**«Санкт-Петербургский государственный электротехнический университет**

**«ЛЭТИ» им. В.И.Ульянова (Ленина)»**

**(СПбГЭТУ «ЛЭТИ»)**

|  |  |  |
| --- | --- | --- |
| **Направление** | [01.03.02 Прикладная математика и информатика](http://www.eltech.ru/ru/abiturientam/napravleniya-podgotovki/bakalavriat/01040062-prikladnaya-matematika-i-informatika) | |
| **Профиль** | Математическое обеспечение программно-информационных систем | |
| **Факультет** | КТИ | |
| **Кафедра** | МО ЭВМ | |
| *К защите допустить* |  | |
| Зав. кафедрой |  | А. А. Лисс |

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

БАКАЛАВРА

Тема: Реализация диффузионного декодера для задач распознавания речи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  |  |  | Е. А. Павлов |
|  |  | *подпись* |  |  |
| Руководитель |  |  |  | А.А. Лисс |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
| Консультанты |  |  |  | А. А. Панкратов |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
|  |  |  |  | О. С. Артамонова |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Санкт-Петербург

2024

**ЗАДАНИЕ**

**на выпускную квалификационную работу**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой МО ЭВМ |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ А.А. Лисс |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студент | Павлов Е. А. | | | |  | Группа | 0381 | |
| Тема работы: Реализация диффузионного декодера для задач распознавания речи | | | | | | | | |
| Место выполнения ВКР: СПбГЭТУ «ЛЭТИ», кафедра МО ЭВМ | | | | | | | | |
| Исходные данные (технические требования):  Проверить гипотезу о применимости текстовой диффузии в задаче распознавания речи | | | | | | | | |
| Содержание ВКР:  Введение, Актуальность, Обзор аналогов, Теоретическая часть, Практическая часть, Заключение | | | | | | | | |
| Перечень отчетных материалов: пояснительная записка, иллюстративный материал | | | | | | | | |
| Дополнительные разделы: Обеспечение качества разработки, продукции, программного продукта. | | | | | | | | |
|  | | | | | | | | |
| Дата выдачи задания | | | Дата представления ВКР к защите | | | | | |
| «\_02\_»\_\_апреля\_\_\_\_\_2024\_ г. | | | «19»\_\_июня\_\_\_2024\_ г. | | | | | |
|  | | |  | | | | | |
| Студент | |  | | Е. А. Павлов | | | |  |
| Руководитель к.т.н., доцент | |  | | А.А. Лисс | | | |  |
| *(Уч. степень, уч. звание)* | |  | |  | | | |  |
| Консультант | |  | | А. А. Панкратов | | | |  |
| *(Уч. степень, уч. звание)* | |  | |  | | | |  |

**календарный план выполнения**

**выпускной квалификационной работы**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой МО ЭВМ |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ А.А. Лисс |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024\_ г. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | Павлов Е. А. |  | Группа | **0381** |
| Тема работы: Реализация диффузионного декодера для задач распознавания речи | | | | |

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование работ | Срок выполнения |
| 1 | Обзор литературы по теме работы | 25.03 – 01.04 |
| 2 | Обзор предметной области | 01.04 – 08.04 |
| 3 | Подготовка данных | 08.04 – 15.04 |
| 4 | Обучение автоэнкодера | 15.04 – 03.05 |
| 5 | Обучение диффузионного декодера | 03.05 – 21.05 |
| 6 | Обеспечение качества разработки, продукции, программного продукта | 21.05 – 24.05 |
| 7 | Оформление пояснительной записки | 24.05 – 25.05 |
| 8 | Оформление иллюстративного материала | 25.05 – 26.05 |

|  |  |  |
| --- | --- | --- |
| Студент |  | Е. А. Павлов |
| Руководитель к.т.н., доцент |  | А.А. Лисс |
| *(Уч. степень, уч. звание)* |  |  |
| Консультант |  | А. А. Панкратов |
| *(Уч. степень, уч. звание)* |  |  |

**РЕФЕРАТ**

Пояснительная записка 50 стр., 12 рис., 2 табл., 26 ист.

РАСПОЗНАВАНИЕ РЕЧИ, ТЕКСТОВАЯ ДИФФУЗИЯ, АВТОКОДИРОВЩИК, ИСКУСТВЕННЫЕ НЕЙРОННЫЕ СЕТИ, ГЛУБОКОЕ ОБУЧЕНИЕ

**Объектом исследования** является диффузионные нейросетевые модели распознавания речи

**Предметом исследования** является применимость нейросетевых алгоритмов, основанных на диффузии в непрерывных пространствах, в распознавании речи

**Цель работы:** Исследовать потенциал использования диффузионных моделей в задаче распознавания речи

В ходе выполнения данной работы были проведены исследования применимости диффузионных моделей в задачах распознавания речи, проведены эксперименты по обучению искуственных нейронных сетей и выполнена оценка качества распознавания

**ABSTRACT**

In the course of this work, studies were conducted on the applicability of diffusion models in speech recognition problems, experiments were conducted on training artificial neural networks, and an assessment of the quality of recognition was performed.

**СОДЕРЖАНИЕ**

Оглавнение

[ВВЕДЕНИЕ 12](#__RefHeading___Toc879_1863243606)

[1. Актуальность 13](#__RefHeading___Toc881_1863243606)

[1.1 Применение систем распознавания речи в современном мире 13](#__RefHeading___Toc883_1863243606)

[1.2 Обоснование использования диффузионных моделей в задаче распознавания речи 15](#__RefHeading___Toc885_1863243606)

[2. Обзор аналогов 16](#__RefHeading___Toc887_1863243606)

[2.1 CTC — подходы 16](#__RefHeading___Toc889_1863243606)

[2.2 TransFusion 19](#__RefHeading___Toc891_1863243606)

[2.2.1 Мультиноминальная диффузия 20](#__RefHeading___Toc893_1863243606)

[2.2.2 Получение векторных представлений аудио 20](#__RefHeading___Toc1354_1042567274)

[2.2.3 Архитектура модели 21](#__RefHeading___Toc895_1863243606)

[2.2.4 Обучение модели 21](#__RefHeading___Toc897_1863243606)

[3. Теоретическая часть 23](#__RefHeading___Toc899_1863243606)

[3.1 Обзор предметной области 23](#__RefHeading___Toc901_1863243606)

[3.1.1 Формальная постановка задачи 23](#__RefHeading___Toc903_1863243606)

[3.1.2 Метрики качества 23](#__RefHeading___Toc905_1863243606)

[3.1.3 Предобработка аудиоданных 24](#__RefHeading___Toc907_1863243606)

[3.2 Спектральные аугментации (SpecAug) 27](#__RefHeading___Toc1671_2315343878)

[3.3 Диффузионные модели 27](#__RefHeading___Toc909_1863243606)

[3.3.1 Прямой процесс диффузии 28](#__RefHeading___Toc911_1863243606)

[3.3.2 Обратный процесс диффузии 28](#__RefHeading___Toc913_1863243606)

[3.4 Архитектура трансформер 29](#__RefHeading___Toc1673_2315343878)

[3.5 Автокодировщик 31](#__RefHeading___Toc915_1863243606)

[3.6 Latent Diffusion for Language Generation 32](#__RefHeading___Toc917_1863243606)

[3.6.1 Обучение автокодировщика 33](#__RefHeading___Toc1356_1042567274)

[3.6.2 Обучение диффузионной модели 34](#__RefHeading___Toc921_1863243606)

[3.7 wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations 35](#__RefHeading___Toc923_1863243606)

[3.7.1 Архитектура 36](#__RefHeading___Toc1358_1042567274)

[3.7.1 Обучение 37](#__RefHeading___Toc1360_1042567274)

[3.8 Предлагаемый подход 38](#__RefHeading___Toc1362_1042567274)

[3.8.1 Обучение автокодировщика 38](#__RefHeading___Toc1323_1605817446)

[3.8.2 Получение эмбеддингов для аудио 38](#__RefHeading___Toc1325_1605817446)

[3.8.3 Обучение диффузионной модели 38](#__RefHeading___Toc1327_1605817446)

[4. Практическая часть 40](#__RefHeading___Toc1364_1042567274)

[4.1 Используемые инструменты и библиотеки 40](#__RefHeading___Toc1366_1042567274)

[4.2 Описание данных 40](#__RefHeading___Toc1368_1042567274)

[4.3 Описание экспериментов 41](#__RefHeading___Toc1370_1042567274)

[4.3.1 Обучение автокодировщика текста 41](#__RefHeading___Toc935_1863243606)

[4.3.2 Получение векторных представлений аудио 42](#__RefHeading___Toc2443_1605817446)

[4.3.3 Обучение предсказателя шума 42](#__RefHeading___Toc2445_1605817446)

[4.3.4 Оценка и сравнения полученной модели 43](#__RefHeading___Toc2447_1605817446)

[5. Обеспечение качества разработки, продукции, программного продукта 44](#__RefHeading___Toc941_1863243606)

[ЗАКЛЮЧЕНИЕ 45](#__RefHeading___Toc943_1863243606)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 46](#__RefHeading___Toc945_1863243606)

[ПРИЛОЖЕНИЕ А 49](#__RefHeading___Toc947_1863243606)

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

Токен – единица текста, обычно представляет собой несколько букв или слово.

Токенизация – процесс преобразования исходного текст в последовательность токенов.

Референсная транскрипция – истинная транскрипция аудиофрагмента с речью, чаще всего получена с помощью людей-разметчиков

Искуственная нейронная сеть (нейросеть) – математическая модель, в некотором ходе схожая с нейронными сетями живых организмов.

Диффузионные модели — это класс моделей скрытых переменных, которые обучаются итеративно преобразовывать случайный Гауссовский шум, который может быть сгенерирован аналитически, в наблюдения из данных с неизвестным распределением [8].

