ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Образовательная программа «Прикладная математика и информатика»

Отчет о программном проекте

на тему Разработка MVP сервиса синтеза речи на английском языке
(промежуточный, этап 1)

Выполнил:

студент группы БПМИ203 _____ А. Е. Ишутин
Подпись И.О. Фамилия

15.05.2022
Дата

Принял:

руководитель проекта Святослав Владиславович Шишкин
Имя, Отчество, Фамилия

lead ml engineer
Должность, ученое звание

АО «Тинькофф Банк»
Место работы (Компания или подразделение НИУ ВШЭ)

Дата проверки 15.05. 2022 10 _____
Оценка (по 10-тибалльной шкале) Подпись

Москва 2022

Реферат

Целью данной работы является создание MVP text2speech сервиса для английского языка. В качестве результата есть телеграмм-бот, который в ответ на текстовое сообщение присылать аудио. Кроме того, через него можно посмотреть аналитику по сервису - скорость синтеза и т. д.

По результатам работы был получен RTF 4.5 на сервере с GPU NVIDIA® Tesla® V100. RTF тем больше, чем длиннее аудио, т. к. задержки в коммуникации сервисов становятся менее значимыми.

Ключевое слово	Описание
text2speech, tts	процесс/технология синтеза устной речи (аудио) по тексту
Mel-спектрограмма	Это спектрограмма, в которой по горизонтальной оси отложено время. По вертикальной частоты в специальной Mel-шкале. За цвет отвечает интенсивность амплитуды.
вокодер	Модель, которая из Mel-спектрограммы делает полноценное аудио
акустика, акустическая модель	Модель, которая из текста, генерирует Mel-спектрограмму
MVP, minimum viable product	минимальный жизнеспособный продукт, т. е. самый простой прототип, все еще являющийся достаточно полезным для пользователя.
Семиотический класс	Множество лексических единиц, которые обозначают объекты одного вида и имеют схожие правила написания и произношения
NLP	Область ML, которая занимается обработкой естественного языка. Также называется и сама обработка.
FST	Модификация детерминированного конечного автомата, в которого добавили выходной алфавит и функцию выходов.
RTF, real-time factor	показывает аудио какой длительности может генерировать система за 1 секунду работы

Содержание

Введение

Синтез речи это процесс создания из текста на каком-то языке аудио, содержащее речь, которую носитель этого языка распознает как этот же самый текст. Синтез речи обычно используется в системах с голосовым управлением таким, как, например, голосовые помощники.

Во многих ситуациях можно было бы записать отдельные куски речи конкретного человека и склеивать их. Это будет работать в системах с ограниченным количеством реплик. Например, навигаторах. Однако в общем случае могут потребоваться произвольные фразы, которые невозможно хорошо синтезировать этим способом. В данный момент state of the art подходы используют DL (deep learning).

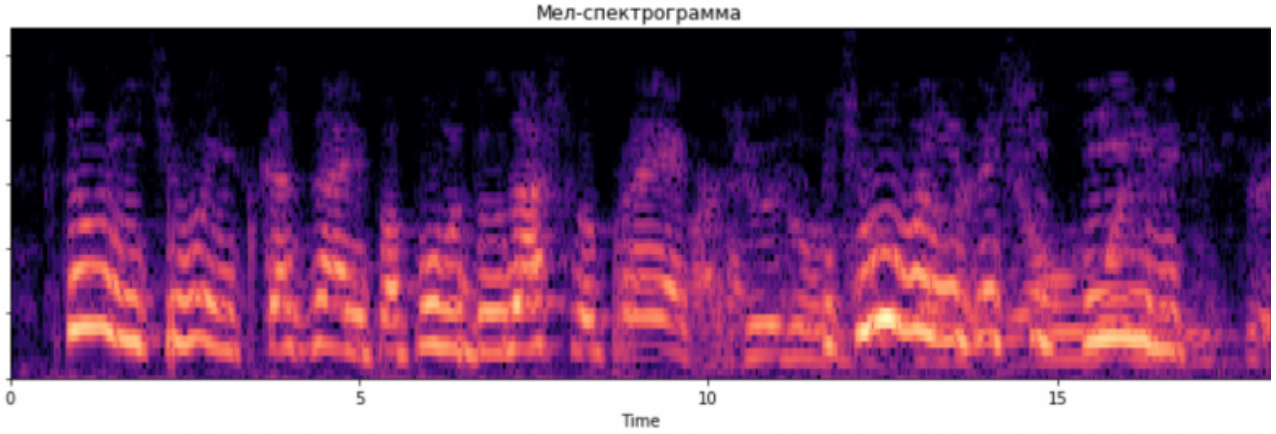
Типичный пайплайн состоит из двух больших этапов:

1. NLP-фронтенд
2. собственно аудиопроцессинг

NLP (Natural Language Processing) фронтенд - включает в себя нормализацию текста и перевод графем в фонемы. Под нормализацией подразумевается нормализацией каждого из семиотических классов, например, раскрытие сокращений и спец. символов (“km” в “kilometer”), работа с числительными (“3rd” в “third”, “1924” в “nineteen twenty four”). В зависимости от контекста числа могут нормализоваться в разные числительные: “in 1924” в “in nineteen twenty four”, но “1924$” в “nine thousands nine hundred twenty four dollars”. Это создает неопределенность. Обычно ее разрешают в 2 этапа. На первом с помощью FST (Finite State Transducer))выделяются участки текста, которые нужно нормализовывать. Если можно однозначно их нормализовать они нормализуются. После этого, seq2seq сетку прогоняют по местам, которые нужно нормализовать, отбирая возможные варианты. Затем FST для пост-обработки выбирают самый адекватный вариант.

Для большинства слов есть поставленное ударение и фонемы в словаре.Это покрывает примерно 90% случаев, однако в остальных ситуациях случается out-of-vocabulary проблема. Ее обычно решает с помощью простой нейросети, например, RNN. Кроме того, существуют омографы (“cOnflict”, “conflIct”), которые добавляют сложностей. Их разрешают с помощью нейросетей таких как BERT, например, смотря на эмбеддинги контекстов слов.

Второй большой этап делится на: создание MEL-спектрограммы с помощью акустической модели (например, такотрона), создание аудио на основе спектрограммы с помощью вокодера (в нашем случае hi-fi gan) и часто улучшение аудио с помощью третьей нейросети.

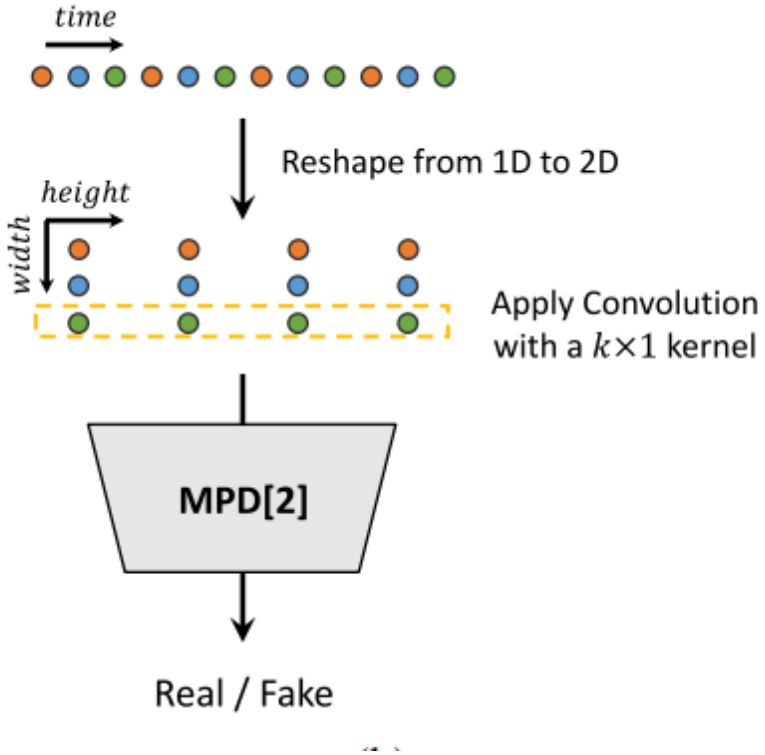


На данный момент синтезом речи занимаются и крупные компании типа Яндекс. Mail.ru, Apple, Google и другие, в том числе экстремистские, так и стартапы. Например, Silero. По теме ежегодно публикуется множество статей и SoTA в разных частях синтеза меняется соответственно. Нашей целью было создать сервис синтеза речи, который бы включал в себя методы и этапы, характерные для SoTA подходов в области. А затем дать удобный интерфейс через телеграм бота, чтобы сервисом можно было пользоваться и тестировать.

