**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**
**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

УДК 004.05

СОГЛАСОВАНО

Научный руководитель,
приглашенный преподаватель

_____ А. Аланов
«_____» _____ 2022 г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия»,
профессор департамента программной
инженерии, канд. техн. наук

_____ В. В. Шилов
«_____» _____ 2022 г.

**Выпускная квалификационная работа**
(академическая)

на тему: **Исследование методов обучения без учителя для автоматического распознавания речи и синтеза речи по тексту**

по направлению подготовки 09.03.04 «Программная инженерия»

ВЫПОЛНИЛ

студент группы БПИ185
образовательной программы
09.03.04 «Программная инженерия»

_____ Б. Т. Малашенко
«_____» _____ 2022 г.

Москва 2022

# Реферат

Обучение моделей автоматического распознавания речи и синтеза речи по тексту часто требует сбора большого количества дорогостоящих транскрибированных данных. Для решения этой проблемы было предложено множество методов обучения без учителя. В данной работе рассмотрено несколько методов, используемых в настоящее время в этой области, основанных на двойственной сущности рассматриваемых задач. Последовательно обучая ASR и TTS модели, используя для обучения лишь небольшое число парных данных, мы можем получить качество, сравнимое с лучшими моделями при обучении с учителем. Все эксперименты проводятся при помощи языка Python и фреймворка Neural Modules, что позволяет построить полный конвейер обучения модели.

Данная работа состоит из 29 страниц, 3 глав, 8 рисунков, 5 таблиц. Использовано 30 источников.

**Ключевые слова:** автоматическое распознавание речи; синтез речи по тексту; обучение без учителя.

# Abstract

Training models for automatic speech recognition and speech synthesis often requires a huge corpus of expensive transcribed data. Many unsupervised learning methods have been proposed to solve this problem. This paper discusses several methods currently in use in the field, based on the dual nature of the tasks observed. By sequentially training ASR and TTS models, using only a small number of paired data for training, we can obtain quality comparable to the best supervised models. All experiments are performed using the Python language and the Neural Modules framework, which allows us to build a complete learning pipeline for the model.

The paper contains 29 pages, 3 chapters, 8 figures, 5 tables. 30 sources are used.

**Keywords:** automatic speech recognition; speech synthesis; unsupervised learning.

# Содержание

# Используемые определения и термины

**ASR (Automatic Speech Recognition)** – задача перевода человеческой речи в текст.

**CER (Character Error Rate)** – метрика для оценки качества ASR моделей, путём подсчёта верно предсказанных символов.

**MOS (Mean Opinion Score)** – метрика для оценки качества TTS моделей, вычисляется путём средней оценки, выставленной опрашиваемыми или иными методами.

**PER (Phoneme Error Rate)** – метрика для оценки качества ASR моделей, путём подсчёта верно предсказанных фонем.

**TTS (Text To Speech)** – задача генерации человеческой речи по тексту.

# Введение

Синтез речи и автоматическое распознавание речи являются двумя важными задачами в обработке речи и актуальными темами исследований в области искусственного интеллекта. Системы синтеза речи прошли множество этапов эволюции, от раннего синтеза формант, артикуляторного синтеза речи, конкатенативных подходов, до статистических параметрических подходов и до современных нейронных подходов на основе глубокого обучения, которые стали доминировать в задачах работы с речью.

Одна из наиболее актуальных проблем - сложность и большая стоимость сбора транскрибированных данных. Текущие наиболее точные модели обучены на сотнях часов качественной речи, что недоступно для большинства языков и диалектов. Например, крупный языковой корпус MCV [1] включает данные для обучения 91 языка, что составляет лишь небольшой процент от всех языков мира. Для решения этой проблемы мы будем опираться не на наращивание объемов данных, а на более эффективную работу с имеющимися ограниченными ресурсами.

Наибольшее внимание в этой работе уделяется принципу двойственного обучения, его первоначальной задаче и некоторым расширениям. Две задачи машинного обучения являются двойственными, если одна задача отображает из пространства $X$ в пространство $Y$, а другая - из пространства $Y$ в пространство $X$. Грубо говоря, основная идея такого обучения заключается в том, чтобы использовать симметричную структуру задач машинного обучения для получения эффективной обратной связи или регуляризирующих сигналов для улучшения процесса обучения.

Целью данной работы было изучение и сравнение современных методов обучения ASR и TTS моделей в условиях ограниченности ресурсов, проведение экспериментов для воспроизведения и улучшения результата.

Для этого были поставлены следующие задачи:

1) Изучить и сравнить существующие методы обучения без учителя для распознавания речи и синтеза речи;

2) Выбрать наборы данных для проведения экспериментов;

3) Обучить современные модели на парных данных, посчитать метрики CER и MOS;

4) Выбрать подходящие архитектуры для обучения методом двойственной трансформации и разработать пайплайн для обучения;

5) Сравнить результаты разработанных методов и методов обучения с учителем;

6) Проанализировать ошибки и предложить идеи для дальнейшего развития.

В Главе 1 приведен обзор предыдущих работ по темам обучения без учителя в задачах распознавания речи и синтеза речи. В Главе 2 описаны предложенные методы

обучения без учителя в задачах распознавания речи и синтеза речи. В Главе 3 приведены результаты экспериментов с предложенными методами. В заключении приведены краткие выводы по проделанной работе.

# Глава 1. Обзор источников

В этом разделе приведён краткий разбор предыдущих работ по теме обучения без учителя для задач распознавания речи и синтеза речи по тексту.

## 1.1. Распознавание речи

Распознавание речи - это процесс автоматического выявления закономерностей в речевой форме волны. В число закономерностей, которые могут быть обнаружены в речи, входят личность говорящего, язык, эмоции и текстовая транскрипция произнесенной речи. Последнее является основным результатом работы ASR. Наименьшей распознаваемой единицей речи является фонема, которая представляет собой звуки, различающие слова в данном языке.

Первые модели ASR состояли из трех основных компонент: акустической модели, словаря произношения и языковой модели. Акустическая модель рассчитывает вероятность акустических единиц, например фонем, которые могут быть смоделированы с помощью гауссовых моделей смешивания [2] и марковских моделей [3]. Позднее в качестве акустической модели было предложено использовать нейронные сети. Языковая модель рассчитывает вероятности последовательности слов и используются для повышения точности акустических моделей путем включения лингвистических знаний из больших текстовых корпусов. Модели ASR обычно оцениваются с помощью коэффициента ошибок слов, который подсчитывает общее число замен, вставок и удалений, необходимых для воспроизведения верного решения.

Недавние исследования [4; 6; 5] показали, что можно обучать ASR без учителя, когда размеченные данные больше не требуются. В общем случае эти методы представляют двухэтапный конвейер: этап предварительной обработки, за которым следует этап обучения без учителя, целью которого является перевод речи в последовательности фонем. Этап предварительной обработки направлен на преобразование представлений входной речи посредством сегментации речи в признаки, которые лучше соответствуют основному фонетическому содержанию. Целью сегментации речи является обнаружение границ во входной последовательности, чтобы обозначить переход от одной фонемы к другой. На практике границы могут быть определены с помощью кластерных представлений [6] или других методов [8; 7]. Помимо сегментации речи, предшествующие работы также считали необходимым шагом нахождение зависимостей. Размер входной речи уменьшается путем объединения внутрисегментных признаков с помощью k-means кластеризации [4] или анализа главных компонент [6].