Автокодировщик – класс искуственных нейронных сетей, который используется для получения оптимальных векторных представлений каких-либо данных [4]

Лосс функция (функция потерь) – дифференцируемая функция, которая даёт некоторую оценку на разницу между истинным ответом и предсказанием модели

Эмбеддинг — векторое представление, полученное с помощью нейросети, чаще всего представляет собой промежуточное значение с какого-либо слоя

Нейросети прямого распространения — один из простейших видов нейростей, состоит из матричных умножений и нелинейных функций активации

Мультиголовый механизм самовнимания (multi-head self-attention) — один из типов слоёв искуственных нейронных сетей, предложенный в статье Attention is all you need [9]

Леернорм (Layernorm) — один из типов слоёв искуственных нейронных сетей, выполняет нормировку входных данных по оси признаков, содержит два обучаемых параметра — разброс и смещение [24]

Датасет — набор размеченных данных

Тренировочный датасет (трейнсет) — датасет, используемый на этапе обучения модели

Валидационный датасет (валсет) — датасет, используемый на этапе оценки модели, примеры из него строго не должно содержаться в тренировочном датасете

Функция активации — нелинейная функция, используемая между линейными и квазилинейный преобразованиями в нейронных сетях с целью улучшения обощающей способности

Self-supervised — способ обучения нейронной сети, в условиях малого количества размеченных данных, где сначала модель обучается на имеющихся размеченных данных, после чего используется для генерации разметки на неразмеченных данных, после чего полученная разметка используется для обучения следующей модели и так далее

Свёрточный слой — один из типов слоёв искусственных нейронных сетей, вычисляет операцию свёртки между входными данными и фильтром, фильтр является обучаемым параметром

Трансформер — архитектура искусственной нейронной сети, предложенная в статье [9], построенная на использовании multi-head self-attention

N-грамма — некоторое множество грамматических единиц текста, например, букв, слов или токенов

Кодбук — заранее заданное множество векторов, чаще всего ортогональных, иногда кодбуки бывают обучаемыми

Батч — множество объектов, представленных в виде матрицы или тензора большей размерности, используемых на одном прогоне модели

Кросс-аттеншен — один из типов слоёв искусственых нейронных сетей, где механизм самовнимание получает в качестве key и query векторов данные из различных последовательностей

Аугментации — различные незначительные преобразование входных данных, помогающие разнообразить обучающий датасет модели и увеличить её обощающую способность

Позиционные эмбеддинги — специальная добавка ко входным данным модели, которая кодирует информацию о позиции элемента в последовательности

## ВВЕДЕНИЕ

В последнее время резко возросла потребность в распознавании речи в большом количестве сценариев. Задача перевод речи в текст возникает в системах видеоконференцсвязи, системах голосового управление и многих других. Также в последнее время большое развитие получили нейросетевые подходы к задаче распознавания, которые показывают лучшие результаты среди всех алгоритмов-конкурентов. Одним из новых типов искуственных нейронных сетей являются диффузионные модели, хорошо показавшие себя в нескольких классах задач. Всё это приводит к потребности в исследовании диффузионных моделей в задаче распознавания речи

Цель работы: Исследовать потенциал использования диффузионных моделей в задаче распознавания речи

**Объектом исследования** является диффузионные нейросетевые модели распознавания речи

**Предметом исследования** является применимость нейросетевых алгоритмов, основанных на диффузии в непрерывных пространствах, в распознавании речи

Для достижения поставленной цели требуется решить следующие задачи:

1. Анализ существующих решений

2. Выбор используемых методов

3. Реализация и обучения диффузионной модели

4. Оценка качества и сравнение с аналогами

## 1. Актуальность

В современном мире большое количество информации представлено в аудиоформате и количество такой информации продолжает расти с каждым днём. Источниками такой информации могут послужлить системы телефонии, видеосвязи, стриминговые сервисы, а также множество других.

С ростом количества аудиоданных усложняется работа с ними на многих уровнях. Начиная со скорости восприятия информации — для человека намного быстрее прочитать транскрипцию, чем прослушать аудио — заканчивая более сложными задачами, по типу текстового поиска и составления краткого содержания. Всё это приводит к увеличению потребности в продвинутых системах распознавания речи.

В настоящее время уже проведен ряд исследований в этой области, а также разработаны работоспособные системы. Однако ошибка таких систем не равна нулю, что вызывает потребность в проведении дальнейших исследований в этой области. Основным подходом к задаче распознавания речи является использования машинного обучения, а конкретно — глубокого обучения.

Глубокое обучение — раздел машинного обучения, занимающийся изучением и применением многослойных нейронных сетей. В последние годы эта область человеческого знания активно развивается, предлагаются новые подходы и методы по созданию нейросетевых алгоритмов. Такие алгоритмы хорошо показывают себя на большом классе задач — распознавание речи, сегментации изображений, генерации текста и фото.

Одним из последних достижений науки являются диффузионные модели, хорошо показавшие себя в задачах генерации изображений и речи [1], [2].

### 1.1 Применение систем распознавания речи в современном мире

Как было сказано ранее, количество аудио контента, производимого человечеством позволяет расти, а работа с его содержанием путём прослушиванию вызывает всё больше трудностей. Также распознавать речь требуется не только в задачах, связанных с обработкой или поиском информации человеком, но и в задачах создания голосовых интерфейсов управления. Ниже представлен список возможных применений систем распознавания речи:

1. Видеоконференции

В ходе пандемии COVID-19, а также нескольких последующих лет сильно возрасло количество ВКС, а также их аудитория. Такие системы используются в большом количестве организаций — корпорациях, образовательных и государственных учреждениях, сотрудники пользуются видеосвязью для проведения совещаний, лекций, планирований и иных встреч. Часто, в результате таких встреч требуется составить протокол встречи, чтобы после можно было вернуться к нему и получить доступ к информацию с прошедей встречи.

2. Телефония

В большом количестве компанию существуют отделы продаж и технической поддержки, активно использующие в своей работе телефонную связь. Информация о содержании разговоров сотрудников может потребоваться для улучшения качество облуживания клиентов и проведения речевой аналитики.

3. Переводчики

Ещё одним применением распознавания речи могут служить системы перевод. Большинство современных систем перевода работают на уровне текста, поэтому в сценарии, когда пользователь переводчика не знает языка и имеет доступ только к аудио информации, системы распознавания могут помочь ему с перевод аудио в текст для дальнейшего перевода на другой язык

4. Голосовые ассистенты

Также распознавание речи может быть необходимо в голосовых ассистенах для понимания команд и запросов пользователя, озвученных голосом

### 1.2 Обоснование использования диффузионных моделей в задаче распознавания речи

За небольшое количество времени диффузионные модели показали впечатляющие результаты в работе с изображениями и синтезе речи, а также нашли применение в генерации текста. Это позволяет ожидать, что и из их применения в задаче распознавания речи получится извечь выгоду

## 2. Обзор аналогов

В этом разделе будут рассмотрены существующие подходы к задаче распознавания речи. Перед тем, как перейти к рассмотрению аналогов опишем задачу распознавания речи формально — на вход поступает цифровой аудиосигнал, а результатом является текст на каком-либо языке. Таким образом:

X — входной вектор размером T, где T — количество дискретных измерений

L — референсная транскрипция для аудиозаписи X

Задача состоит в том, чтобы найти отображение из пространства X в пространство L. Так как данная задача не имеет аналитического решения в общем виде принято использовать различные модели, которые стараются оценить и аппроксиморовать данное отображение.

### 2.1 CTC — подходы

Connectionist Temporal Classification (CTC) — метод при обучении нейронных сетей на задачу матчинга последовательностей без знания точного временного соответсвия между элементами последовательности. Обычно данный метод используется в задаче распознавания речи или задачи распознавания рукописного текста. Метод был предложен в 2006 году Алексом Грейсвом и его коллегами [22].

Для описания метода введём обозначения:

x — входная последовательность из векторов размерности m

T — длина последовательности x

L — алфавит из элементов которого состоит выходная последовательность

y — выходная последовательность, состоящая из элементов L

бланк (<blank>) — специально добавляемый символ алфавита, используется на декодировании

L` — алфавит L, к которому добавили бланк

N — мощность алфавита L`

Алгоритм получаем на вход один вектор из m значений, а на выход выдаёт вероятностное распределение над алфавитом L`.

В ходе обработки всей последовательности x, получается T вероятностых распределений над L`, обозначим их как матрицу A размера N на T.

Введём понятие пути. Путь — это последовательность элементов матрицы A, которая однозначно определяет выходную последовательность z. Для такого пути (обозначием его l) можно построить оценку его вероятности, согласно следующей формуле:

, где

i — номер элемента выходной последовательности

l(i) — индекс в алфавите элемента, находящегося на позиции i в выходной последовательности

P(y|x) — вероятность получить выходную последовательность x, при входной последовательности y.

На этапе обучения нейронной сети есть набор (D) размеченных пар (x,y) — в данном случае аудио и речь. Тогда на обучении алгоритм будет по последовательности x генерировать матрицу вероятностей A, согласно которой будут рассчитываться оценки вероятностей для референсных транскрипций. На основе этих оценок будем считать следующую лосс-функцию:

Которую будем оптимизировать с помощью изменения параметров модели.

Однако при таком подходе есть проблема с распознаванием идущих подряд одинаковых элементов выходной последовательности. Допустим, есть два соседних, с точки зрения времени, распределения, максимальную вероятность в которых имеет один и тот же элемент. В такой ситуации нельзя разделить ситуации — один элемент на несколько временных отрезков или это два одинаковых элемента на соседних временных отрезках. В задаче распознавания речи это наглядно видно в словах с повторяющимися буквами. Например, слово «стресс».

Для разделения таких случаев и вводится бланк символ. Он вставляется между одинаковыми элементами алфавита и означает переход к следующему символу. Тогда последовательности вида: {с, т, р, е, с, <blank>, с} будут переводится в «стресс», а последовательности вида {с, т, р, е, с, с}, будут приводится к виду «стресс». На представленном ниже рис. 1 можно видеть пример работы ctc:

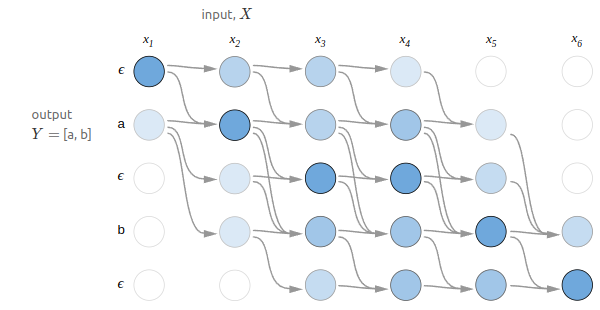


Рисунок 1 – Пример ctc на этапе получения выходной последовательности. По вертикали расположены элементы выходной последовательности, а по горизонтали входной. Глубина цвета означает вероятность, эпсилон — это бланк-символ

Примером модели, обученной с помощью ctc может служить нейросеть с архитектурой Conformer [19].

Conformer — одна из первых удачных попыток совмещения свёртночных нейронных сетей и механизма самовнимания в задаче распознавания речи. Архитектура Conformer представлена на рис. 2.

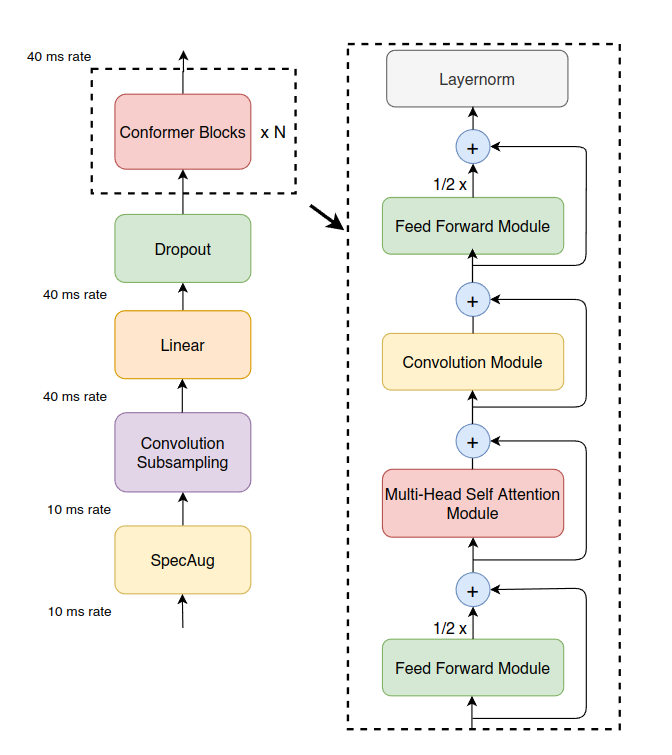


Рисунок 2 – Архитектура Conformer

В начале модели идёт блок «Convolution Subsampling», который сжимает размер входной последовательности в 4 раза, что уменьшает требуемое количество вычислений.

Далее следует последовательность из «Conformer Block»-ов - основного модуля этой архитектуры, состоящего из свёрток, сетей прямого распространения и механизма самовнимания, а также леернормов. Визуализацию «Conformer Block» можно видеть на рис. 2.

### 2.2 TransFusion

Это диффузионная модель для распознавания речи, предложенная в статье [3]. Основная идея — обучить модель текстовой диффузии, у которой диффузионный процесс может быть обусловлен на вектороное представление аудио, содержащего человеческую речь.

#### 2.2.1 Мультиноминальная диффузия

Обычно диффузионные модели работают с вещественными представлениями данные, это могут быть эмбеддинги, картинки или спектрограммы. Однако текст имеет дискретную природу, поэтому авторы используют мультиноминальную диффузию предложенную в статье [15].

Конкретно мультиномильная диффузия определяется входы диффузионной модели, как последовательность дискретных элементов (символов или слов), представленных в виде векторов содержащих одну единицу, а остальное заполнено нулями.

Формальные обозначения:

- индекс временного шага, T — количество временных шагов

- индекс позиции в последовательности

- вектора, представляющие элементы последовательности, размерность векторов совпадает с мощностью алфавита символов

Мультиноминальная диффузия определяет прямой процесс диффузии следующим образом:

,

C обозначает категориальное распределение, - коэффициент расписания добавления шума,

Таким образом предиктор шума предсказывает на каждой итерации диффузии распределение над словарём

#### 2.2.2 Получение векторных представлений аудио

Для получения аудиоэмбеддингов авторы используют предобученную модель WavLM [16]. Данная модель обучалась на огромном количестве данных в self-supervised режиме, что позволило достичь высокой обобщающей способности внутренних представлений модели.

Во время обучения предсказателя шума авторы не изменяют параметры WavLM, эмбеддинги строятся для каждых 20 миллисекунд исходного аудио. Модель содержит порядка 317 миллионов параметров и обучалась на данных объёмом порядка 94 тысяч часов.

#### 2.2.3 Архитектура модели

В качестве предсказателя шума авторы используют архитектуру нейросети на основе трансформера. Для обуславливания процесса диффузии на содержаниие аудио авторы добавляют небольшой блок состоящий из линейных проекций, нормировок и нелиненых функций активации. Всего сеть содержит 24 трансформер-блока и имеет 253 миллиона параметров, архитекутра представлена на рис. 3.

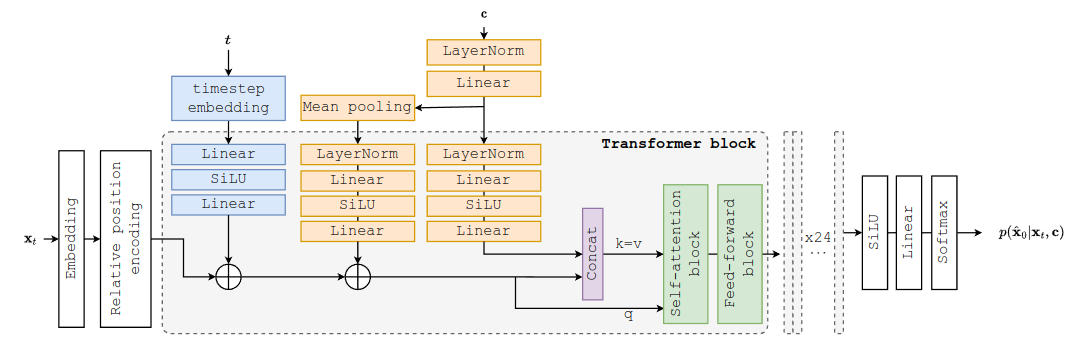


Рисунок 3 – Архитектура предсказателя шума

#### 2.2.4 Обучение модели

Лосс функция, используемая авторами также берётся из статьи [15]. Так как основная цель при обучении диффузионной модель — это построение модели, которая будет предсказывать максимально близкое к апостериорному распределению , то в качестве лосс-функции используется дивергенция Кульбака-Лейблера:

При обучении используется 200 временных шагов диффузии, аудиоэмбеддинги из WavLM-large, используется оптимизатор AdamW [25] со значениями бет (0.9, 0.999) и с константным расписанием обучающих коэффициентов с разогревом 10000 итераций и пиковым значением . В качестве обучающего датасета используется LibriSpeech-960h, аугментации не используются.

## 3. Теоретическая часть

### 3.1 Обзор предметной области

В данном разделе описан ряд понятий, используемых в большом количестве работ, а также при описании предлагаемого в данной работе решения.

#### 3.1.1 Формальная постановка задачи

На вход поступает аудиосигнал, содержащий один канал, который может быть представлен в виде вещественных или натуральных чисел. В общем случае аудио сигнал представляет собой вектор из , где N — количество отсчётов, полученных при дискретизации аналогового сигнала.

На выход поступает последовательность из элементов из заранее известного конечного алфавита.

#### 3.1.2 Метрики качества

В задаче распознавания речи сущесвует огромное количество метрик, который пытаются оценивать те или иные аспекты качества. В данной работе будет использоваться метрика WER (Word Error Rate), основанная на расстоянии Левенштейна.

, где S — количество замен, D — количество ударений, I — количество вставок, а N — количество слов во референсной транскрипции

В задаче оценки генерации текста с помощью автокодировщика используются метрики Rouge-1, Rouge-L и BLEU.

Расчёт Rouge-1 производится, согласно следующим формулам:

, где в числителе — количество общих юниграмм в референсном тексте и в полученном, а в знаменателе — количество юниграмм в референсном тексте

, где в числителе — количество общих юниграмм в референсном тексте и в полученном, а в знаменателе — количество юниграмм в сгенерированном тексте

Расчёт Rouge-L производится, согласно следующим формулам:

, где ДОП — длиннейшая общая подпоследовательность

Расчёт BLEU [7] производится, согласно следующим формулам:

, где

c — длина сгенерированного текста,

r — «эффективная» длина референсного текста

- вес для данной n-граммы

, где

Count — количестве слов в n-грамме

MaxRefCount — максимальное количество слов в референсной транскрипции

#### 3.1.3 Предобработка аудиоданных

Зачастую, работать напрямую с аудиосигналом неудобно — это приводит обработке очень длинных последовательностей, что усложняет вычисления. Кроме того, представление в виде сырого аудиосигнала может быть не очень инфомативным в задаче распознавания речи.

Одним из самых распространённых представлений аудиосигнала является лог-мел-спектрограмма.

Спектограмма — это визуальное представление зависимости спектральной плотности мощности сигнала от времени, получаемое с помощью оконного преобразования Фурье. Пример такого представления можно видеть на рис. 4.

Формула для расчёта интенстивности:

, где

t — время

w — частота

stft — оконное преобразование Фурье

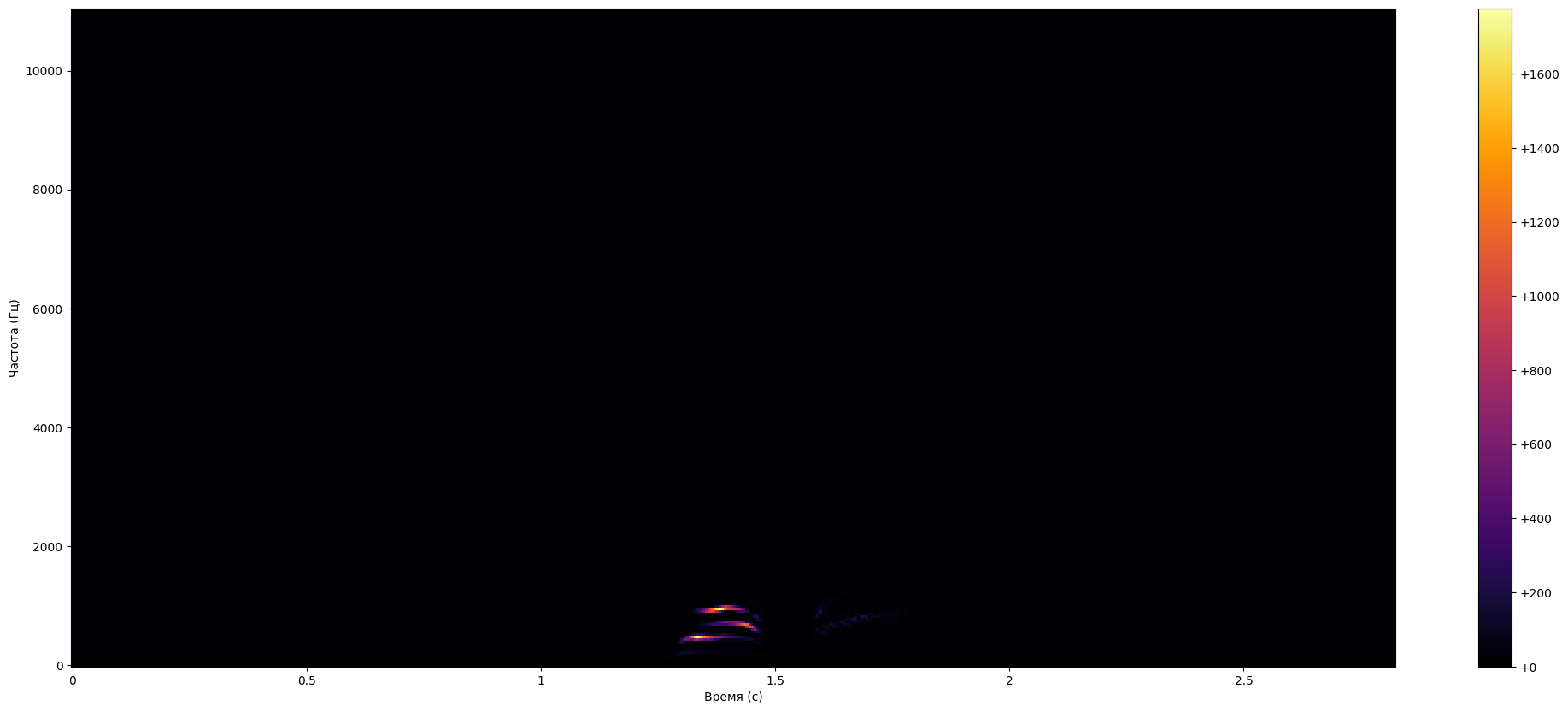


Рисунок 4 – Спектрограмма

Лог-спектрограмма — это спектрограмма, значения интенсивности которой отложены в децибелах. Пример такой спектрограммы можно видеть на рис. 5.

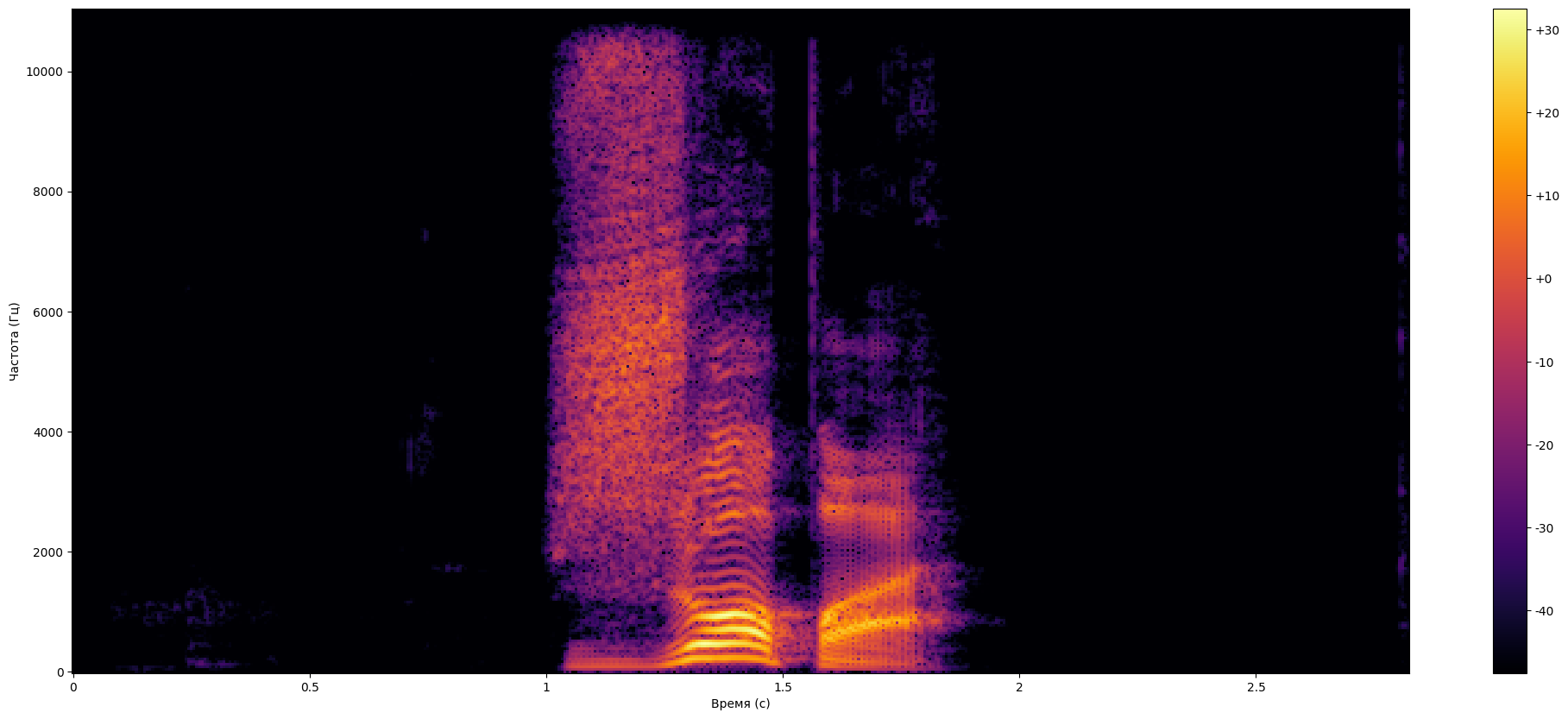


Рисунок 5 – Лог-спектрограмма

Лог-мел-спектрограмма — это лог-спектрограмма, частоты которой преобразованы специальным образом. Мел-шкала создана в связи с тем, что люди нелинейно воспринимают разницу между частотами в разных диапазонах.

Для перевода значений в мел-шкалу используется следующая формула:

Пример такой спектрограммы можно видеть на рис. 6.

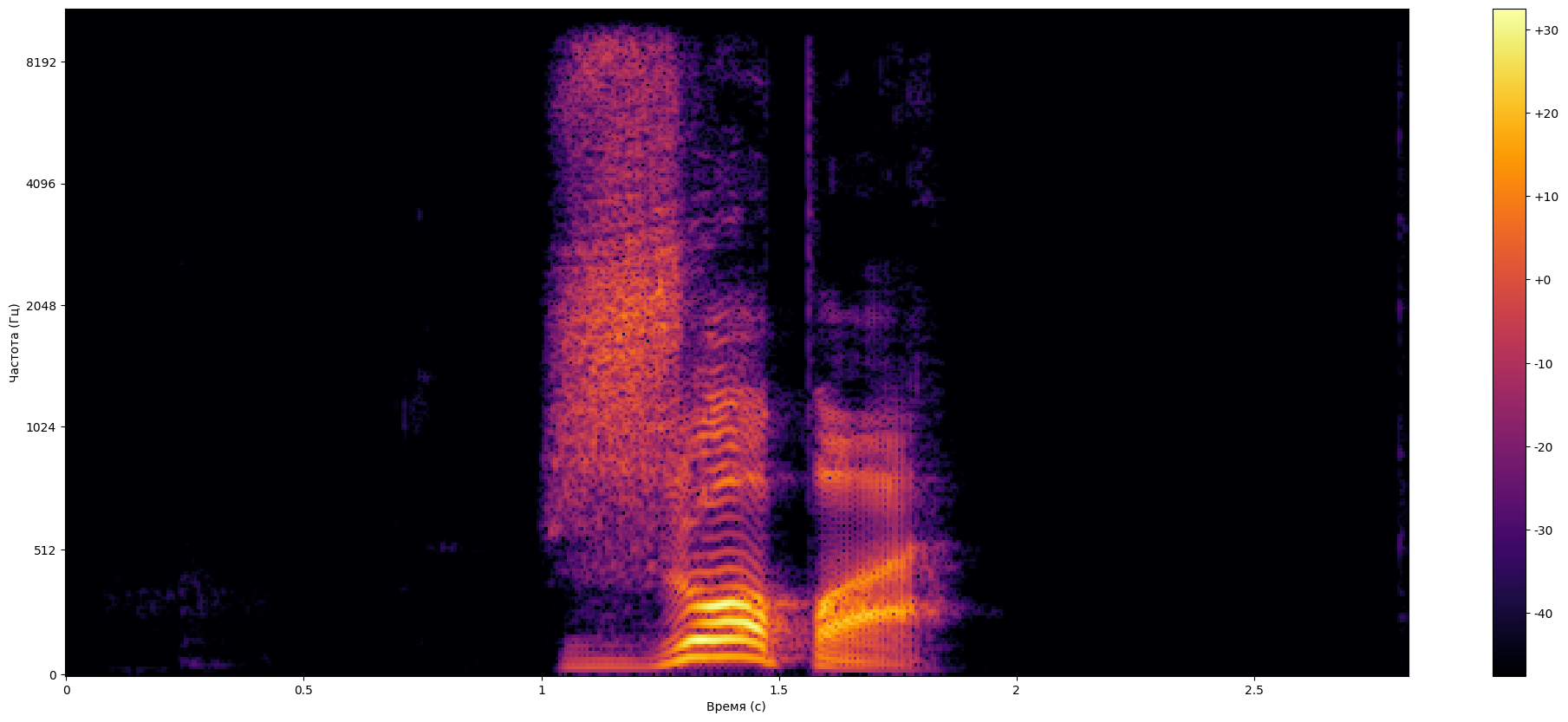


Рисунок 6 – Лог-мел-спектрограмма

### 3.2 Спектральные аугментации (SpecAug)

Одним из самых простых и эффективных способов аугментации спектрограмм являются спектральные аугментации [26].

Пусть у нас есть спектрограмм в границах по оси времени и по оси частоты, тогда спектральная аугментация это процесс в ходе которого выделяется случай отрезов спектрограммы по одной из осей и удаляется.

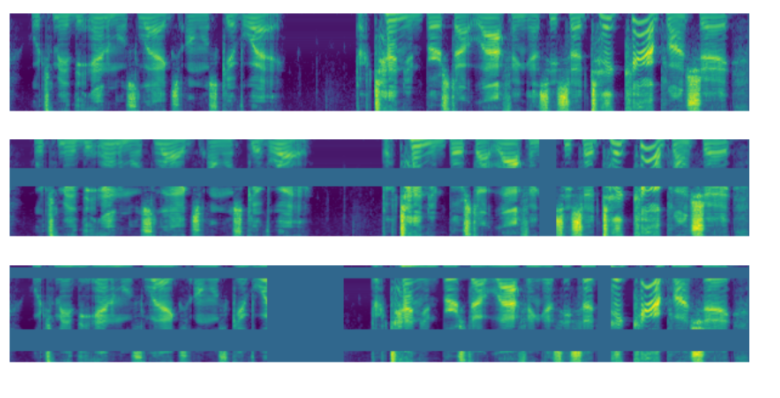


Рисунок 7 – Пример результата применения спектральных аугментаций

На рис. 7 можно чётко видеть «вырезанные» области прямоугольного цвета, которые были заполнены нулями. Аугментации такого типа не позволяют моделям переобучаться на те или иные частоты и делают их более устойчивыми к кратковременным помехам или потерям сигнала.

### 3.3 Диффузионные модели

Диффузионные модели — это класс моделей скрытых переменных, которые обучаются итеративно преобразовывать случайный Гауссовский шум, который может быть сгенерирован аналитически, в наблюдения из данных с неизвестным распределением.

Этот класс моделей был впервые предложен в статье в 2015 году [8]

#### 3.3.1 Прямой процесс диффузии

Прямой процесс диффузии — процесс зашумления исходного наблюдения с помощью добавление нормального шума.

Данный алгоритм получает на вход некое наблюдение из реальных данных - , после чего начинается добавлять небольше количество нормального шума. После повторения такой операции T раз получается последовательносить замушлённых наблюдений . Добавление шума параметризовано с помощью коэффициентов .

В ходе такого процесса исходное наблюдение с каждым шагом теряет всё больше данных и содержит всё меньше информации об исходном объекте. Причём для вычисления зашумлённого x на произвольном шаге не требуется вычислять все промежуточные шаги так как легко можно репараметризовать формулу, введя обозначения и

#### 3.3.2 Обратный процесс диффузии

Обратный процесс диффузии — процесс восстановление исходного наблюдений по его зашумлённому состоянию.

Для построения точной оценки требуется весь набор данных, что является трудновыполнимым условием, поэтому для этой задачи обучается модель , способная аппроксимировать эти условные вероятности и запустить обратный процесс диффузии.

Для этого нужно обучить модель для предсказания

Однако значение известно нам на обучении, поэтому можно предсказывать сразу добавленный шум .

В статье [ссылка на папиру про упрощения лосса] эмпирически обнаружено, что диффузионные модели лучше работают при обучении с лоссом:

### 3.4 Архитектура трансформер

Трансформер — одна из самых популярных архитектур нейронных сетей, основанная на muti-head self-attention, была предложена в статье 2017 года [9].

Трансформер состоит из двух больших блоков, которые идентичны между собой — энкодер и декодер. Разница лишь в аттеншен маска т.е к каким токенам имеется доступ при вычислении multi-head self-attention.

Энкодер и декодер состоят, в свою очередь, из multi-head self-attention и небольших сетей прямого распространения, пример блока изображён на рис. 8.

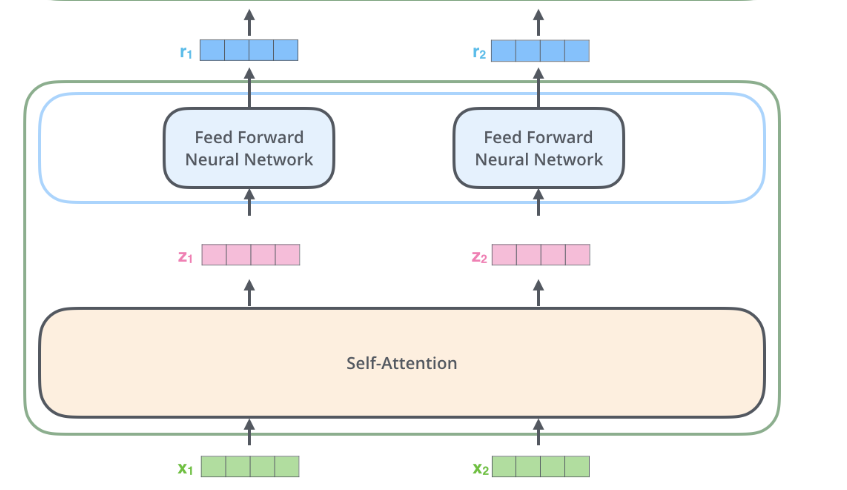


Рисунок 8 – Высокоуровневое представление одного трансформер-блока

Здесь - эмбеддинги элементов последовательности, которую обрабатывает трансформер. На первом слое такие эмбеддинги берутся из специального словаря, который хранит споставления (уникальный элемент, его эмбеддинг), а далее такие эмбеддинги получаются в ходе работыть предыдущих трансформер-блоков.

Внутри multi-head self-attenion есть три матрицы проекций:

* Query (запросы)
* Key (ключи)
* Values (значения)

По входному эмбеддингу вычисляются его query, key и value с помощью этих матриц проекций. Далее query вектора каждого элемента умножаются на каждый элемент key векторов каждого элемента, в результате чего получается некий «вес», который каждый элемент присваивает всем остальным элементам. После эти веса нормируются и используются для взвешенной суммы value векторов, которые становятся новыми эмбеддингами.

В матричном виде эти преобразования можно записать следующим образом:

, где

Q — значения query векторов

K — значения key векторов

V — значения value векторов

d — внутрення размерность модели

Причем таких наборов матриц проекций строится несколько, каждый такой набор называется «головой» self-attention

### 3.5 Автокодировщик

Автокодировщик – класс искуственных нейронных сетей, который используется для получения оптимальных векторных представлений каких-либо данных, предложенный в [4].

Архитектура таких моделей выглядит следующим образом :

Энкодер — нейронная сеть, которая получает на вход данные в исходном пространстве и выполняет ряд преобразований, в результате которых получает новое представление данных в латентном пространстве, размерность которого существенно меньше размерности представления в исходном пространстве

Декодер — нейронная сеть, которая получает на вход векторное представление из латентного пространства и восстанавливает исходный объект в исходном пространстве

Изображение архитектуры в общем виде можно видеть на рис. 9.

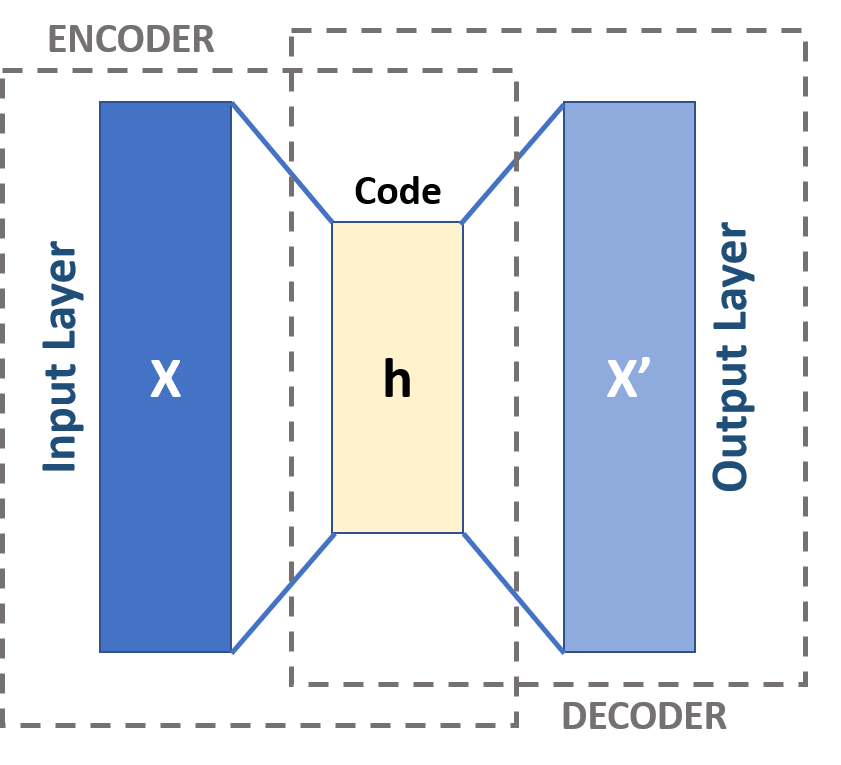


Рисунок 9 – Общий вид архитектуры автокодировщика

Для обучения таких моделей требуется ввести лосс функцию реконструкции, например, среднеквадратичную ошибку между входным вектором и результатом работы декодера то есть результатом восстановления наблюдения из полученного латентного представления.

### 3.6 Latent Diffusion for Language Generation

В этой статье [21] авторы обучают текстовую диффузию для задачи генерации текста. Для этого они используют несколько предобученных sequence-to-sequence моделей (BART и T5). Предлагаемый метод состоит из двух основных частей:

1. Расширяют преодбученную языковую модель для получения автокодировщика текста

2. Предлагают модели непрерывной диффузии, которые обучаются генерировать текста на основе латентного распределения автокодировщиков

Полученный метод представлен на рис. 10.

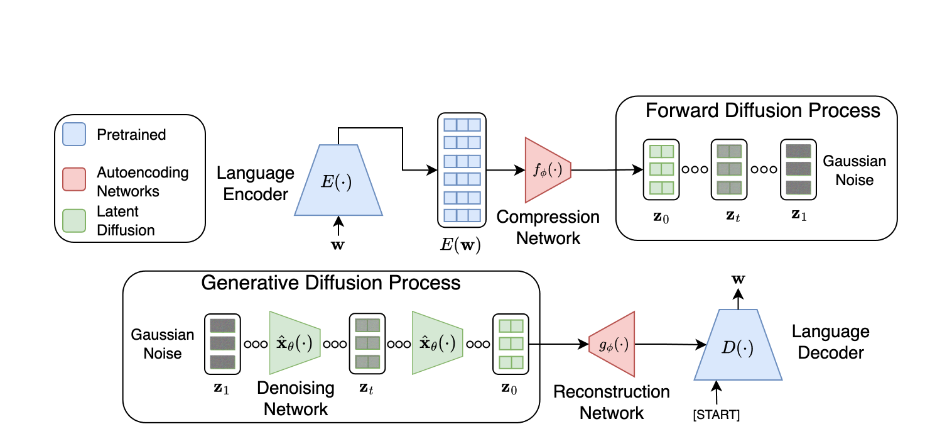


Рисунок 10 – Визуализация архитектуры автокодировщика из статьи Latent Diffusion for Language Generation

### 3.6.1 Обучение автокодировщика

На первом этапе авторы обучает автокодировщик на основе языковой модели, построенной с помощью архитектуру трансформер. Эксперименты проводятся с двумя моделями — BART-base [5], обученной командой исследователей из Facebook AI и FLAN-T5-base [6], обученной командой исследователей из Google.