Теоретическая часть

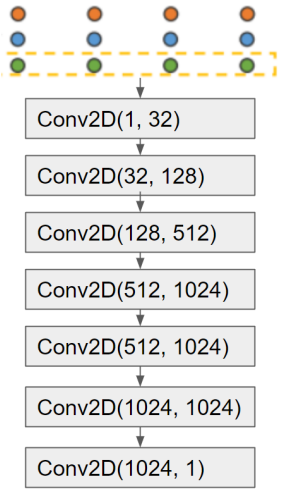
Вокодер

В качестве вокодера был выбран HiFi GAN. Это такая генеративно-состязательная сеть, у которой один генератор и два дискриминатора. Дискриминаторы пытаются понять, настоящее ли аудио им дали или же сгенерированное. В отличие от классического GAN-а, здесь на вход генератору не подается случайный шум. Первый дискриминатор называется MPD - multi-period discriminator. Его задача - проверить, что все для разных периодов аудио консистентно. Он смотрит для периодов 2, 3, 5, 7, 11 в отдельности. В итоге MPD состоит из 5 суб-дискриминаторов



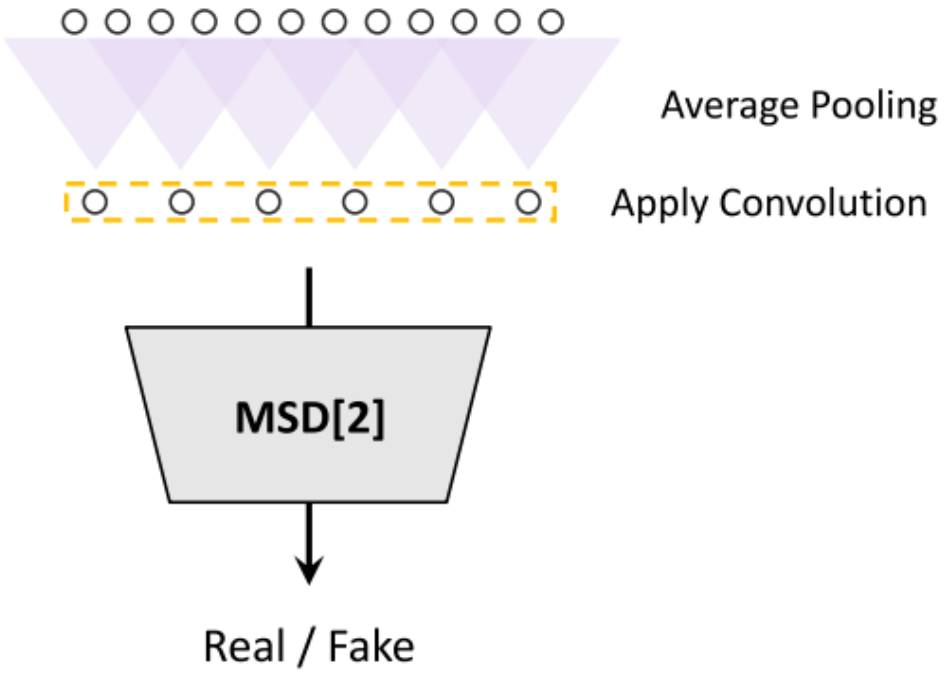
На схеме показана предобработка данных для периода 3.

Вот сама архитектура:

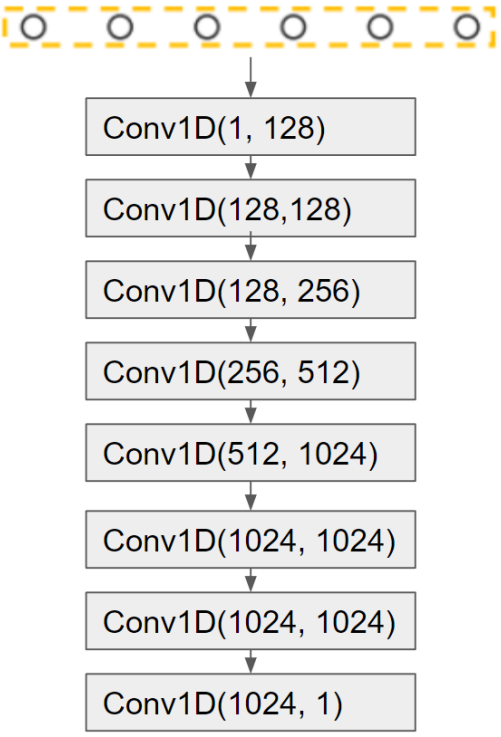


Второй дискриминатор это MSD (Multi-Scale Discriminator) его задача проверить консистентность аудио в масштабе. Для этого он сглаживает сигнал с помощью Average Pooling и уменьшает его

длину.