В работе [9] авторы отмечают, что точность сегментации может значительно повлиять на последующие шаги, однако сегментация речи без наблюдения обычно далека от совершенства [10]. Это приводит к росту ошибки от сегментации и других этапам обучения. Для решения этих проблем была предложена система wav2vec-U 2.0 [11], которая способна заменить все этапы предварительной обработки речевого сигнала с

помощью нейронной сети «речь-фонема». Основная идея заключается в обучении модели, которая сопоставляет признаки, извлеченные из модели предварительно обученной на немаркированных речевых данных, в последовательность фонем.

## 1.2. Синтез речи

Задача синтеза речи состоит в формировании речевого сигнала по текстовым данным. С развитием технологий, исследовательская цель синтеза речи эволюционировала от разборчивости и ясности к естественности и выразительности. Понятность описывает четкость синтезированной речи, в то время как естественность относится к легкости прослушивания и стилистической согласованности.

Аналогично ASR, ранние модели синтеза речи состояли из трех компонент: языковой модели, акустической модели и синтезатора речи. Компоненты обучаются независимо друг от друга, поэтому ошибки в обучении одной из компонент приводят к общему плохому качеству модели. Для решения этих проблемы существуют методы сквозного синтеза речи, которые объединяют эти компоненты в единую структуру.

В последние годы было предложено несколько подходов для обучения TTS без учителя. Данные для обучения далеко не всегда охватывают полный словарь языка. Для решения этой проблемы текст преобразуется в фонемы после передачи в модель, чтобы модели нужно было только выучить произношение небольшого словаря и соответствие между символами и фонемами [12]. Человеческая орфография и письмо могут иметь одни и те же основные шаблоны. Некоторые работы специально направлены на достижение лучшего распознавания моделей языков из одной языковой семьи. Межъязыковой перенос на основе данных из больших ресурсов предварительное обучение. В работах [14; 13] авторы обучали среднеязычную модель на большом корпусе мандаринского языка. Полученная модель TTS смогла восполнить недостаток тибетских речевых данных. Модель Tacotron из [15] использует межъязыковой перенос для производства высококачественной речи из языков малых народностей, поскольку она была обучена на большом корпусе аудиоданных.

Методы речевой цепочки [16] и двойственной трансформации [17] используют как синтез речи, так и распознавание речи, обучая их одновременно. Такой способ практически не требует размеченных и использует большое число непарных текстов и аудиосообщений. На каждом шаге обучения ASR генерирует текст из речи, а TTS генерирует речь из текста. Полученные образцы составляют пары, на которых обучаются обе модели.

Отдельно стоит отметить методы предварительного обучения для улучшения понимания моделью языка [18; 19]. Например, в качестве дополнительного слоя в TTS может быть добавлена предварительно обученная BERT-модель [18], а декодировщик речи может быть предварительно обучен с помощью авторегрессионного прогнозирования по спектрограмме. Кроме того, речь может быть квантифицирована в дискретные последовательности токенов, напоминающие фонемы [20]. Таким образом, квантован-

ные дискретные лексемы и речь можно рассматривать как псевдопарные данные для предварительного обучения модели TTS, которая затем точно настраивается на небольшом количестве парных текстовых и речевых данных [21].

## Выводы по главе

В данной главе был проведен краткий обзор последних работ по темам обучения без учителя для распознавания речи и синтеза речи. Было показано, что модели могут показывать высокое качество при ограниченном объёме данных для обучения или при полном их отсутствии.

# Глава 2. Описание предложенных методов

## 2.1. Речевая цепочка

Механизм речевой цепочки был вдохновлен процессом общения между людьми. В разговоре слух и воспроизведение речи важны не только для слушателя, но и для говорящего. Одновременно слушая и говоря, говорящий может контролировать свою речь и лучше спланировать, что и как он будет говорить дальше. Дети потерявшие слух часто испытывают трудности с произношением четкой речи из-за неспособности следить за собственной речью.

Для построения процесса обучения при помощи речевой цепочки используются два двойственных модуля. ASR преобразует речь в текст, TTS выполняет обратную задачу. Ключевая идея заключается в совместном обучении моделей на основе принципа двойственной реконструкции, с использованием как парных, так и непарных данных, рис 1.

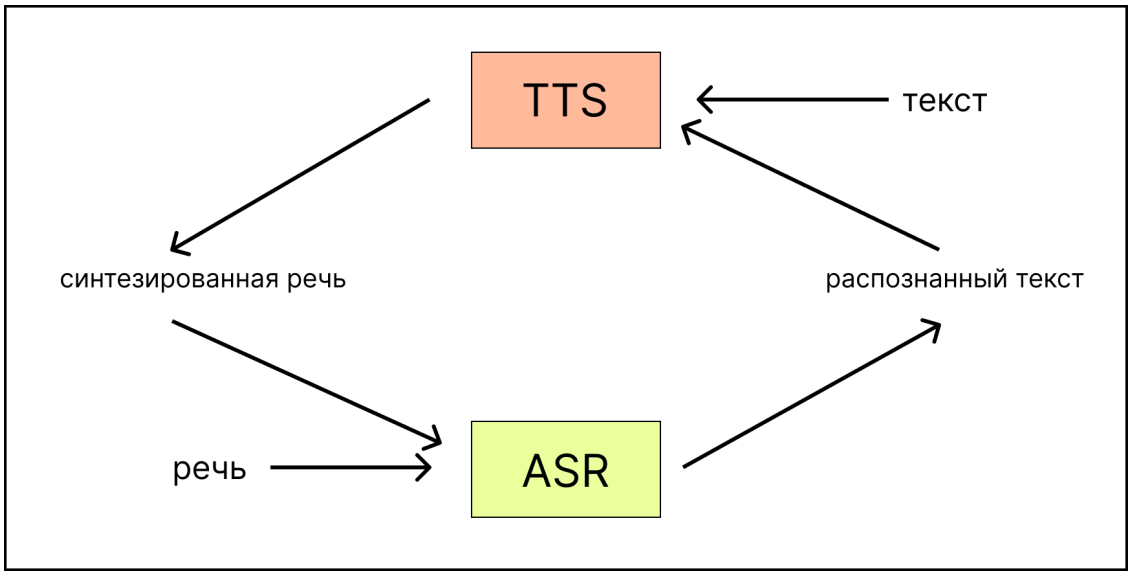



Рис. 1 — Общая архитектура речевой цепочки

Для парных данных $(s, t) \in D$ задача обучения ASR состоит в минимизации логарифмической функции правдоподобия:

$$L_{asr} = -\sum_{(s,t)\in D} \log P(t|s;\phi), \tag{1}$$

где $P(t|s;\phi)$ – вероятность модели с параметром $\phi$ сгенерировать текстовую последовательность $t$ из речевой последовательности $s$.

Задачи обучения TTS на парных данных могут варьироваться. Общая задача состоит в минимизации суммы среднеквадратичных ошибок:

$$L_{tts} = \sum_{(s,t)\in D} (q_1(s) - q_1(f(t;\psi)))^2 + \ldots + \sum_{(s,t)\in D} (q_n(s) - q_n(f(t;\psi)))^2, \tag{2}$$

где $f(t;\psi)$ – предсказание модели, а функции $q_1, \ldots, q_n$ – могут варьироваться в зависимости от задачи. Например, для модели Fastpitch [22] $q_1() = mel()$, $q_2() = dur()$, $q_3() = pitch()$, соответственно, спектрограмма, длительность и высота тона.

