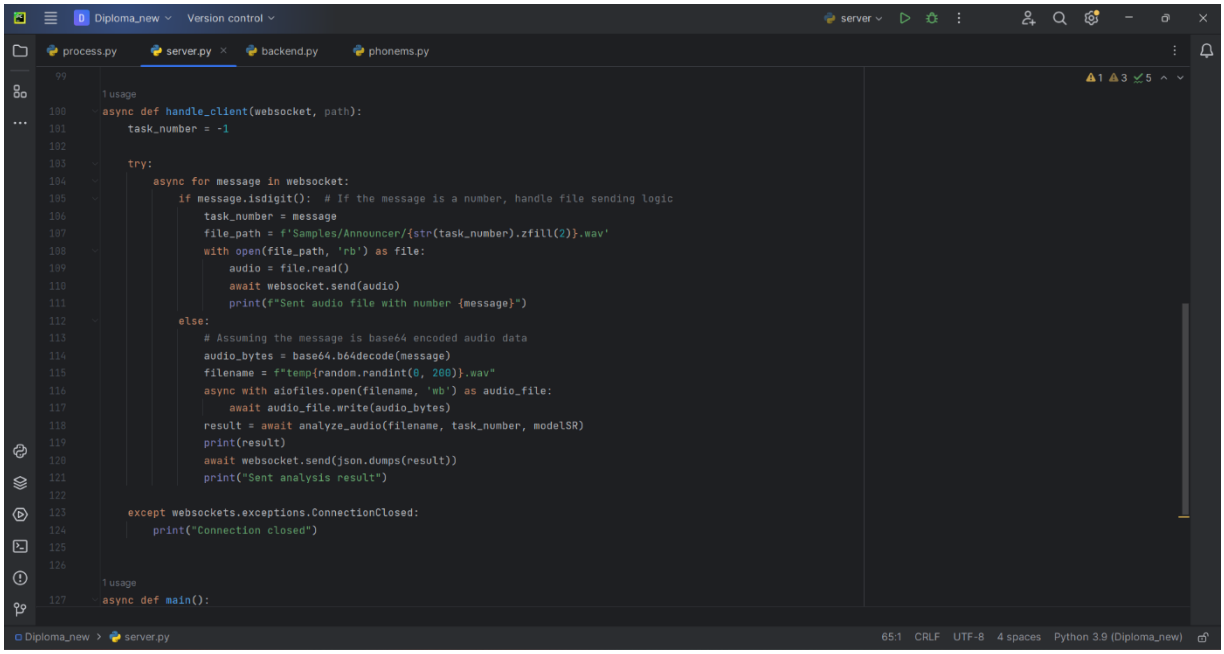

Рисунок 14. Среда разработки PyCharm

Данная среда разработки обладает несколькими преимуществами:

- удобная система импортирования пакетов,
- автоматическое создание виртуальное окружения,
- интуитивно понятный интерфейс.

Следует заметить, что среда PyCharm, как можно судить из названия, разработана специально для создания проектов на языке программирования Python.

Отдельно хотелось бы отметить систему локального контроля версий. Высокая частота автоматического сохранения версий программы позволяет отследить прогресс разработки и при необходимости вернуться к предыдущей версии программы (см. рис. 15).

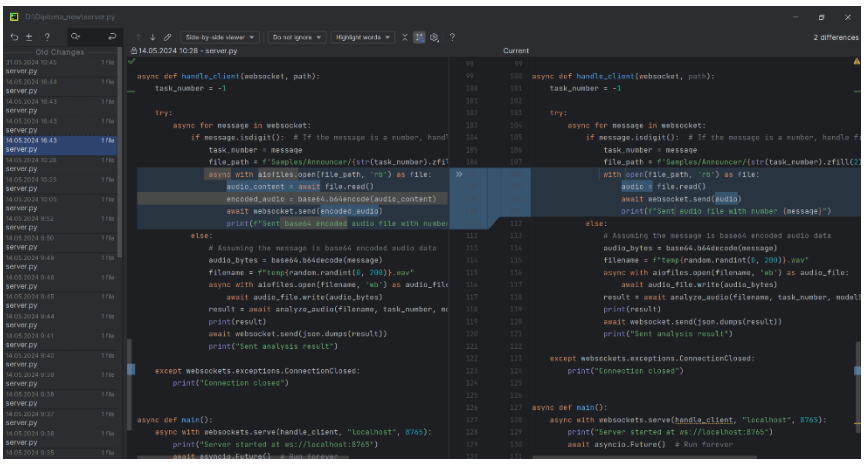

Рисунок 15. Система контроля версий в PyCharm

Распространение данной версии среды программирования PyCharm Community Edition ведется на некоммерческой основе, в то же время, она обладает всем необходимым функционалом для разработки широкого спектра программного обеспечения, что также является преимуществом. Безусловно, в бесплатной версии присутствуют определенные ограничения, но на рабочий процесс в рамках ВКР это не повлияло.

3.1.2 Среда разработки клиентской составляющей решения

Поскольку платформой для размещения данного мобильного приложения служит WeChat, для разработки приложения использовался специально созданный для разработки мини приложений инструмент – WeChat Mini Program Developing Tool (微信小程序开发工具).

Инструмент имеет классический для подобного рода сред интерфейсом (см. рис. 16), но, в связи со спецификой, присутствуют особенности. Одной из самых заметных особенностей выступает демонстрация экрана мобильного устройства в течение разработки. Такое решение оказывается очень полезным, так как изменения, внесенные в интерфейс, практически моментально отражаются на экране мобильного устройства. Также существует широкий выбор устройств с разными диагоналями и габаритами, что также полезно при адаптивной верстке.

Также приложение предусматривает автоматическое создание виртуальной среды и оболочки для приложения, так как она является стандартной в большинстве случаев, что многократно ускоряет процесс разработки на первых этапах.

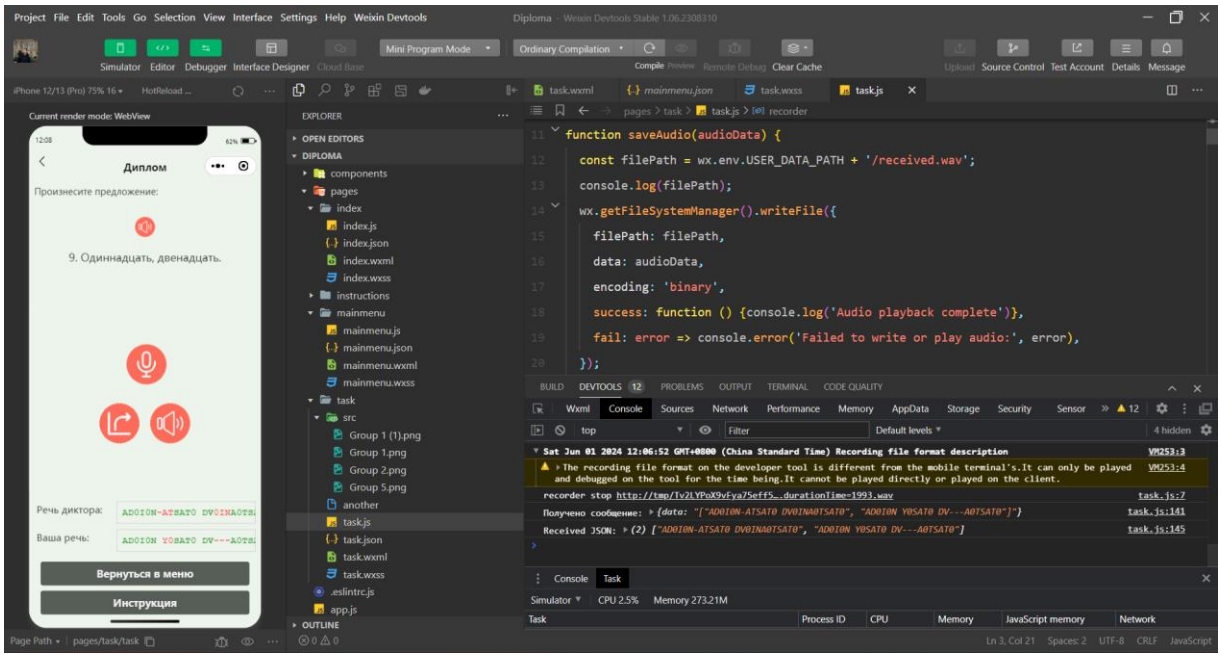

Рисунок 16. Интерфейс среды программирования от WeChat

В целом, данный инструмент оказался удобным и сложностей, связанных с его работой, в процессе выполнения задач в рамках ВКР встречено не было.

3.2 Разработка серверной части и настройка соединения через веб-сокет

3.2.1 Алгоритм оценки речи

Алгоритм оценки произношения состоит из нескольких этапов. В первую очередь алгоритм принимает пользовательскую запись от клиентской части мобильного приложения и подает ее на обработку нейросети для выделения фонемной последовательности.

Далее необходимо сравнить полученную фонемную последовательность с заранее известной фонемной последовательностью соответствующей дикторской звукозаписи. Сравнение осуществляется с помощью алгоритма Нидлмана-Вунша путем поиска наиболее оптимального выравнивания двух фонемных последовательностей. В данном случае параметры для алгоритма были выбраны следующие:

- match = 1;
- mismatch = -1;
- gap = -1.

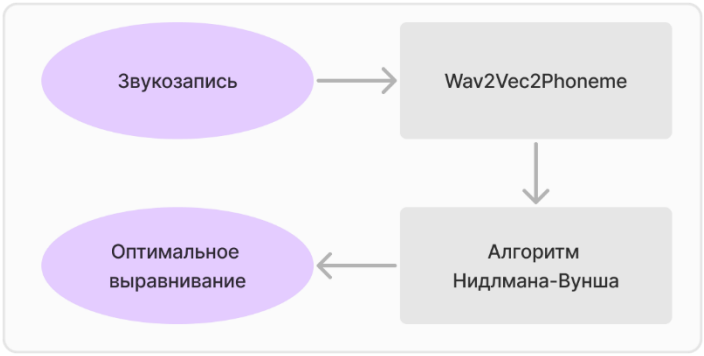

Рисунок 17. Блок-схема алгоритма анализа звукозаписи

Поскольку ошибкой является как неправильно произнесенная фонема, так и пропуск, штраф одинаковый. Подробная оценка будет вынесена на основе полученного выравнивания. Для расчета данной оценки выведены группы похожих по звучанию фонем. Это сделано с целью уточнения оценки, например, если человек произнес вместо “О” “А”, то ошибка не столь критична и итоговый балл будет снижен на меньшее количество пунктов, чем если вместо “О” сказано “Е”.

Полученный результат, а то есть полученное выравнивание и общая оценка, отправляется на клиентскую часть приложения.

3.2.2 Настройка взаимодействия с клиентской частью

Как было упомянуто в заголовке, взаимодействие происходит через веб-сокет соединение. Веб-сокет (WebSocket) – протокол связи поверх протокола TCP, обеспечивающий обмен сообщениями между веб-сервером и клиентской частью. Изначально технология использовалась для обмена сообщениями между веб-сервером и браузером, но сейчас существует ряд решений для адаптации технологий под различных участников такого обмена. В рамках данной работы используется WebSocket API для языка Python.

Получаемое сообщение может содержать данные только двух типов:

- Число, обозначающее номер задания, к выполнению которого приступил пользователь;

- Бинарный массив данных, то есть звукозапись пользователя, которую необходимо проанализировать на предмет ошибок речи.

Отправляемое сообщение, соответственно, может также содержать данные двух типов:

- Бинарный массив данных – дикторская звукозапись, соответствующая номеру задания;
- Результат анализа звукозаписи, а именно две строки фонемных представлений и оценка.

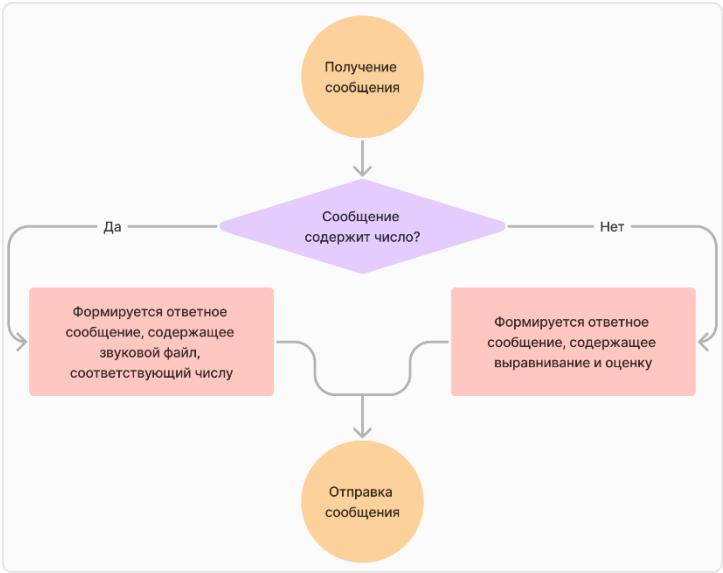

Рисунок 18. Блок схема обработки сообщений

Следует отметить, что на самом деле, данные отправляются и получаются той и другой стороной в формате JSON, то есть в строковом формате, а далее в зависимости от содержимого приводятся к конкретным типам, а именно к бинарным данным, целым числам или остаются в строковом представлении.

В Приложении 1 к ВКР представлен листинг кода, который реализует серверную часть приложения.

3.3 Разработка клиентской части

3.3.1 Общее описание системы страниц

Приложение имеет простую структуру из четырех страниц. Главный акцент был сделан на разработку функционала оценки произношения, так как задачи сделать сложную систему не стояло. Однако, разработанное приложение обладает всем необходимым функционалом для проверки качества произношения.

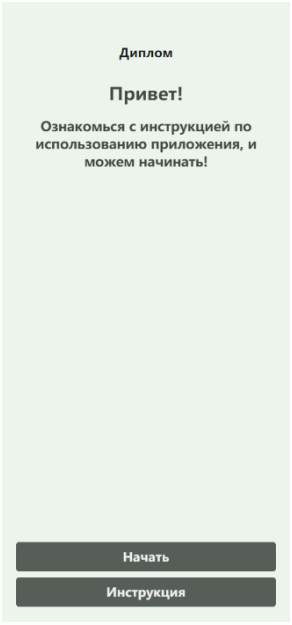

Рисунок 19. Главный экран

Рисунок 20. Инструкция

Рисунок 21. Меню выбора упражнения

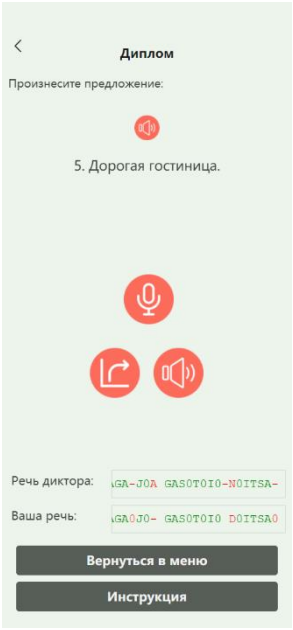

Рисунок 22. Экран упражнения

На рисунках 19-22 можно увидеть все страницы, разработанные для приложения. При первом рассмотрении сразу можно заметить пастельную цветовую гамму. Решение не делать дизайн приложения слишком ярким было принято из соображений комфорта пользователя. Поскольку предполагается, что пользователь может проводить достаточно длительное время в приложении, тренируя навыки произношения, целесообразно не использовать слишком яркие оттенки для снижения нагрузки на глаза. Все интерактивные элементы выделены темными цветами, элементы на странице с заданием выделены более ярко.

Каждая страница содержит свой набор интерактивных элементов. На страницах главного меню и инструкции (см. рис. 19-20)пользователь имеет возможность только прочесть содержимое, вернуться на предыдущую страницу или продвинуться далее. На странице с выбором задания (см. рис. 21) пользователь может выбрать задание или

вернуться в главное меню. Более детального рассмотрения заслуживает страница задания (упражнения). Рассмотрим ее более детально.

Для удобства рассмотрения рисунок 23 размечен на смысловые блоки красными прямоугольниками. Рассмотрим каждый блок по-отдельности:

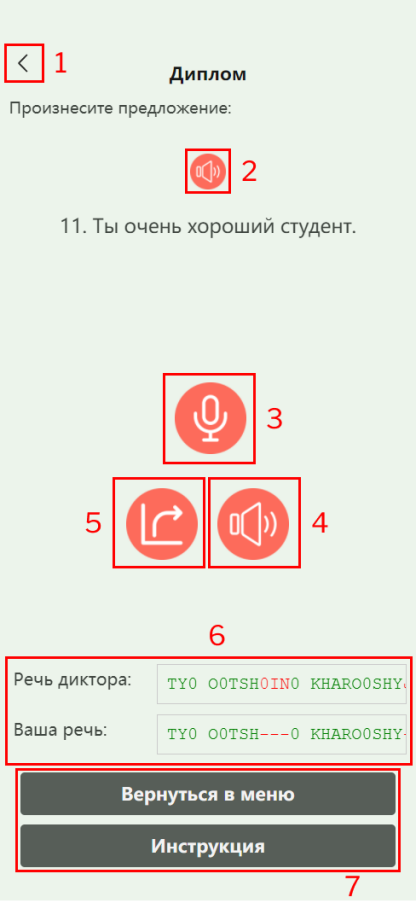

Рисунок 23. Экран задания (пояснение)

1. Кнопка возврата на предыдущую страницу;
2. Кнопка для воспроизведения дикторской звукозаписи, согласно тексту задания, указанному строчкой ниже (см. рис. 23);
3. Кнопка записи голоса пользователя;
4. Кнопка воспроизведения голоса пользователя;
5. Кнопка отправки на сервер для оценки произношения;
6. Блок для отображения фонемных представлений звукозаписей диктора и пользователя;
7. Стандартные кнопки для возврата в меню и перехода к странице с инструкцией.

Фонемные представления отображаются не полностью, так как они обладают слишком большой длиной. Чтобы справиться с данной проблемой, блок под номером 6 реализован с возможностью горизонтальной прокрутки. Также следует отметить, что

шрифт, используемый для отображения фонемных последовательностей, моноширинный, что позволяет беспрепятствввенно сравнивать дикторскую речь с речью пользователя.

3.3.2 Описание технических аспектов разработки

Разработка клиентской части мини приложения напоминает процесс разработки веб-приложения или веб-сайта. Мини приложение, по сути, и является веб-сайтом для мобильных устройств с определенными преимуществами, за счёт того, что оно открывается внутри мессенджера. Клиентская часть мини приложения написана с помощью языка программирования JavaScript, языка гипертекстовой разметки WXML и языка каскадных таблиц стилей WXSS. Язык JavaScript – самый популярный и широко используемый язык для написания логики работы веб-приложений и веб-сайтов. WXML является аналогом популярного языка гипертекстовой разметки HTML, созданным для разработки мини приложений. WXSS также является аналогом более популярного CSS.

В ходе разработки и выполнения задач, поставленных в рамках дипломной работы, различий между аналогами от WeChat и оригинальными языками, которые могли бы замедлить процесс не было. Наоборот, в данные языки уже встроены некоторые элементы, которые можно использовать непосредственно «из коробки». Например, в набор элементов для WXML уже встроен элемент, отвечающий за навигационную панель, что позволяет не тратить время на самостоятельное описание.

Однако, несмотря на преимущества данных языков с точки зрения разработки мини приложений, есть и недостатки. Один из которых не позволил в еще большей степени ускорить процесс – это отсутствие возможности импортирования технологий извне. То есть при разработке можно использовать только предоставленный пакет функций, а нестандартные решения приходится прописывать вручную.

Также недостатком инструмента в целом является недоработанная файловая система. Прописывание в логике программы доступа к локальному файлу невозможно – система отвечает, что не имеет права доступа. Это ограничение обходится с помощью указания вместо пути к файлу URL-ссылки на него.

Перейдем к описанию клиентской части приложения с технической точки зрения. Для упрощения восприятия на рисунке 24 приведена схема возможных переходов пользователя между страницами. Эти переходы также осуществлены благодаря встроенной функции-аналоге перехода между страницами веб-сайта.

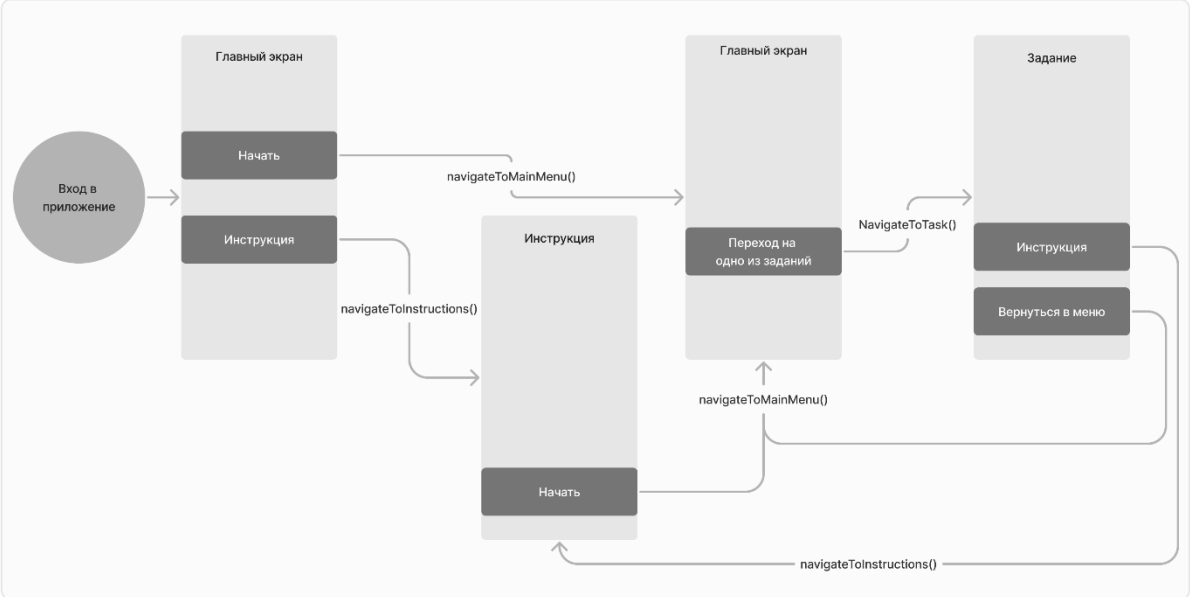

Рисунок 24. Схема переходов пользователя

Как и в предыдущем пункте, обратим внимание на последнюю страницу с заданием (см. рис. 23), так как на этой странице реализована часть более сложного функционала.

Начнем с перехода на страницу с заданием из главного меню. При загрузке страницы с заданием строится туннель WebSocket соединения с сервером для дальнейшего обмена сообщениями. Первым сообщением клиент отправляет серверу номер задания, выполнение которого начал пользователь. Ответом на данное сообщение является массив бинарных данных, который передается в формате JSON по туннелю. При показе страницы данная звукозапись уже доступна к воспроизведению.

Далее предполагается, что пользователь записывает то, как он произносит фразу, показанную на экране (см. рис. 23). Звукозапись сохраняется, и пользователь имеет возможность ее прослушать и отправить на сервер для анализа на предмет ошибок.

После отправки на сервер и анализа пользователю возвращаются фонемные последовательности, появляющиеся внизу экрана по завершении всех операций по распаковке и разметке ошибок этих последовательностях.

При желании улучшить свой результат, пользователь имеет возможность повторно записать произношение не обновляя страницу с заданием.

Полный код логической составляющей страницы с заданием представлен в Приложении 2 к данной ВКР.

Заключение

В рамках данной выпускной квалификационной работы было разработано приложение для изучения русского языка для иностранцев. Акцент в изучении был сделан на качество произношения. Приложение помогает обнаружить неверно произнесенные звуки через фонетическое представление звукозаписи пользователя с размеченными ошибками. Таким образом, приложение частично автоматизирует процесс развития навыков произношения.

Для решения задачи оценки речи было рассмотрено несколько различных способов, и было выбрано сочетание двух технологий: нейронной сети, переводящей звукозапись в фонетическую последовательность, и алгоритма Нидлмана-Вунша для поиска оптимального выравнивания двух последовательностей. Также следует упомянуть, что алгоритм Нидлмана-Вунша был использован нестандартным образом, так как, в основном, для анализа геномных последовательностей.

Также планируется продолжить работу над приложением после завершения процедур, связанных с дипломной работой. Планируется

- сделать приложение более интерактивным и информативным, чтобы обеспечить максимально возможный темп развития навыков говорения пользователей;
- ввести статистику выполнения заданий, как индивидуальную, так и общую;
- выложить приложение в общий доступ;
- довести до конца исследование метода оценки речи с использованием частотных коэффициентов.

Не исключается заимствование некоторых элементов и идей у приложений аналогов, давно присутствующих на рынке для повышения общего качества данного приложения.

Список литературы

[1] Большакова Е. И., Баева Н.В. Написание и оформление учебно-научных текстов (курсовых, выпускных, дипломных работ). Составление презентаций: Учебно-методическое пособие. - М.: Издательский отдел факультета ВМиК МГУ имени М.В. Ломоносова

[2] Vibha Tiwari, MFCC and its applications in speaker recognition // International Journal on Emerging Technologies. - 2010. 19-22. - ... Helmer Strik, Khiet Truong, Febe de Wet, Catia Cucchiarini, Comparing classifiers for pronunciation error detection // INTERSPEECH. - 2007

[3] Shikha Gupta, Jafreezal Jaafar, Wan Fatimah wan Ahmad, Arpit Bansal, Feature extraction using MFCC // Signal & Image Processing: An International Journal (SIPIJ) Vol.4, No.4.

[4] Chadawan Ittichaichareon, Siwat Suksri and Thaweesak Yingthawornsuk, Speech Recognition using MFCC // International Conference on Computer Graphics, Simulation and Modeling. - 2012. 07.28

[5] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi, Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques // JOURNAL OF COMPUTING. - 2010. 03

[6] Qiantong Xu, Alexei Baevski, Michael Auli, SIMPLE AND EFFECTIVE ZERO-SHOT CROSS-LINGUAL PHONEME RECOGNITION, 2021

Приложение 1. Листинг кода веб-сервера

```
import asyncio
import aiofiles
import websockets
import base64
import json
import random
import numpy as np
from huggingsound import SpeechRecognitionModel
def needleman_wunsch(x, y, match=1, mismatch=1, gap=1):
  nx = len(x)
  ny = len(y)
  F = np.zeros((nx + 1, ny + 1))
  F[:, 0] = np.linspace(0, -nx * gap, nx + 1)
  F[0, :] = np.linspace(0, -ny * gap, ny + 1)
  P = np.zeros((nx + 1, ny + 1))
  P[:, 0] = 3
  P[0, :] = 4
  t = np.zeros(3)
  for i in range(nx):
    for j in range(ny):
      if x[i] == y[j]:
        t[0] = F[i, j] + match
      else:
        t[0] = F[i, j] - mismatch
      t[1] = F[i, j + 1] - gap
      t[2] = F[i + 1, j] - gap
      tmax = np.max(t)
      F[i + 1, j + 1] = tmax
      if t[0] == tmax:
        P[i + 1, j + 1] += 2
```

```
            if t[1] == tmax:
                P[i + 1, j + 1] += 3
            if t[2] == tmax:
                P[i + 1, j + 1] += 4
    i = nx
    j = ny
    rx = []
    ry = []
    while i > 0 or j > 0:
        if P[i, j] in [2, 5, 6, 9]:
            rx.append(x[i - 1])
            ry.append(y[j - 1])
            i -= 1
            j -= 1
        elif P[i, j] in [3, 5, 7, 9]:
            rx.append(x[i - 1])
            ry.append('-')
            i -= 1
        elif P[i, j] in [4, 6, 7, 9]:
            rx.append('-')
            ry.append(y[j - 1])
            j -= 1
    rx = ''.join(rx)[::-1]
    ry = ''.join(ry)[::-1]
    return rx, ry
def get_alignment(audio_path1, audio_path2, model):
    transcription = [e['transcription'] for e in model.transcribe([audio_path1, audio_path2])]
    return needleman_wunsch(transcription[0], transcription[1])
def wrong_words(dictor, student):
    res = []
    miss_spaces = []
    match = True
```

```
    for i in range(len(dictor)):
        if dictor[i] != ' ':
            if match:
                match = dictor[i] == student[i]
        else:
            if student[i] == '-':
                miss_spaces.append(len(res))
            res.append(match)
            match = True
    res.append(match)
    print(res)
    return res, miss_spaces
def load_model():
    return SpeechRecognitionModel("./snu-nia-12/wav2vec-large-xlsr-53_nia12_phone-nsu-ai-_russian")
async def analyze_audio(filename, task_number, model):
    return get_alignment(f"Samples/Announcer/{str(task_number).zfill(2)}.wav", filename, model)
modelSR = load_model()
async def handle_client(websocket, path):
    task_number = -1
    try:
        async for message in websocket:
            if message.isdigit():
                task_number = message
                file_path = f'Samples/Announcer/{str(task_number).zfill(2)}.wav'
                with open(file_path, 'rb') as file:
                    audio = file.read()
                    await websocket.send(audio)
                    print(f"Sent audio file with number {message}")
            else:
                # Assuming the message is base64 encoded audio data
                audio_bytes = base64.b64decode(message)
                filename = f"temp{random.randint(0, 200)}.wav"
```

```
            async with aiofiles.open(filename, 'wb') as audio_file:
                await audio_file.write(audio_bytes)
            result = await analyze_audio(filename, task_number, modelSR)
            print(result)
            await websocket.send(json.dumps(result))
            print("Sent analysis result")
    except websockets.exceptions.ConnectionClosed:
        print("Connection closed")
async def main():
    async with websockets.serve(handle_client, "localhost", 8765):
        print("Server started at ws://localhost:8765")
        await asyncio.Future()
if __name__ == "__main__":
    asyncio.run(main())
```

Приложение 2. Листинг кода мобильного приложения с заданием на JavaScript

```
const recorder = wx.getRecorderManager();
var tempFilePath = "";
recorder.onStop((res) => {
  console.log('recorder stop', res.tempFilePath);
  tempFilePath = res.tempFilePath;
})
function saveAudio(audioData) {
  const filePath = wx.env.USER_DATA_PATH + '/received.wav';
  console.log(filePath);
  wx.getFileSystemManager().writeFile({
    filePath: filePath,
    data: audioData,
    encoding: 'binary',
    success: function () {console.log('Audio playback complete')},
    fail: error => console.error('Failed to write or play audio:', error),
  });
}
Page({
  data: {
    content: "",
    id: "",
    result1: "",
    result2: "",
    startbutton: "Вернуться в меню",
    instructions: "Инструкция",
    isRecording: false,
    filePath: "",
```

```
},
 navigateToMainMenu() {
   wx.navigateBack();
},
 navigateToInstructions() {
  wx.navigateTo({
    url: '/pages/instructions/instructions',
  })
},
updateUI(text) {
  this.setData({
    content: text
  })
},
play_self(){
  wx.playBackgroundAudio({
    dataUrl: tempFilePath,
  })
  console.log("record is played");
},
record() {
  if (!this.isRecording) {
    recorder.start({
      format: "wav"
    });
    this.isRecording = true;
    console.log("recording started")
  }
  else {
    recorder.stop({})
```

```
    this.isRecording = false;
  }
},
playRecording() {
  wx.playBackgroundAudio({
    dataUrl: "http://usr/received.wav",
  })
},
startRecording(){
  recorder.start({
    format: "wav"
  });
  this.isRecording = true;
  console.log("recording started")
},
endRecording() {recorder.stop({})},
send(){
  wx.showLoading({
    title: 'Анализ',
  })
  wx.getFileSystemManager().readFile({
    filePath: tempFilePath,
    encoding: "base64",
    success(res){
      wx.sendSocketMessage({
        data: res.data,
      })
    },
    fail(res){ console.error(res) }
  })
```

```
},
onLoad(options) {
  this.setData({
    id:wx.getStorageSync('task_id'),
    content:wx.getStorageSync('task_text'),
  })
  wx.removeStorageSync('task_id');
  wx.removeStorageSync('task_text');
},
onReady() {
  var that = this;
  wx.sendSocketMessage({
    data: this.data.id,
  });
  console.log('Сообщение отправлено', this.data.id);
  wx.onSocketOpen(function() {
    console.log('WebSocket соединение установлено');
  });
  wx.onSocketMessage(function(data) {
    if (typeof data.data === 'string') {
      try {
        const json = JSON.parse(data.data);
        const string1 = json[0];
        const string2 = json[1];
        if (string1.length !== string2.length) {
          wx.showToast({
            title: 'Strings must be of the same length!',
            icon: 'none'
          });
          return;
```

```
      }
      const result1 = [];
      for (let i = 0; i < string1.length; i++) {
        result1.push({
          char: string1[i],
          match: string1[i] === string2[i]
        });
      }
      const result2 = [];
      for (let i = 0; i < string2.length; i++) {
        result2.push({
          char: string2[i],
          match: string1[i] === string2[i]
        });
      }
      that.setData({
        result1: result1,
        result2: result2
      });
      wx.hideLoading();
    } catch (error) {
      console.error('Error parsing JSON:', error);
    }
  } else {
    // Assume binary data is a WAV file
    console.log('Received WAV file');
    saveAudio(data.data);
  }
});
wx.onSocketError(function(error) {
```

```
    console.log('Ошибка WebSocket:', error);
  });
  wx.onSocketClose(function() {
    console.log('WebSocket соединение закрыто');
  });
 }
})
```