Для каждого из коэффициентов сглаживания 1, 2, 4 свой суб-дискриминатор. Архитектура на картинке:



Сам генератор состоит из идущих подряд сверточных блоков. Для повышения рецептивного поля используется набор Diluted Convolution с разным параметром dilation, результат применения которых суммируется.

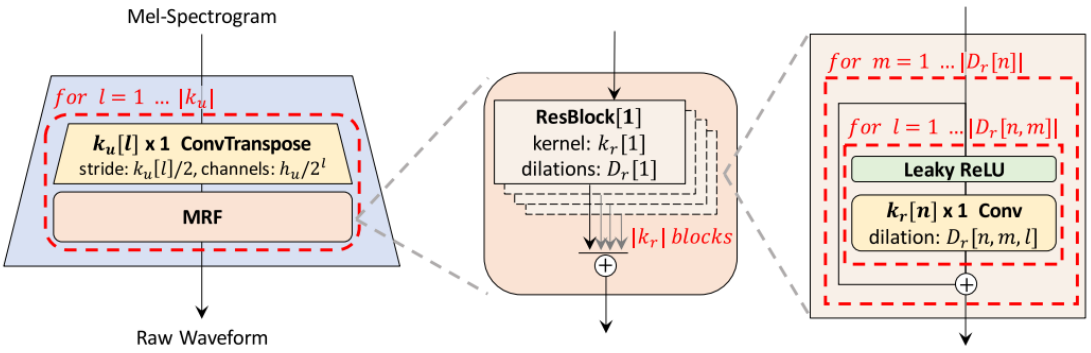


Figure 1: The generator upsamples mel-spectrograms up to $|k_u|$ times to match the temporal resolution of raw waveforms. A MRF module adds features from $|k_r|$ residual blocks of different kernel sizes and dilation rates. Lastly, the n-th residual block with kernel size $k_r[n]$ and dilation rates $D_r[n]$ in a MRF module is depicted.

Функция потерь в формульном виде записываются так:

$$\mathcal{L}_{Adv}(D;G) = \mathbb{E}_{(x,s)}\left[(D(x)-1)^2 + (D(G(s)))^2\right]$$

$$\mathcal{L}_{Adv}(G;D) = \mathbb{E}_{s}\left[(D(G(s))-1)^2\right]$$

Для краткости оба дискриминатора обозначаются как один, на самом деле к каждому дискриминатора применяется такая функция потерь, а к генератору соответствующая сумма.
D - обозначается как дискриминатор, G - как генератор. Под x понимается настоящее аудио, под s - mel-спектрограмма. Считается, что дискриминатор предсказывает вероятность, что аудио настоящее, т. е. для оригинальных аудио идеальный дискриминатор должен предсказывать 1.0

У генератора есть свои специфические функции потерь:

$$\mathcal{L}_{Mel}(G) = \mathbb{E}_{(x,s)}\left[||\phi(x) - \phi(G(s))||_1\right]$$

У генератора, где **φ** обозначает функцию, переводящую аудио в mel-спектрограмму.

$$\mathcal{L}_{FM}(G;D) = \mathbb{E}_{(x,s)}\left[\sum_{i=1}^{T}\frac{1}{N_i}||D^i(x) - D^i(G(s))||_1\right]$$

Эта функция штрафует генератор за разницу между фичами внутри дискриминаторов для настоящего аудио и сгенерированного.

Изначально GAN обучается на оригинальных мел-спектрограммах. На второй стадии обучения, он берет спектрограммы от акустической модели.

Примечание: не все тонкости HiFi GAN-а были упомянуты здесь.

Акустика

При реализации акустика использована модель Fastpitch, основанная на архитектуре Transformer.