Теперь рассмотрим случай, когда данные состоят либо только из текста $t \in T$, либо же только из речи $s \in S$. Тогда дополним недостающие данные следующим образом:

$$\begin{aligned} \hat{t} &= \arg\max P(t|s;\phi) \\ \hat{s} &= f(t;\psi) \end{aligned} \tag{3}$$

Тогда функции потерь 1 и 2 примут вид:

$$\begin{aligned} L_{a\hat{s}r} &= -\sum_{(t)\in T} \log P(t|\hat{s};\phi), \\ L_{t\hat{t}s} &= \sum_{(s)\in S} (s - f(\hat{t};\psi))^2 \end{aligned} \tag{4}$$

По сути задача моделей состоит в реконструкции изначальных данных по сгенерированным.

Общая процедура обучения цепочки машинной речи показана в Алгоритме 1.

**Algorithm 1** Речевая цепочка

Данные $D$, $T$, $S$

1: Инициализируем TTS и ASR случайными весами $\psi, \phi$ или используем предобученные модели
2: **repeat**
3: Считаем $L(\psi; D) + L(\psi; X)$
4: Считаем $L(\phi; D) + L(\phi; Y)$
5: Шаг обучения: обновляем веса моделей
6: **until** не выполнится критерий остановы

В качестве шага может быть выбрана одна эпоха обучения, а порядок действий 4-5 может быть изменен. Тогда алгоритм примет следующий вид:

В работе [16] показано, что речевая цепочка может эффективно работать с данными, как из одного источника, так и из различных. Кроме того алгоритм может эффективно улучшить как одноязычную, так и многоязычную естественную речь. Предобученные модели могут быть эффективно дообучены совсем без парных данных.

**Algorithm 2** Речевая цепочка, последовательное обучение

Данные $D$, $T$, $S$

1: Инициализируем TTS и ASR случайными весами $\psi, \phi$ или используем предобученные модели
2: **repeat**
3: Эпоха обучения TTS
4: Считаем $L(\psi; D) + L(\psi; X)$
5: Обновляем веса TTS
6: Эпоха обучения ASR
7: Считаем $L(\phi; D) + L(\phi; Y)$
8: Обновляем веса ASR
9: **until** не выполнится критерий остановы

Помимо долгого времени для обучения, проблема данного алгоритма состоит в сильном ухудшении качества распознавания модели на новых данных. Одним из возможных решений является интеграция дополнительной модели распознавания диктора, позволяющей изучать отдельные характерисики речи, как поступили в работе [20]. При необходимости работы с новыми речевыми данными, качество распознавания может быть сильно повышено при помощи лишь нескольких парных данных.

### 2.1.1. Обучение модели

В работе [16] авторы дважды обучают модель речевой цепочки, используя 10000 парных и 40000 непарных данных. В первой задаче используются аудиоданные только одного говорящего. В результате удалось достичь повышения качества распознавания речи (с 10.06% до 5.44% в метрике CER) и немного понизить ошибку при генерации речи. Вторая задача состоит из аудиоданных 50 разных человек. В этом случае удалось лишь незначительно улучшить результат ASR (с 26.47% до 19.99% в метрике CER), а качество генерации не повысилось.

## 2.2. Двойственная трансформация

Авторы работы [17] продолжают идею синтеза и распознавания речи в условиях ограниченных ресурсов на основе речевой цепочки. Отличие состоит в том, что непрерывное цикличное обучение дополняется методом снижения чувствительности кодировщика у обеих моделей, используемого в задачах машинного перевода. Кроме того, в задачах работы с последовательностями (в данном случае текстовыми и звуковыми), неверное предсказание начала последовательности практически всегда приводит к неверному предсказанию конца. Из-за этого качество второй половины последовательности в среднем хуже. Эта проблема распространения ошибки ещё больше усугубляется при недостатке данных для обучения.

В работе [17] предложено опираться на двунаправленное моделирование для генерации последовательностей. Тогда правая часть последовательности может иметь

лучшее качество при генерации в противоположном направлении. Соответственно, распространение ошибки при двойственном обучении также будет нивелировано, а также новые данные будут некоторой аугментацией нашего датасета.

### 2.2.1. Понижение чувствительности кодировщика

Для ограничения сходимости модели на парных данных воспользуемся функцией искажения данных $C$. Пусть $X = (x_1, x_2, \ldots, x_n)$ – последовательность данных, $p \in (0, 1)$ – вероятность искажения.

$C_1(X) = (x_1, x_3, x_6, \ldots, x_n)$ – удаляем каждый элемент с вероятностью $p$
$C_2(X) = (x_1, null, x_3, null, null, x_6, \ldots, x_n)$ – маркируем каждый элемент с вероятностью $p$
$C_3(X) = (x_1, \ldots, x_k, x_{k-1}, \ldots, x_{k-n*p}, \ldots, x_n)$ – переворачиваем часть последовательности

В работе [17] используется комбинация $C_2$ и $C_3$ для текстовых и речевых последовательностей. Стоит отметить, что функцию искажения можно применить как к изначальным данным (текст и речь), так и после этапа предварительной обработки (фонемы, спектрограмма).

Функции потерь для задачи понижения чувствительности будут иметь следующий вид:

$$L_{dae} = L_{asr}(\phi; C(T)) + L_{tts}(\psi; C(S)) \tag{5}$$

### 2.2.2. Двунаправленное моделирование последовательностей

Для определения цели двунаправленного моделирования нам необходимо дополнить функцию потерь обратной последовательностью:

$$\begin{aligned} L_{\overrightarrow{dae}} &= L_{asr}(\phi; C(\overrightarrow{T})) + L_{tts}(\psi; C(\overrightarrow{S})) \\ L_{\overleftarrow{dae}} &= L_{asr}(\phi; C(\overleftarrow{T})) + L_{tts}(\psi; C(\overleftarrow{S})), \end{aligned} \tag{6}$$

где $\overrightarrow{T}$ - классическое направление последовательностей, а $\overleftarrow{T}$ - последовательность в обратном порядке.

Итоговая функция потерь будет содержать сумму функций 1, 2, 4, 5 и 6 с применением двунаправленного моделирования:

$$
\begin{aligned}
L = & L_{\overrightarrow{asr}} + L_{\overleftarrow{asr}} \\
& + L_{\overrightarrow{tts}} + L_{\overleftarrow{tts}} \\
& + L_{\overrightarrow{\hat{asr}}} + L_{\overleftarrow{\hat{asr}}} \\
& + L_{\overrightarrow{\hat{tts}}} + L_{\overleftarrow{\hat{tts}}} \\
& + L_{\overrightarrow{dae}} + L_{\overleftarrow{dae}}
\end{aligned} \tag{7}
$$

### 2.2.3. Обучение модели

Эксперименты с обучением моделей методом двойственной трансформации описаны в работе [17] . Авторы используют лишь 200 пар данных и около 25000 непарных данных. Результаты показали, что этот метод может сильно повысить качество моделей. Качество синтеза речи повысилось (с 0 до 2.68 по метрике MOS), точность распознавания возросла (с 72.3 до 11.7 в метрике PER).

## 2.3. Продолжение идей двойственной трансформации

В главе 2.1 для обучения модели требовалось 10000 пар данных, в главе 2.2 уже только 200. Создатели LRSpeech [23] продолжают идеи [17] и работают в условиях следующих ограничений: стоимость сбора данных должна быть чрезвычайно мала, крайне высокая точность итоговой модели, цель на промышленное развертывание.