Обе модели основаны на архитектуре трансформер, а значит содержат, как и большинство нейронных сетей, энкодер и декодер. В обычном сценарии работы векторные представления энкодера передаются в декодер, который, на их основе, генерирует новый текст. Авторы, с целью получения латентного пространства, добавляют между энкодером и декодером две нейросети — нейросеть сжатия и нейросеть реконструкции.

Нейросеть сжатия получает на вход векторные представления данных из энкодера, после чего сжимает их. Один слой нейросети сжатия состоит из двух блоков — мультиголового механизма самовнимания и сети прямого распространения. Архитекутра сети изображена на рис. 11.

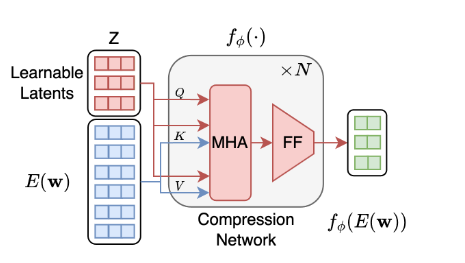


Рисунок 11 – Архитектура нейросети сжатия (compression network). E(w) — векторные представления, полученные из энкодера, Z — обучаемые query вектора. MHA — мультиголовый механизм самовнимания, FF — сеть прямого распространения

Архитектура нейросети реконструкции совпадает с архитектурой с сети сжатия с точность до размерности матриц так как проекция осуществляется в обратную сторону, из латентного пространства в пространство векторных представлений языковой модели

Обучение моделей сжатия и реконструкции происходит по следующему алгоритму:

1) Полученный текст токенизируется

2) Токенизированный текст поступает на вход в энкодер, который генерирует векторные представления

3) Векторное представления поступают в сеть сжатия, где переводятся в латентное пространства

4) Латенты поступают в сеть реконструкции, где переводятся в исходные векторное представления и поступают в декодер

5) Декодер, на основе полученных представлений генерирует вероятностное распределение над токенами

6) Зная исходный текст, рассчитывается лосс-функция от исходного текста и полученных распределений. В качестве лосс-функции используется кросс-энтропия

7) С помощью градиентных метод оптимизации веса нейросетей сжатия и реконструкции обновляются. Веса энкодера и декодера в ходе обучение заморожены то есть не изменяются

#### 3.6.2 Обучение диффузионной модели

Входе первого этапа авторы получают автокодировщик, который умеет генерировать латентные представления для текста фиксированного.

На втором этапе авторы переходят к обучению диффузионной модели. Для набора текстовых данных авторы строят латентные представления, используя полученный автокодировщик. Для этого они обучают предиктор шума для восстановления исходного x, с использованием среднеквадратичной ошибки в качестве лосс-функции:

,

где x — обучающие данные, t — временной шаг, - случайный Гауссовский шум, - означает расписание добавляемого шума, - зависящий от времене коэффициент взвешивания

Для генерации текста авторы генерируют случайный латентный вектор из стандартного нормального распределения и запускают процесс обратной диффузии используют полученный вектор в качестве

При обучении авторы используют косинусное расписание добавляемых коэффициентов:

При генерации авторы используют стандартное DDPM [20] расписание и делают 250 временных шагов.

В качестве архитектуры предиктора шума используют трансформер с леернормами перед мультиголовыми механизмами внимания с 12 слоями и внутренней размерностью эмбеддингов 768.

### 3.7 wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations

В данной работе [23] авторы обучают модель для извлечения эмбеддингов из аудио, содержащих речь, которую затем дообучают на задачу распознавания речи.

#### 3.7.1 Архитектура

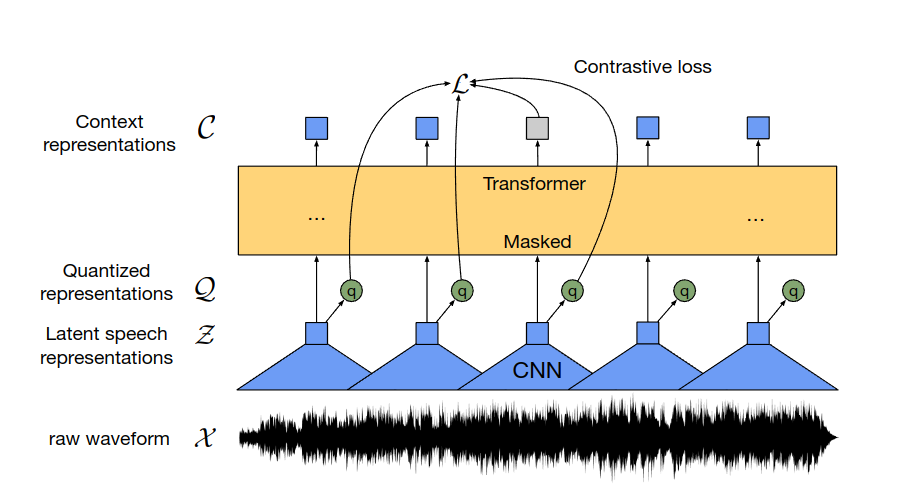


Рисунок 12 – Архитектура wav2vec 2.0

Модель состоит из трёх основных элементов:

1. Фича-энкодер - нейросеть для извлечения первичных признаков из аудио и понижения размерности. Состоит из нескольких свёрточных слоёв по оси времени, а также леернормов между ними и функций активации GELU. Ожидается, что аудио нормализовано и имеет среднее 0 и дисперсию 1.

2. Нейросеть контекста, которая получает эмбеддинги из фича-энкодера и строит более высокоуровневые эмбеддинги. Архитекутра этой нейросети основывается на трансформере, для кодирования позиционной информации используются свёрточные блоки, похожие на [17], после которых идёт GELU и леернорм

3. Модуль квантизации, который дискритизирует выходы фича-энкодера. Для квантизации создаётся G кодбуков с V эмбеддингами. Далее из каждого кодбука берётся по эмббедингу, они объединются и подаются в линейный слой. Для сохранения дифференцируемости этой операции используются софтмакс Гумбеля [18]

Общий вид архитектуры изображён на рис. 12.

#### 3.7.1 Обучение

Фича-энкодер получает на вход исходное аудио X, и на выходе генерирует набор из представление, где T — количество временных отрезков.

Далее представления, полученные из фича-энкодера передаются в нейросеть контекста, которая генерирует новые векторные эмбеддинги и используются далее при расчёте лосс функция.

Для обучения модели в режиме без учителя то есть в отсутствии разметки авторы используют маскирование. Целью обучения является определение правильного квантованного представления звука среди набора таких представлений.

Итоговая лосс функция состоит из двух компонент:

Первая компонента - - отвечает за правильность классификации квантизованного представления:

, где

- множество возможноых квантизованных представлений

- замаскированное квантизованное представление

- векторое представление, полученное из модели

- косинусная близость

Вторая компонента - , так называемая лосс-функция разнообразия и отвечает за увеличения использования представлений из кодбуков. Авторы определяют её как задачу максимазации средней энтропии вероятностного распределения над векторами кодбука:

### 3.8 Предлагаемый подход

Рассмотрев все вышеизложенные публикации можно перейти к описанию подхода, предлагаемого в данной работе. Текстовая диффузия с обуславливанием на аудиоэмбеддинги уже была представлена в работе TransFusion, однако там использовалась мультиноминальная диффузия т.е диффузия на уровне вероятностного распределения. В то время, как последний публикации в областе диффузионных моделей показывают, что намного эффективней использовать диффузионные модели, действиющие в латентных пространствах.

### 3.8.1 Обучение автокодировщика

Для обучения диффузионной модели требуется создать автокодировщик текста, в латентном пространстве которого будет работать модель. Подходящий автокодировщик уже был разобран выше при рассмотрении статьи Latent Diffusion for Language Generation. Осталось лишь обучить такой автокодировщик на домене транскрипций. Основной для обучения автокодировщика послужит модель от компании Facebook AI — BART-base.

### 3.8.2 Получение эмбеддингов для аудио

На данный момент, одной из лучших и популярнейших моделей в области получения эмбеддингов является предобученный wav2vec2.0 от компании Facebook AI, архитектура и принцип работы которого также рассмотрены выше.

В данной работе используется версия wav2vec2.0-large.

#### 3.8.3 Обучение диффузионной модели

В качестве архитекутры предсказателя шума был выбран трансформер с пре-леернормами (то есть нормировка производится перед механизмом внимания), что позволяет улучшить стабильность и сходимость модели. Для построения эмбеддингов временного шага используются синусоидальные абсолютные эмбеддинги:

Для кодирования позиционной информации аудио эмбеддингов используется относительный позиционный кодировщик, основанный на обучаемых свёртках.

В качестве лосс-функции используется среднеквадратичная ошибка между шумом, добавленным в ходе прямого процесса диффузии и шумом предсказанным в ходе обратного на каждой итерации.

На обучении используется DDPM расписание добавления шума c 200 временных шагов.

## 4. Практическая часть

### 4.1 Используемые инструменты и библиотеки

Для написания кода и последующего проведения экспериментов был использован ряд фреймворков и библиотеки:

PyTorch — самый популярный фреймворк для глубокого обучения, который содержит в себе абстракции для тензорных вычислений, автоматического дифференцирования, вычислений на CUDA и ряд реализаций типовых слоёв нейронных сетей и многое другое [11].

Transformers — фреймворк для глубокого обучения, разработанный компанией huggingface, содержащий большое количество реализаций современных архитектур нейронных сетей и предобученных моделей, основанных на архитектуре трансформер [12].

Diffusers — фреймворк для глубокого обучения, разработанный компанией huggingface, содержащий большое количество реализаций современных архитектур нейронных сетей и предобученных моделей, основанных на диффузионных моделях [13].

Accelerate — фреймворк для ускорения обучения нейронных сетей, разработанный компанией huggingface, содержит ряд абстракций для упрощения обучения в DistriburedDataParallel (параллелизм по графическим процессорам) режиме, облегчает работу с перемещением данных между оперативной памятью и памятью графического процессов, оптимизирует вычисления [14].

### 4.2 Описание данных

В рамках данной работы был использован корпус данных LibriSpeech-960h [10], который содержит 960 часов размеченной английской речи из аудиокниг для обучения, большая часть книг взята из проекта «Гутенберг». Также в корпусе есть несколько валидационных сетов, которые разделены на две категории - «чистые» и «прочие», в зависимости от качества условий записи. Каждый из валидационных сетов содержит примерно 5 часов человеческой речи. Исходные аудио представлены в формате .flac c частотой 16 килогерц в виде моноканальных канальных записей.

### 4.3 Описание экспериментов

Эксперименты проводились на 8 графических процессорах NVIDIA A100-PCIE-40GB. Обучение происходило на датасете LibriSpeech-960h. Для обучения автокодировщика использовались референсные транскрипции, а для обучения диффузионной модели эмбеддинги аудио из wav2vec2.0-large и эмбеддинги текста из обученного автокодировщика.

#### 4.3.1 Обучение автокодировщика текста

Был проведён ряд экспериментов по обучению автокодировщика текста с различными параметрами.

Параметры при обучении нейросети сжатия:

* 3 скрытых слоя
* Оптимизатор AdamW с бетами (0.9, 0.999)
* Линейное расписание коэффциентов обучения
* Размер батча 365
* Общее количество итераций 50000
* Количество итераций разгогрева 1000
* Пиковое значение коэффициента обучение 1e-5

Параметры при обучении нейросети реконструкции:

* 3 скрытых слоя
* Оптимизатор AdamW с бетами (0.9, 0.999)
* Линейное расписание коэффциентов обучения
* Размер батча 365
* Общее количество итераций 50000
* Количество итераций разгогрева 1000
* Пиковое значение коэффициента обучение 1e-5

В качестве предобученной языковой модели использовался BART-base от компании Facebook AI. Результаты экспериментов представлены в таблице 1.

Таблица 1 – значение метрик для автокодировщика

|  |  |  |
| --- | --- | --- |
| Параметры | Rouge-L | BLEU |
| dim\_ae=64, num\_layers=1, num\_encoder\_latents=32, num\_decoder\_latents=32 | 0.9917 | 0.9716 |
| dim\_ae=64, num\_layers=3, num\_encoder\_latents=32,  num\_decoder\_latents=32 | 0.9964 | 0.9872 |
| dim\_ae=128, num\_layers=3, num\_encoder\_latents=64, num\_decoder\_latents=128 | 0.997 | 0.9916 |

#### 4.3.2 Получение векторных представлений аудио

Для получения векторых представлений использовалась модель wav2vec2.0-large от компании Facebook AI.

#### 4.3.3 Обучение предсказателя шума

После получения аудиоэмбеддингов и обучения автокодировщиках становится возможным обучить диффузионную модель.

На вход модель получает текущее латентное представление текста, аудиоэмбеддинг и номер шага диффузии.

Архитектура представляет собой трансформер с пре-леернормами и кросс-аттеншеном на части слоёв между внутренними представлениями сети и аудиоэмбеддингами.

Для эмбеддингов аудио используются обучаемые относительные позиционные эмбеддинги из [17] с параметрами conv\_pos = 32, conv\_pos\_groups = 4.

Для эмбеддингов аудио используются абсолютные синусоидальные эмбеддинги с параметрами t\_emb\_dim = 768, t\_emb\_max\_period = 1000.

По полученным эмбеддингам и информации о временном шаге диффузии модель предсказывает шум.

Исходя из значений предсказаний и истинных значений добавленного шума расчитывается среднеквадратичная ошибка, а далее происходит минимизация данной ошибки.

Параметры обучения:

* 24 трансформер-блока
* Оптимизатор AdamW с бетами (0.9, 0.999)
* Линейное расписание коэффициентов обучения
* Размер батча 500
* Общее количество итераций 700k
* Количество итераций разгогрева 30000
* Пиковое значение коэффициента обучения 3e-5

#### 4.3.4 Оценка и сравнения полученной модели

Оценка проводилась на валидационных датасетах из корпуса LibriSpeech, значение метрик для TransFusion и Conformer были взяты из статей.

Результаты экспериментов представлены в таблице 2.

Таблица 2 – значение WER на валидационных датасетах

|  |  |  |  |
| --- | --- | --- | --- |
| Название | TransFusion | Conformer | Предлагаемое  решение |
| dev-clean | 6.1 | 1.9 | 40.3 |
| dev-other | 8.3 | 4.4 | 45.6 |
| test-clean | 6.7 | 2.1 | 42.4 |
| test-other | 8.8 | 4.3 | 50.1 |

Результаты для Conformer указаны с учётом применения языковой модели для последующей переоценки возможных транскрипций.

Для объяснения причин таких низких метрик можно рассмотреть кривые обучения.

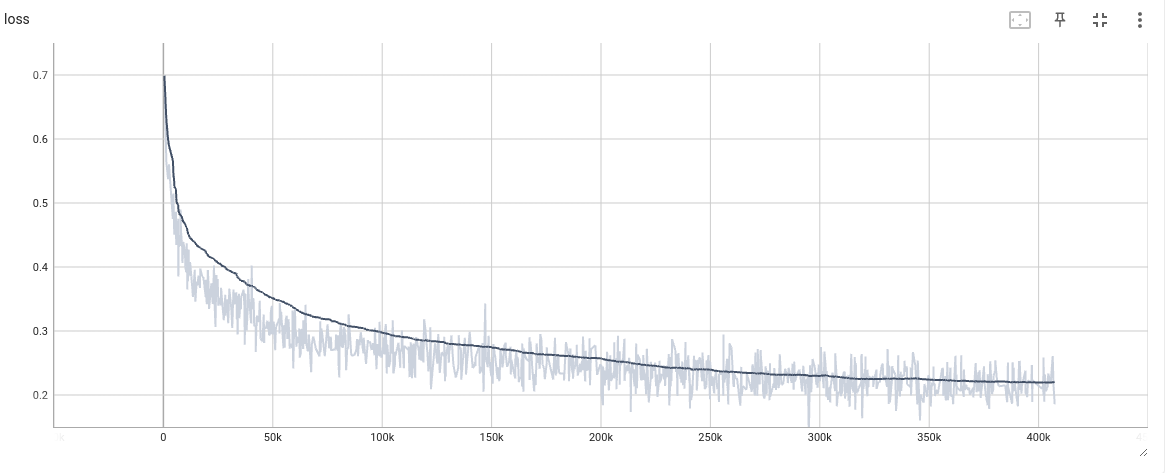


Рисунок 13 – График лосс функции на обучении предиктора шума

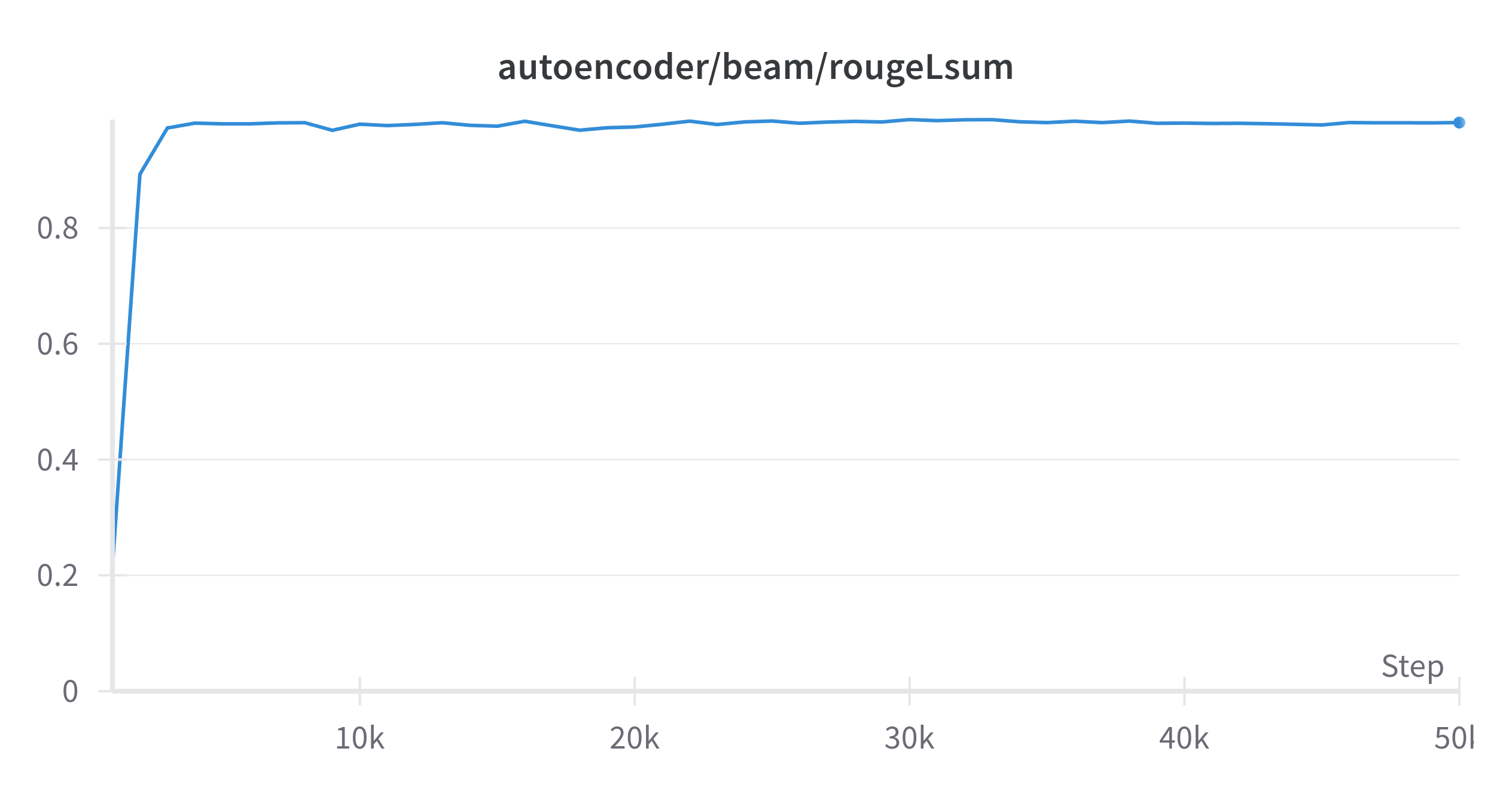


Рисунок 14 – График метрики rougeLsum на валидацци при обучении автокодировщика

Исходя из графика на рис. 13 можно сделать вывод о том, что автокодировщик хорошо выполняет задачу восстановления и почти не теряет данные, однако это ничего не говорит об организации его латентного пространства.

Вместе с тем на рис. 14 видно, что при обучении предиктора шума лосс-функции быстро выходит на плато на уровне 0.2 и начинает там осциллировать, также видно, что кривая лосс-функции довольно шумная. Так как процесс диффузии происходит в латеном пространстве автокодировщика можно сделать вывод о том, что оно устроено слишком сложно и предиктор шума не может достаточно хорошо оценить добавленный шум.

## 5. Обеспечение качества разработки, продукции, программного продукта

### 5.1 Определение потребителей

Потребителями разработанной модели могут быть:

* Компании, разрабатывающие системы видеоконфенцсвязи, использующие системы распознавания речи для транскрибации прошедших встреч
* Компании, разрабатывающие системы речевой аналитики, использующие системы распознавания речи для транскрибации звонков и последующего анализа текстов
* Компании, разрабатывающие голосовых помощников, использующие системы распознавания речи для распознавания команд и запросов пользователя

### 5.2 Функции продукции

Для того, чтобы ответить на вопрос, какие функции имеет, разработанный продукт, необходимо ввести определение функции изделия или услуги. Функции изделия или услуги – это требования и ожидания потребителя, которые могут быть установлены, предполагаются или являются обязательными.

Перечень пользовательских требования к разработанной системе:

* Пользователь должен получать результат распознавания речи
* Пользователю должна предоставляться оценка качества модели по метрике WER на размеченных данных

### 5.3 Качество и характеристики

Для обеспечения качества разрабатываемой системы был проведен обзор стандартов, протоколов, отраслевых требований и современных лучших практик, нужных в разработке. При этом качество – степень соответствия совокупности присущих характеристик объекта требованиям (согласно ГОСТ Р ИСО 9000-2015).

Знание протоколов, стандартов и пр. позволяет обеспечить соответсвие системы современным требования, повысить эффективность и удобство использования.

Анализ функциональных требований к разработке отражён в таблице 3.

Таблица 3 – функциональные требования

|  |  |  |
| --- | --- | --- |
| Функции по группам | Источник | Требование |
| Удовлетворенность | ГОСТ Р ИСО/МЭК 25010-2015 | Модель должна обеспечивать обработку результатов с помощью графического процессора |
| Полноценность | ГОСТ Р ИСО/МЭК 25010-2015 | Модель должна обеспечивать возможность распознавания речи в аудио |
| Доступность | ГОСТ Р ИСО/МЭК 25010-2015 | Разработанная модель должна предоставлять возможность запуска как на графическом процессоре, так и на центральном |
| Конфиденциальность | ГОСТ Р ИСО/МЭК 25010-2015 | Доступ к результатам работы алгоритмов может быть предоставлен только авторам исходных аудиозаписей |
| Документация и отчётность | Лучшие отраслевые практики | Ведение документации и составление отчётов о применении модели к задаче распознавания речи, включая описание формата аудиоданных, используемые вычислительные мощности, версии программного кода |

### 5.4 Измерение характеристик качества. Операциональное определение

Операциональное определение (ОО) – это уточнение значения того или иного термина применительно к данной системе, находящейся в конкретных условиях и для людей, в ней задействованных.

ОО необходим для уменьшения случаев разночтения и неоднозначности при общении между людьми, задействованными в системе, а также для более точного определения целей и задач, которые необходимо достичь.

ОО должно содержать как минимум три компоненты:

1. Требования или стандарт, относительно которого оценивается результат измерения или испытания (критерий).
2. Метод испытания или процедура измерения свойства объекта (тест)
3. Процедура принятия решения (анализ), которое показывает, соответствует ли результат испытания стандарту.

Операциональные определения для разработанной модели описаны в таблице 4.

По итогам рассмотрения оперциональных опеределений было установлена, что разработанная модель не удовлетворяет описанным требования, однако существует ряд предложений по улучшению, сформированных в таблице 5

Таблица 5 — предложения по улучшению модели.

|  |  |  |  |
| --- | --- | --- | --- |
| Требование | Анализ текущего состояния | Возможные причины невыполнения критерия | Предложение по улучшению |
| Улучшение качества распознавания | Текущее значение WER составляет 40% | Недостаточное качество автокодировщика | Изменение архитектуры автокодировщика для получения лучшего латентного пространства, замена его на другие алгоритмы для получения эмбеддингов для дальнейшего использования в диффузии, например, flow-matching |

## ЗАКЛЮЧЕНИЕ

В рамках данной работы был проведён анализ существующих решений, рассмотрен подход к диффузионным моделям в задаче распознавания речи из статьи TransFusion.

Далее были изучены актуальные методы для генерации текста с помощью текстовой диффузии и её обуславливания на информацию из аудио.

После выбора метода была реализована диффузионная модель для генерации текста на основе аудиоэмбеддинга. Данная модель использует текстовый автокодировщик для получения латентных представлений текста, которые используются в диффузии.

Для полученной модели была проведена оценка качества и сравнение с аналогами на основе валидационных датасетов из корпуса LibriSpeech-960h

В результате работы была провена гипотеза о применимости диффузионных моделей на непрерывных представлениях в задаче распознавания речи. Результатом проверки гипотезы стал вывод о неприменимости данного подхода из-за плохого качества.

Для улучшения качества имеет смысл используются другие способы получения латентного пространства, такие как вариационный автоэнкодер или flow-matching.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. High-Resolution Image Synthesis with Latent Diffusion Models [Электронный ресурс] // URL: <https://arxiv.org/abs/2112.10752> (дата обращения: 10.05.2024)
2. Popov V. et al. Grad-tts: A diffusion probabilistic model for text-to-speech //International Conference on Machine Learning. – PMLR, 2021. – С. 8599-8608.
3. Baas M., Eloff K., Kamper H. Transfusion: Transcribing speech with multinomial diffusion //Southern African Conference for Artificial Intelligence Research. – Cham : Springer Nature Switzerland, 2022. – С. 231-245.
4. Hinton G. E., Salakhutdinov R. R. Reducing the dimensionality of data with neural networks //science. – 2006. – Т. 313. – №. 5786. – С. 504-507.
5. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension [Электронный ресурс] // URL: <https://arxiv.org/abs/1910.13461> (дата обращения: 10.05.2024)
6. Raffel C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer //Journal of machine learning research. – 2020. – Т. 21. – №. 140. – С. 1-67.
7. Papineni K. et al. Bleu: a method for automatic evaluation of machine translation //Proceedings of the 40th annual meeting of the Association for Computational Linguistics. – 2002. – С. 311-318.
8. Deep Unsupervised Learning using Nonequilibrium Thermodynamics [Электронный ресурс] // URL: <https://arxiv.org/abs/1503.03585> (дата обращения: 10.05.2024)
9. Vaswani A. et al. Attention is all you need //Advances in neural information processing systems. – 2017. – Т. 30.
10. LIBRISPEECH: AN ASR CORPUS BASED ON PUBLIC DOMAIN AUDIO BOOKS [Электронный ресурс] // URL: <https://www.danielpovey.com/files/2015_icassp_librispeech.pdf> (дата обращения: 10.05.2024)
11. PyTorch [Электронный ресурс] // URL: <https://pytorch.org/docs/stable/index.html> (дата обращения: 10.05.2024)
12. Transformers [Электронный ресурс] // URL: <https://huggingface.co/docs/transformers/index> (дата обращения: 10.05.2024)
13. Diffusers [Электронный ресурс] // URL: <https://huggingface.co/docs/diffusers/index> (дата обращения: 10.05.2024)
14. Accelerate [Электронный ресурс] // URL: <https://huggingface.co/docs/accelerate/index> (дата обращения: 10.05.2024)
15. Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions [Электронный ресурс] // URL: <https://proceedings.neurips.cc/paper/2021/hash/67d96d458abdef21792e6d8e590244e7-Abstract.html> (дата обращения: 10.05.2024)
16. Chen S. et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing //IEEE Journal of Selected Topics in Signal Processing. – 2022. – Т. 16. – №. 6. – С. 1505-1518.
17. PAY LESS ATTENTION WITH LIGHTWEIGHT AND DYNAMIC CONVOLUTIONS [Электронный ресурс] // URL: <https://arxiv.org/pdf/1901.10430> (дата обращения: 10.05.2024)
18. Categorical reparameterization with gumbel-softmax [Электронный ресурс] // URL: <https://arxiv.org/abs/1611.01144> (дата обращения: 10.05.2024)
19. Conformer: Convolution-augmented Transformer for Speech Recognition // URL: <https://arxiv.org/abs/2005.08100> (дата обращения: 10.05.2024)
20. Ho J., Jain A., Abbeel P. Denoising diffusion probabilistic models //Advances in neural information processing systems. – 2020. – Т. 33. – С. 6840-6851.
21. Lovelace J. et al. Latent diffusion for language generation //Advances in Neural Information Processing Systems. – 2024. – Т. 36.
22. Graves A., Graves A. Connectionist temporal classification //Supervised sequence labelling with recurrent neural networks. – 2012. – С. 61-93.
23. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations [Электронный ресурс] // URL: <https://arxiv.org/abs/2006.11477> (дата обращения: 10.05.2024)
24. Layer Normalization [Электронный ресурс] // URL: <https://arxiv.org/abs/1607.06450> (дата обращения: 10.05.2024)
25. Decoupled Weight Decay Regularization [Электронный ресурс] // URL: <https://arxiv.org/abs/1711.05101> (дата обращения: 10.05.2024)
26. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition [Электронный ресурс] // URL: <https://arxiv.org/abs/1904.08779> (дата обращения: 10.05.2024)

## ПРИЛОЖЕНИЕ А

Таблица 4 – Операциональные определения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Требование | Тест | | Решение (правило) | | |
| Характеристика | Метод измерения | Соответствие | Частичное соответствие | Несоответствие |
| Качественное распознаванияе | Метрика WER (word error rate) | Расчёт метрики между эталонными транскрипциями и результатами работы модели | 15% и меньше | От 15% до 25% | Выше 25% |
| Скорость распознавания | Метрика RTF (real-time factor) | Замер RTF на наборе аудио | Более 10 | От 1 до 10 | Менее 1 |