Детальное описание архитектуры представлено в статье [9]. Детальное описание модели Fastpitch представлено в статье [8].
Модель Fastpitch построена на основе 2-х feed-forward трансформеров (FFT). После первичного преобразования входных токенов первый блок трансформера генерирует представление входной последовательности (h), которое используется для предсказания длительности и средней высоты звука, соответствующего каждому символу. Результаты объединяются и отправляются во второй блок FFT, который преобразует данные в мел спектрограммы.
При обучении вместо предсказаний о длительности и высоте звука используются реальные данные. Модель оптимизирует среднеквадратичную ошибку (MSE):

$$\mathcal{L} = \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|_2^2 + \alpha\|\hat{\boldsymbol{p}} - \boldsymbol{p}\|_2^2 + \gamma\|\hat{\boldsymbol{d}} - \boldsymbol{d}\|_2^2.$$

, где

$$\hat{\boldsymbol{d}} = \text{DurationPredictor}(\boldsymbol{h}), \quad \hat{\boldsymbol{p}} = \text{PitchPredictor}(\boldsymbol{h}),$$

$$\boldsymbol{g} = \boldsymbol{h} + \text{PitchEmbedding}(\boldsymbol{p})$$
$$\hat{\boldsymbol{y}} = \text{FFTr}([\underbrace{g_1, \ldots, g_1}_{d_1}, \ldots \underbrace{g_n, \ldots, g_n}_{d_n}]).$$

Модель является параллельной, позволяет быстро синтезировать высокоточные мелкомасштабные спектрограммы с высокой степенью контроля над параметрами речи. Модель предоставляет возможность для регуляции темпа, выразительности, высоты тона и интонации, что позволяет сделать синтезированную речь более естественной.

Реализация проекта

В качестве датасета для обучения использовался LJSpeech-1.1. На нем обучалась акустика и вокодер. Вокодер прошел обе стадии обучения (т. е. без и вместе с акустики). Все модели упакованы в докеризованные микросервисы.

В качестве стека использовались:
Python 3.8
Pytorch разных версий в зависимости от сервиса
Docker
FastAPI
Flask
NeMo

Все сервисы работают внутри одной одной закрытой сети. С внешним миром общается только Telegram Bot Service, активно обращаясь к серверам Telegram.

Telegram Bot service
Пример фронтенда, который можно подключить к системе. Он контролирует телеграм бота и форматирует ответы tts service под мессенджер.

Бот поддерживает 2 режима:

- some text

Бот отвечает аудиосообщением с синтезированной речью. Этот запрос может отправить любой пользователь.

Если речь по каким-то причинам не удалось синтезировать присылает текстовое сообщение с ошибкой.

- /analytics

Бот присылает статистику по системе, включая:

- uptime системы

- квантили времен ответов сервиса в зависимости от длины текста за время с последнего перезапуска

- текущую конфигурацию системы.

- графики времен ответа в зависимости от длины текста

Этот запрос могут отправить только пользователи с идентификаторами, зашитыми в конфиг сервиса.

Dispatcher Service
Единая точка входа в систему.

Отвечает на 2 запроса:

- GET /analytics

Выдает JSON с метриками, собранными за uptime сервиса, и конфигурацией. См. Telegram Bot

service для уточнений.

- GET /tts?text=some text

Выдает аудиофайл с синтезированной речью или сообщением об ошибке, если речь не удалось синтезировать.

Для ответа на запрос обращается к остальным сервисам. Первым делом он ищет в кэше была ли уже синтезирована речь для такого же текста с такими же конфигурациями сервиса.

Normalization Service
- запрос с параметром text

Возвращает JSON с параметром text, в котором лежит нормализованный текст. Пример, "at the 13th min" превращается в "at the thirteenth minute".

Нормализует семиотические классы, представленные в NVIDIA NeMo Text Normalization

Acoustic Service
- запрос с параметром text, в котором лежит текст для синтеза.

Возвращает Mel-спектрограмму.

Vocoder Service
- запрос с параметром mel, в котором хранится спектрограмма.

Возвращает аудиофайл.