Идея их подхода состоит в последовательном использовании различных типов данных на различных этапах обучения. Первоначально мы предобучаем модели на наиболее качественных доступных данных из любых языков и дообучаем на данных целевого говорящего. Второй этап состоит в использовании парных данных голоса целевого говорящего и непарных данных из любых других источников для обучения методом речевой цепочки. Третий этап аналогичен второму, только вместо непарных используется полностью синтезированные данные. В итоге мы получаем качественную «одноголосую» TTS модель и ASR модель, способную распознавать различные голоса. Далее мы более подробно распишем эти этапы обучения.

### 2.3.1. Качество итоговой модели

Общая архитектура итоговой модели совпадает с трансформером из [17]. Используя только 5 минут качественной транскрибированной речи, и руководствуясь всеми 3 этапами обучения, авторы сумели достичь высокого качества ASR (10.3% CER) и TST (3.65% MOS). Эти результаты были получены для латвийского языка, основной массив данных был крайне низкого качества.

## 2.4. Распознавание речи не-носителя языка

Как было указано в 2.1 точность распознавания речи сильно падает, если модель не обучалась на этом голосе прежде. Ситуация ухудшается, если говорящий не является носителем языка, зачастую ASR не может распознать подобную речь.

Простой и прямой подход к повышению точности распознавания для не-носителей языка заключается в том, чтобы обучить модель ASR для не-носителей языка. Однако, тяжело представить, где взять такое число транскрибированных данных, чтобы обучить по одной модели для каждой пары (национальность, язык). Поэтому в работе [24] авторы предлагают использовать два вида непарных данных: текст и речь, в точности как в механизме речевой цепочки.

### 2.4.1. Метод, основанный на принципе двойственной реконструкции

Система распознавания речи не-носителя языка опирается на четыре модели:

- Языковая модель $M_T$, обученная на языковом корпусе без парных данных. Модель необходима для: генерации новых предложений и определения вероятности того, что предложение является корректным.
- Акустическая модель $M_S$, обученная на непарных аудиоданных речи не-носителя языка. Модель выполняет функции, аналогичные $M_T$, но для аудиоданных: генерирует новые последовательности и вычисляет вероятность того, что предложенная последовательность произнесена человеком.
- ASR модель $M_{S2T}$.
- TTS модель $M_{T2S}$.

Обучение проводится в два этапа: предобучение и двойственная трансформация. В первом этапе модели $M_T$ и $M_S$ независимо обучаются на непарных данных. На этом этапе модели $M_{S2T}$ и $M_{T2S}$ могут быть обучены на небольшом объеме парных данных. В таком случае второй этап займёт гораздо меньше времени.

Во время этапа двойственной трансформации, языковая модель $M_T$ и акустическая модель $M_S$ помогают обучаться двум другим моделям. Процесс обучения повторяет принцип двойственной реконструкции. Но в данном случае у нас есть два вида цепочек, первая начинается с речи, вторая с текста. Рассмотрим механизм обучения для одной из цепочек:

1) Имеем предложение $t$, которое либо доступно в датасете, либо сгенерировано $M_T$.
2) Модель $M_{T2S}$ получает $t$ на вход и генерирует аудиоданные $s = M_{T2S}(t)$.
3) Акустическая модель $M_S$ оценивает аудиоданные: $r_s = M_S(s)$, где $r_s$ - вероятность, что $s$ - человеческая речь.

4) Модель $M_{S2T}$ переводит $s$ в текст. Теперь мы находим вероятность, что предложение $t$ может быть получено из $s$: $r_t = \log P(t|s; M_{S2T})$.

Постоянно получая вероятности $r_s$ и $r_t$, мы обновляем веса моделей $M_{S2T}$ и $M_{T2S}$.

### 2.4.2. Качество итоговой модели

В результате этого подхода удалось добиться качества ASR при распознавании речи не-носителей английского языка в 17.1% в метрике CER.

# Глава 3. Эксперименты

Описанные модели были обучены на англоязычном наборе данных AN4 [25], который состоит из записей людей, произносящих адреса, имена и т. д. Всего датасет состоит из 1078 образцов транскрибированной речи от 82 различных человек. Выборка для обучения состоит из 200 случайных образцов, выборка для валидации из 130. Речевая цепочка и двойственная трансформация используют оставшуюся часть набора в качестве «непарных» данных. Все модели обучались в течение 200 эпох, с валидацией каждую пятую эпоху. Необходимые для обучения параметры аудиоданных указаны в таблице 1.

| **Параметр** | **Значение** |
|---|---|
| частота дискретизации | 22050 |
| минимальная высота звука | 65.4 |
| максимальная высота звука | 1998.5 |
| средняя высота звука | 190.3 |
| среднеквадратичное отклонение высоты звука | 265.6 |

Таблица 1 — Параметры аудиоданных в AN4

Архитектура и значения параметров моделей инициализируются при помощи библиотеки Neural Models [26] и файлов-гонфигураторов OmegaConf. Для инициализации собственных моделей были добавлены параметры, указанные в таблице 2.

| **Название параметра** | **Описание** |
|---|---|
| use_dae | использовать ли понижение чувствительности |
| corruption_prob | вероятность искажения |
| use_bsm | использовать ли двунаправленное моделирование |
| spec_gen_config_path | путь до файла-конфигуратора генератора спектрограмм |
| epochs_to_pretrain | число эпох для предобучения |
| paired_manifest_path | путь до парных данных |
| unpaired_text_manifest_path | путь до непарных текстовых данных |
| unpaired_speech_manifest_path | путь до непарных аудиоданных |

Таблица 2 — Описание дополнительных параметров для собственных моделей

Все данные для обучения передаются в виде манифестов в формате .json. Кажрый образец на новой строчке: {«audio_filepath»: , «duration»: , «text»: }.

Мы используем оптимизатор Novograd [27] с коэффицентом скорости обучения $lr = 0.05$, значениями $\beta_1 = 0.8, \beta_2 = 0.25$, регуляризацией $\lambda = 0.001$.

## 3.1. Выбор моделей

### 3.1.1. Сравнение ASR на метрике CER

В таблице 3 указаны сравниваемые архитектуры моделей распознавания речи. Модели jasper5x1, quartznet15x5, contextnet_rnnt, conformer_transducer использовали в качестве словаря токенов английский алфавит из строчных букв. Для остальных моделей был создан словарь на основе разбиения на диграммы, итоговый размер составил 339 токенов. Small аналог carnelinet (5 блоков, в каждом 1 подблок) показывает наилучший результат среди всех моделей, при этом имея относительно небольшой вес. Вероятно, это обусловлено техникой «башенного» исключения, описанной в оригинальной работе [28].

| **Модель** | **CER** | **Число параметров (M)** | **Итоговый вес (Мб)** |
|---|---|---|---|
| jasper5x1 | 0.95 | 48.8 | 195.2 |
| quartznet15x5 | 0.88 | 18.9 | 75.69 |
| citrinet15x1 | 0.8 | 21 | 84.1 |
| carnelinet5x1 | 0.7 | 7.5 | 68 |
| contextnet_rnnt | inf | 2.3 | 15.1 |
| conformer_ctc_bpe | inf | 13 | 52.1 |
| conformer_transducer | inf | 14.2 | 56.7 |
| lstm_transducer_bpe | inf | 46.6 | 186.28 |
| lstm_ctc_bpe | inf | 137 | 548 |

Таблица 3 — Сравнение ASR моделей

Рисунок 2 показывает процесс обучения моделей jasper5x1, quartznet15x5, citrinet15x1, carnelinet5x1.

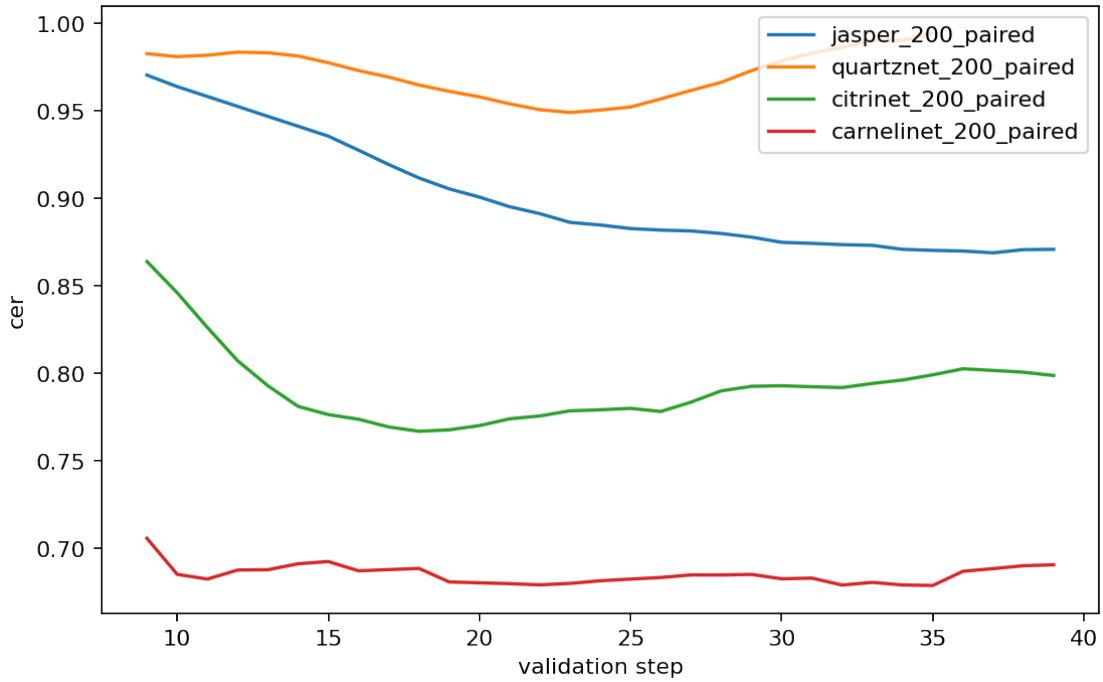



Рис. 2 — Процесс обучения ASR моделей

Выберем для дальнейших экспериментов модель carnelinet5x1, так как она показывает наилучшие результаты ан валидационной выборке. Будем применять функцию искажения $C$ в процессе обучения. После конвертации аудиоданных в спектрограмму каждое значение с вероятностью $corruption\_prob$ зануляется. На рисунке 3 представлены результаты обучения carnelinet5x1 с применением функцией искажения для вероятностей от 0 до 0.5. Функция искажения с $corruption\_prob = 0.5$ значительно улучшает точность нашей модели. Авторы [17] используют значение 0.3, но исходя из текущих экспериментов, это сильно зависит от данных и архитектуры.

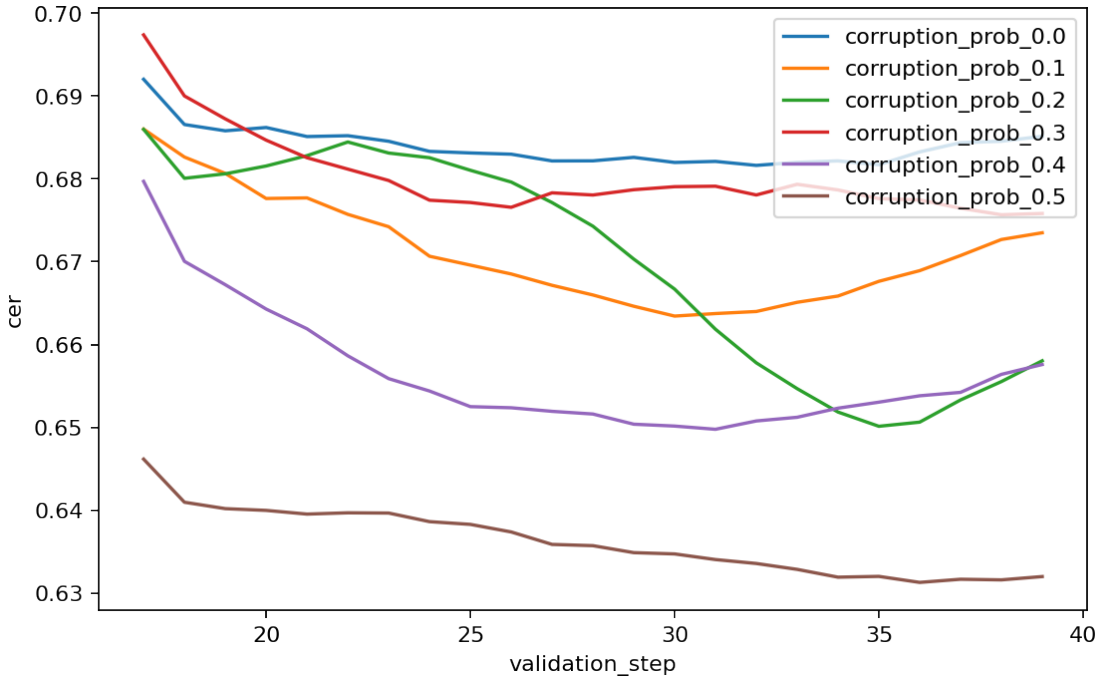



Рис. 3 — carnelinet5x1 с различными коэффицентами функции искажения

Отдельно отметим модель quartznet15x5, которая совсем не сходится на таком числе данных для обучения. Аналогично предыдущему результату, функция искажения значительно улучшает точность модели на валидационной выборке (лучший результат достигается при $corruption\_prob = 0.4$, рисунок 4.

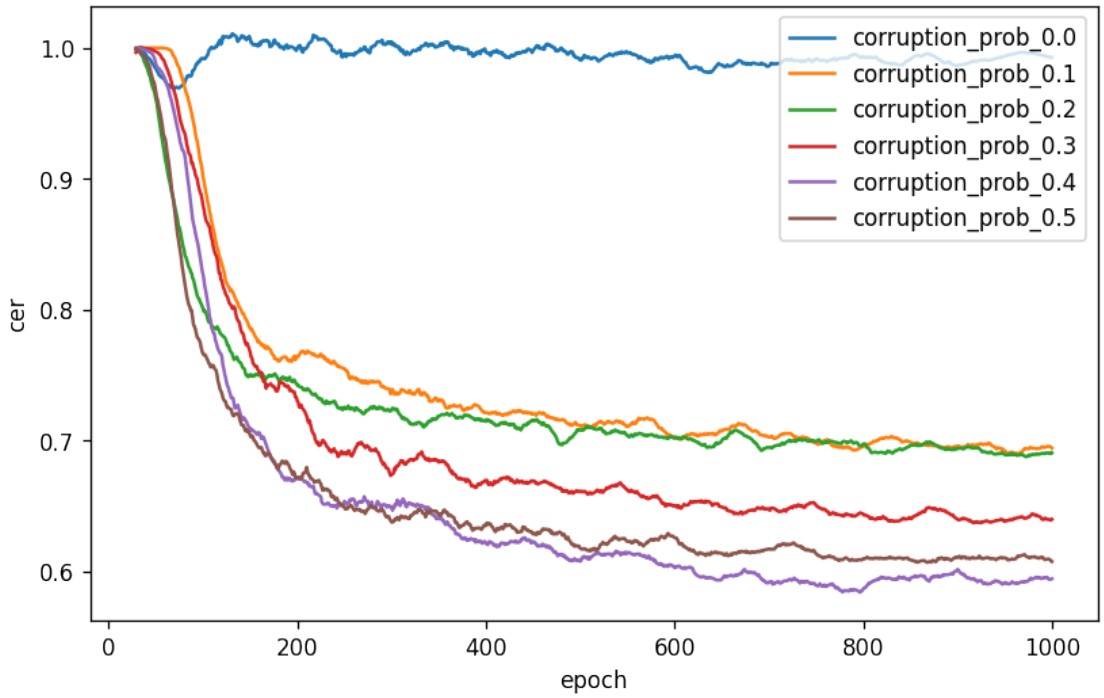



Рис. 4 — quarznet15x5 с различными коэффицентами функции искажения

После нахождения наилучшего значения для коэффицента функции искажения попробуем добавить двунаправленное моделирование в процесс обучения модели. На

рисунке 5 представлен CER при обучении carnelinet5x1 без *dae*, с *dae* и с *dae* + *bsm*.

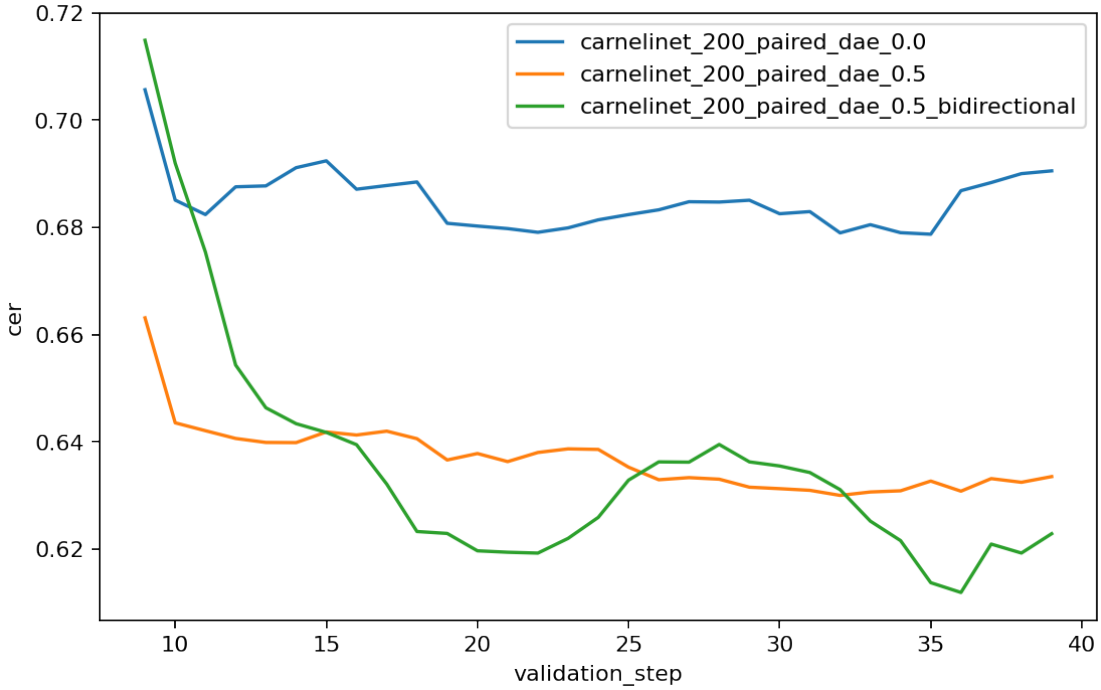



Рис. 5 — Добавляем двунаправленное моделирование в процесс обучения

### 3.1.2. Сравнение TTS

Оценка качества моделей синтеза текста MOS предсказывается при помощи предобученной модели wav2vec-2.0 [29].

| **Модель** | **MOS** | **Число параметров (M)** | **Итоговый вес (Мб)** |
|---|---|---|---|
| tacotron2 | neg | 22.8 | 91.3 |
| glow_tts | neg | 28.6 | 114.5 |
| fastpitch | 2.02 | 32.2 | 128.1 |
| fastspeech2 | 1.81 | 7.5 | 68 |
| mixertts | 2.24 | 27.2 | 112.8 |

Таблица 4 — Сравнение TTS моделей

## 3.2. Двойственная трансформация

Аналогично работе в главе 2.3 мы предобучаем модели перед циклом двойственной трансформации в течение 50 эпох. Процесс обучения выполняется согласно алгоритму 2. Из-за необходимости постоянно вычислять промежуточные значения для Mixer-tts и генерировать предсказания, одна эпоха обучения занимает гораздо больше времени, около 21 минуты.

Рисунок 6 сравнивает указанную выше модель с обученной при помощи метода двойственной трансформации.

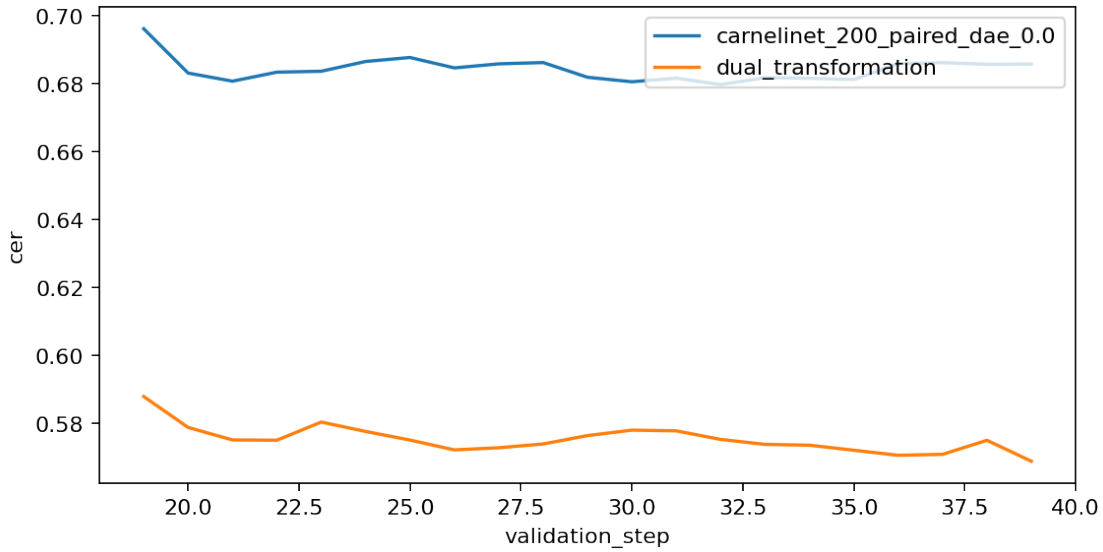


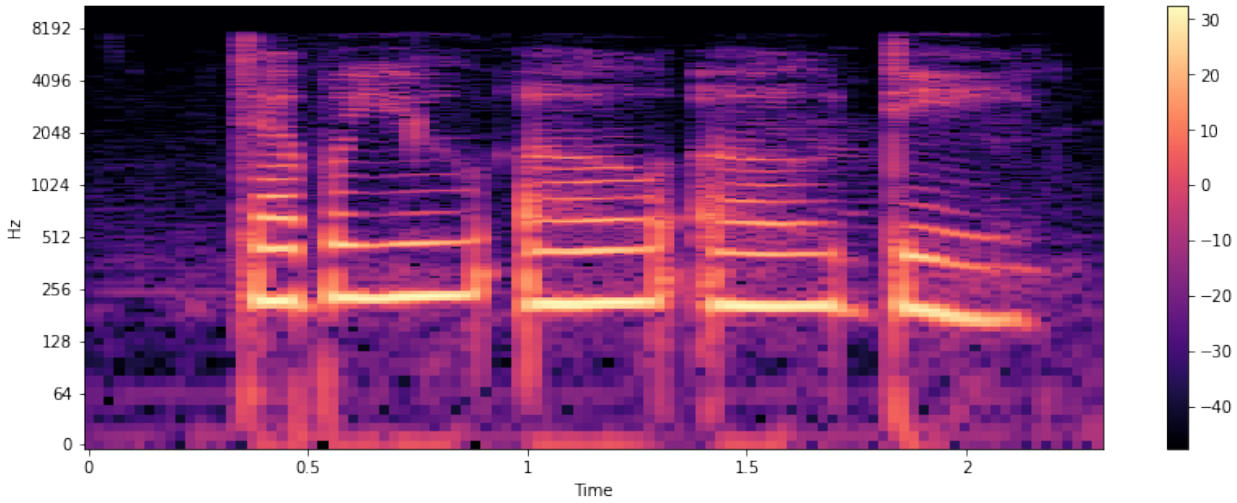

Рис. 6 — Обучение carnelinet5x1 при помощи двойственной трансформации

### 3.2.1. Качественный анализ моделей

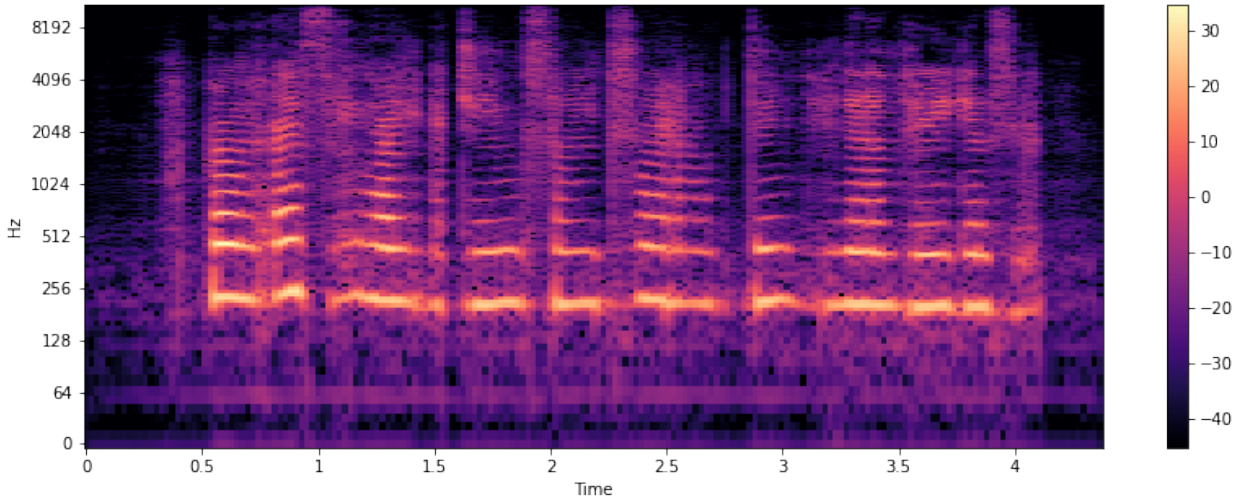
Рассмотрим итоговые значения метрик в таблице 5 и несколько примеров. На изображении 7 рассмотрено 2 спектрограммы одного образца из AN4. Модель пропускает часть фразы, а именно объединяет «O O» в «O». Но высота предсказания совпадает с оригиналом, а шум практически отсутствует.

| **Модель** | **CER** | **MOS** | **CER** | **MOS** |
|---|---|---|---|---|
| | **AN4** | | **LJSpeech** | |
| carnelinet1x5 | 0.7 | | 0.87 | |
| carnelinet1x5+dae_0.5 | 0.637 | | 0.82 | |
| carnelinet1x5+dae_0.5+bsm | 0.62 | | 0.812 | |
| carnelinet1x5+dae_0.5+bsm+dt | 0.56 | | 0.64 | |
| mixertts | | 2.24 | | 0 |
| mixertts+dae_0.5+bsm+dt | | 2.81 | | 1.69 |

Таблица 5 — Итоговые метрики



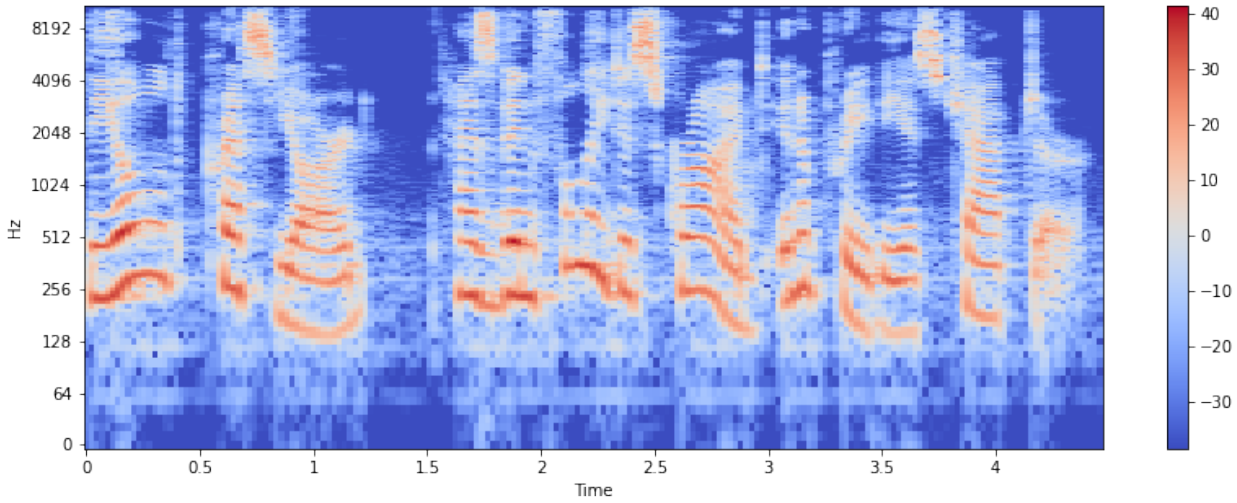

(a)



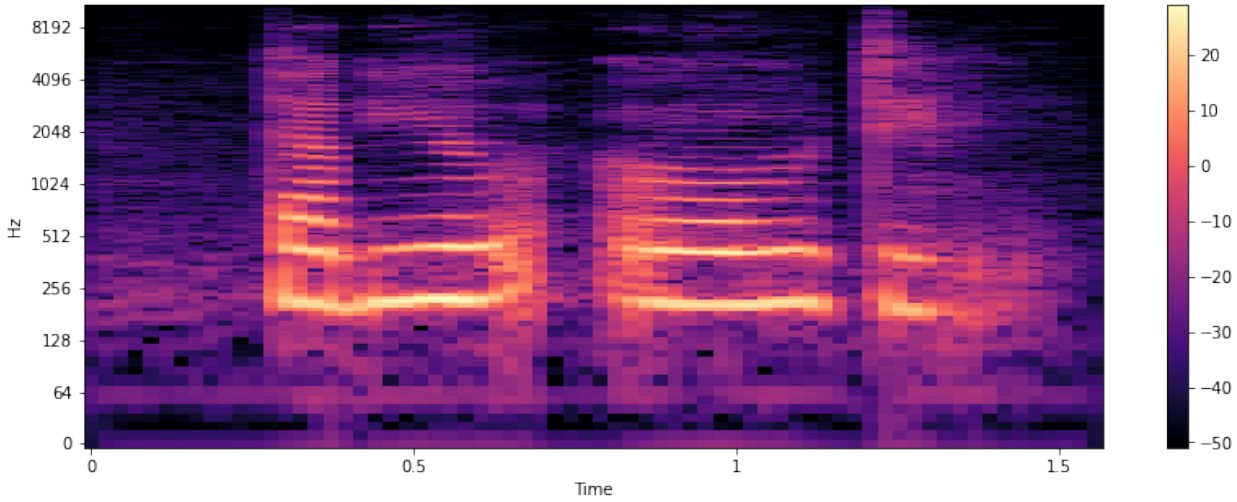

(b)

Рис. 7 — (a) спектрограмма оригинального аудио, женщина проговаривает «R O O M» (b) предсказание модели

Теперь рассмотрим пример из LJSpeech [30]: фраза «the old press yard has been fully described in a previous chapter», рисунок 8. В этом случае модель не понимает большей части сказанного, так как не встречала таких диграмм при обучении.




(a)




(b)

Рис. 8 — (a) спектрограмма оригинального аудио, женщина проговаривает «the old press yard has been fully described in a previous chapter» (b) предсказание модели

## Выводы по главе

В третьей главе были показаны и проанализированы результаты экспериментов с методами, представленными во второй главе. Как и ожидалось, качество синтеза и распознавания речи повысилось, но из-за малого размера датасета, модель очень плохо воспринимает новые данные.

# Заключение

В данной работе были рассмотрены методы, основанные на принципе двойственного обучения. Было экспериментально показано, что качество модели можно повышать без дополнительных парных данных.

Для проведения экспериментов выбран набор данных, отличающийся малым размером и большим числом различных говорящих, что делает обучение моделей на нём достаточно сложной задачей. Рассмотрены различные архитектуры ASR и TTS моделей, с упором на малое число параметров и способности моделей обучаться при малых ресурсах. На основе выбранных архитектур и метода двойственной трансформации получена модель, превосходящая качеством как ASR, так и TTS, обученные независимо.

Произведен анализ результатов качества итоговой модели и описаны причины ухудшения работы модели на данных из других источников. При помощи библиотеки Neural Modules и языка Python был разработан пайплайн для проведения дальнейших экспериментов.

# Список использованных источников


1. Common voice: A massively-multilingual speech corpus / R. Ardila [и др.] // arXiv preprint arXiv:1912.06670. — 2019.
2. Positional cloning of the mouse obese gene and its human homologue / Y. Zhang [и др.] // Nature. — 1994. — Т. 372, № 6505. — С. 425—432.
3. *Juang B. H.*, *Rabiner L. R.* Hidden Markov models for speech recognition // Technometrics. — 1991. — Т. 33, № 3. — С. 251—272.
4. Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings / D.-R. Liu [и др.] // arXiv preprint arXiv:1804.00316. — 2018.
5. Unsupervised speech recognition via segmental empirical output distribution matching / C.-K. Yeh [и др.] // arXiv preprint arXiv:1812.09323. — 2018.
6. Unsupervised speech recognition / A. Baevski [и др.] // Advances in Neural Information Processing Systems. — 2021. — Т. 34.
7. *Wang Y.-H.*, *Lee H.-y.*, *Lee L.-s.* Segmental audio word2vec: Representing utterances as sequences of vectors with applications in spoken term detection // 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). — IEEE. 2018. — С. 6269—6273.
8. Audio word2vec: Sequence-to-sequence autoencoding for unsupervised learning of audio segmentation and representation / Y.-C. Chen [и др.] // IEEE/ACM Transactions on Audio, Speech, and Language Processing. — 2019. — Т. 27, № 9. — С. 1481—1493.
9. Unsupervised cross-modal alignment of speech and text embedding spaces / Y.-A. Chung [и др.] // Advances in neural information processing systems. — 2018. — Т. 31.
10. *Kreuk F.*, *Keshet J.*, *Adi Y.* Self-supervised contrastive learning for unsupervised phoneme segmentation // arXiv preprint arXiv:2007.13465. — 2020.
11. Towards End-to-end Unsupervised Speech Recognition / A. H. Liu [и др.] // arXiv preprint arXiv:2204.02492. — 2022.
12. Non-autoregressive neural text-to-speech / K. Peng [и др.] // International conference on machine learning. — PMLR. 2020. — С. 7586—7598.
13. *Guo W.*, *Yang H.*, *Gan Z.* A DNN-based Mandarin-Tibetan cross-lingual speech synthesis // 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). — IEEE. 2018. — С. 1702—1707.
14. Deep learning for mandarin-tibetan cross-lingual speech synthesis / W. Zhang [и др.] // IEEE Access. — 2019. — Т. 7. — С. 167884—167894.

15. End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning / T. Tu [и др.] // arXiv preprint arXiv:1904.06508. — 2019.

16. *Tjandra A.*, *Sakti S.*, *Nakamura S.* Listening while speaking: Speech chain by deep learning // 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). — IEEE. 2017. — С. 301—308.

17. Almost unsupervised text to speech and automatic speech recognition / Y. Ren [и др.] // International Conference on Machine Learning. — PMLR. 2019. — С. 5410—5419.

18. Semi-supervised training for improving data efficiency in end-to-end speech synthesis / Y.-A. Chung [и др.] // ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). — IEEE. 2019. — С. 6940—6944.

19. Word embedding for recurrent neural network based TTS synthesis / P. Wang [и др.] // 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). — IEEE. 2015. — С. 4879—4883.

20. VQVAE unsupervised unit discovery and multi-scale code2spec inverter for zerospeech challenge 2019 / A. Tjandra [и др.] // arXiv preprint arXiv:1905.11449. — 2019.

21. Towards unsupervised speech recognition and synthesis with quantized speech representation learning / A. H. Liu [и др.] // ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). — IEEE. 2020. — С. 7259—7263.

22. *Łańcucki A.* Fastpitch: Parallel text-to-speech with pitch prediction // ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). — IEEE. 2021. — С. 6588—6592.

23. Lrspeech: Extremely low-resource speech synthesis and recognition / J. Xu [и др.] // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. — 2020. — С. 2802—2812.

24. Dual supervised learning for non-native speech recognition / K. Radzikowski [и др.] // EURASIP Journal on Audio, Speech, and Music Processing. — 2019. — Т. 2019, № 1. — С. 1—10.

25. *NVidia*. The AN4 Speech Dataset. — 2021. — `https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/stable/asr/datasets.html`.

26. Nemo: a toolkit for building ai applications using neural modules / O. Kuchaiev [и др.] // arXiv preprint arXiv:1909.09577. — 2019.

27. Stochastic gradient methods with layer-wise adaptive moments for training of deep networks / B. Ginsburg [и др.] // arXiv preprint arXiv:1905.11286. — 2019.

28. CarneliNet: Neural Mixture Model for Automatic Speech Recognition / A. Kalinov [и др.] // arXiv preprint arXiv:2107.10708. — 2021.

29. *AndreevP*. WV-MOS. — 2022 ; — [Online; accessed 19-May-2022]. https://github.com/AndreevP/wvmos.

30. *Ito K.*, *Johnson L.* The LJ Speech Dataset. — 2017. — https://keithito.com/LJ-Speech-Dataset/.