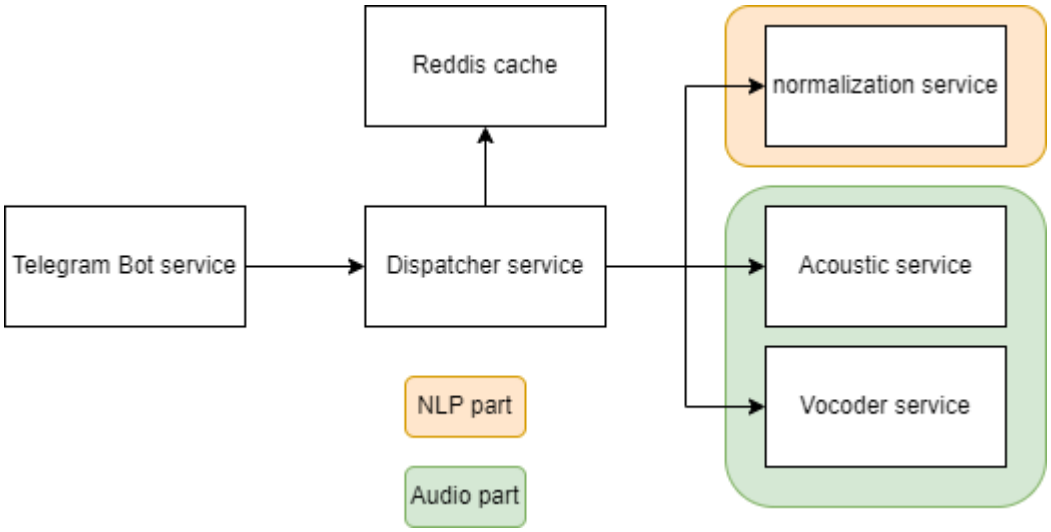
Конкретные модели, их классы и пайплайны были выбраны на основе опыта разработки tts для русского языка в Тинькофф Банке и их качества (часто side by side), указанного в статьях.

Я занимался вокодером, Vocoder Service, Reddis cache, Dispatcher Service, реализовывал аналитику. Также в процессе работы над проектом я делал доклад про работу вокодера. В ходе работы я дополнительно пытался добавить другой вокодер - UnivNet, но это не получилось сделать быстро и мы решили не заниматься этим.

Состав команды курсового проекта: Д. И. Калинина, А. Е. Ишутин, Д. О. Соколова.

Исполнитель:	Часть проекта, за которую отвечает:
Д. И. Калинина	Исследование существующих датасетов на подготовительном этапе, анализ существующих моделей на этапе подготовки, Normalization Service.
Д. О. Соколова	Разработка Telegram Bot service, Acoustic Service, построение Docker compose, разработка программной документации.
А. Е. Ишутин	Разработка архитектуры программного продукта, настройка сервера, TTS Service, Vocoder Service, cash.

Архитектура



Код доступен по ссылке: https://github.com/AIshutin/hse-tts-backend

	1	2	3	4	5	6	7	8	9	10
mean	877,85	2099,8	2099,8	2831,1	3346,9	3972,2	4438,7	5063,5	6611,4	6744,6
max	1217,4	2929,8	2929,8	3910	4843,4	5724,4	7279,6	8299,6	13569	10828
min	685,87	839,81	839,81	898,12	889,32	960,01	858,22	835,59	2300,7	1046,1

В качестве метрики в синтезе речи обычно используют MOS (mean opinion score). Это средняя оценка разметчиков по какой-то заранее оговоренной шкале. В данном случае от 1 до 5 с половинчатыми значениями, где:
1 - плохо, речь вообще не естественна

2 - неудовлетворительно, речь по большей части не естественна

3 - удовлетворительно, речь в равной степени естественна и не естественна.

4 - хорошо, речь естественна по большей части

5 - превосходно, речь неотличима от человеческой

Ground truth получил скор 4. Наша итоговая модель 3.68. Всего участвовало 83 разметчика с перекрытием 5. Эксперимент проводился на платформе Толока.

Заключение

В ходе командной проектной работы, несмотря на уход из проекта части команды, основные задачи были выполнены. Реализован сервис синтеза речи с интерфейсом в телеграмме со средней скоростью синтеза быстрее realtime. Главные этапы в конвейере синтеза речи сделаны такие как нормализация по семантическим классам, акустическая модель и вокодер.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. https://www.youtube.com/watch?v=L7gaINGdUoQ&t=791s
2. https://www.youtube.com/watch?v=N6DnPC0rwIo
3. https://arxiv.org/abs/2006.04558
4. https://arxiv.org/abs/2010.05646
5. https://arxiv.org/pdf/2012.09408v2.pdf
6. https://habr.com/ru/post/135087/
7. https://dl.acm.org/doi/abs/10.1162/coli_a_00349
8. https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/main/
9. https://github.com/jik876/hifi-gan
10. https://arxiv.org/abs/1703.10135
11. https://arxiv.org/abs/2103.14574

Приложение А.

Альтернативное объяснение работы нейронных моделей: