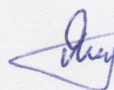


ГОСУДАРСТВЕННЫЙ КОМИТЕТ ПО НАУКЕ И ТЕХНОЛОГИЯМ  
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ  
ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ  
«ИНСТИТУТ ПРОБЛЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»

*На правах рукописи*



УДК 004.93

**Пикалёв Ярослав Сергеевич**

**СОВЕРШЕНСТВОВАНИЕ МЕТОДОВ И ПРОГРАММНЫХ СРЕДСТВ  
РАСПОЗНАВАНИЯ СЛИТНОЙ РУССКОЙ РЕЧИ**

Специальность 05.13.01 – Системный анализ, управление и обработка  
информации (по отраслям) (технические науки)

**Диссертация**

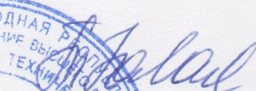
на соискание ученой степени  
кандидата технических наук

Научный руководитель:  
кандидат технических наук  
Ермоленко Т.В.



Идентичность всех экземпляров  
ПОДТВЕРЖДАЮ  
Ученый секретарь диссертационного  
совета Д 01.024.04  
кандидат технических наук, доцент



 Т.В. Завадская



## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
ГЛАВА 1 СОВРЕМЕННЫЕ ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ СЛИТНОЙ РЕЧИ.....	13
1.1 Архитектура современных систем автоматического распознавания слитной речи .....	13
1.2 Технологии извлечения акустических признаков.....	14
1.2.1 Параметризация речевого сигнала .....	15
1.2.2 Акустическое моделирование.....	18
1.3 Нейросетевой подход к распознаванию речи.....	22
1.3.1 Основные принципы функционирования глубоких нейросетей... ..	22
1.3.2 Технологии обучения нейронных сетей .....	24
1.3.3 Архитектуры глубоких нейронных сетей, используемых для распознавания речи .....	29
1.3.4 Технологии обучения акустических моделей на основе глубоких нейросетей.....	35
1.4 Языковое моделирование .....	36
1.5 Генерация транскрипций слов .....	37
1.6 Декодирование.....	39
1.7 Выводы к главе 1 .....	41
ГЛАВА 2 ПОДГОТОВКА ДАННЫХ ДЛЯ ОБУЧЕНИЯ АКУСТИЧЕСКОЙ И ЯЗЫКОВОЙ МОДЕЛЕЙ .....	43
2.1 Описание речевых и текстовых данных для обучения акустической и языковой моделей.....	43
2.1.1 Описание речевых данных.....	43
2.1.2 Описание текстовых корпусов.....	45

2.2 Модификация алгоритма Смита-Ватермана для проверки соответствия текстовых расшифровок и аудио.....	47
2.3 Разработка метода аугментации речевых данных.....	52
2.4 Разработка методов нормализации текста.....	55
2.4.1 Общая схема нормализации текста.....	56
2.4.2 Разработка нейросетевой модели определения языка текста.....	59
2.4.3 Разработка системы синтаксического анализа.....	61
2.4.4 Разработка метода обработки цифробуквенных комплексов и аббревиатур.....	65
2.4.5 Разработка методов «ёфикации» и «йфикации» слов.....	68
2.6 Выводы к главе 2.....	70
<b>ГЛАВА 3 РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ТРАНСКРИПЦИЙ CYR2TRANS.....</b>	<b>72</b>
3.1 Особенности фонетики русского языка.....	72
3.1.1 Особенности произношения гласных.....	72
3.1.2 Особенности произношения иноязычных слов.....	75
3.1.3 Произношение слов с апострофом.....	78
3.1.4 Правила образования сложносоставных слов.....	79
3.2 Общая схема работы системы автоматической генерации транскрипций Cyr2Trans.....	82
3.3 Разработка метода разделения слов на подслова.....	85
3.3.1 Общая схема работы метода разделения слова на подслова.....	85
3.3.2 Модификация алгоритма стемматизации SnowballStemmer.....	87
3.4 Разработка нейросетевой модели для определения позиции ударения..	89
3.5 Разработка метода получения практической транскрипции для вставок на латинице.....	92

3.5.1 Недостатки существующих методов получения практической транскрипции.....	93
3.5.2 Особенности произношения английских вставок носителями русского языка .....	95
3.5.3 Общая схема работы метода получения практической транскрипции.....	97
3.5.4 Разработка нейросетевой модели для генерации практической транскрипции.....	100
3.6 Разработка нейросетевой модели получения транскрипции слов-исключений .....	107
3.7 Выводы к главе 3 .....	108
ГЛАВА 4 РАЗРАБОТКА РОБАСТНЫХ АКУСТИЧЕСКИХ МОДЕЛЕЙ НА ОСНОВЕ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ.....	110
4.1 Факторы, искажающие речевой сигнал в системах распознавания речи	110
4.2 Технология повышения робастности акустической модели .....	112
4.2.1 Разработка алгоритма обучения акустической модели с использованием машинного обучения .....	112
4.2.2 Техника извлечения информативных акустических признаков ..	116
4.3 Разработка нейросетевой модели для предсказания последовательности фонем.....	123
4.4 Численные исследования эффективности использования предложенной мультимодульной архитектуры акустической модели .....	128
4.5 Выводы к главе 4 .....	130
ГЛАВА 5 РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ РУССКОЙ СЛИТНОЙ РЕЧИ .....	131
5.1 Структура системы распознавания речи.....	131



5.1.1	Функционирование системы автоматического распознавания русской речи в режиме обучения.....	132
5.1.2	Функционирование системы автоматического распознавания русской речи в режиме распознавания.....	140
5.1.3	Реализация модуля денормализации текста.....	144
5.2	Оценка эффективности разработанной системы.....	147
5.3	Выводы к главе 5.....	151
	ЗАКЛЮЧЕНИЕ.....	152
	СПИСОК СОКРАЩЕНИЙ.....	155
	СПИСОК ЛИТЕРАТУРЫ.....	158
	ПРИЛОЖЕНИЕ А. Барк- и Мел-шкалы.....	177
	ПРИЛОЖЕНИЕ Б. Схема обучения акустических моделей.....	179
	ПРИЛОЖЕНИЕ В. Документы, подтверждающие внедрение результатов диссертации.....	180

## ВВЕДЕНИЕ

**Актуальность.** Автоматическое распознавание речи является динамично развивающимся направлением в области искусственного интеллекта. Задача распознавания речи получила широкое распространение вследствие высокой применимости на практике. Однако, на сегодняшний день в сфере распознавания русскоязычной слитной речи успехи достигнуты только в пределах словарного запаса, связанного с узкой предметной областью, а распознавание слитной речи до сих пор не имеет четкого решения в силу ряда возникающих трудностей, связанных с отсутствием объёмного аннотированного речевого и нормализованного текстового корпуса, необходимого для статистического моделирования языка; флективностью, а также свободным порядком слов во фразе; орфоэпическими нормами, увеличивающими акустическую вариативность русской речи.

В связи с этим, задача совершенствования методов и программных средств дикторонезависимого распознавания слитной русской речи, позволяющих обеспечивать высокое качество распознавания, учитывать особенности русской речи и адаптироваться под любую предметную область, является актуальной и имеет отраслевое значение.

**Связь работы с научными программами, планами, темами.** В основу диссертационного исследования положены работы, выполненные в Институте проблем искусственного интеллекта в рамках научно-исследовательских работ: «Разработка методов распознавания слитно произнесённых фраз в рамках концепции пофонемного распознавания речи с обобщённой транскрипцией» (№Г/Р 0113U001326); «Исследование и разработка методов семантического анализа и интерпретации потоков данных интеллектуальными системами» (№Г/Р 0118D000003).

**Степень разработанности темы исследования.** Системы распознавания слитной русскоязычной речи от компаний Google и Яндекс демонстрируют высокую точность распознавания речи – около 75–90%, а методы и модели распознавания для русского языка, как правило, заимствуются из другого языка.

Поэтому качество их работы значительно падает при распознавании разговорной речи.

Среди зарубежных исследователей, занимающихся данным направлением, стоит выделить D. Povey, G. Hinton, P. Cosi, A. Graves, O. Abdel-Hamid, A. Baevski, D. Amodei, G. Saon, L. Deng, A. Senior, T. Sainath. Среди российских исследователей следует отметить работы А. Карпова, А. Ронжина, И. Кипятковой, И. Меденникова, Д. Кушнира, И. Тампеля.

Российской компанией ООО «ЦРТ» разработана система автоматической генерации субтитров в режиме реального времени, которая использует искусственные нейросети (ИНС) и обучена на 32 часах записей новостей. Статистическая модель языка построена для конкретной тематики телепередачи. В результате качество распознавания системы при переходе на другие предметные области сильно снижается.

В рамках российской разработки RealSpeaker применяется видеорасширение для увеличения точности программ распознавания речи, однако данное решение не распознает большинство союзов, а при равномерной диктовке пропадают части фраз.

Таким образом, на настоящий момент не существует систем распознавания слитной русской речи, сопоставимых по качеству с вышеупомянутыми системами для английского языка.

Учитывая вышеизложенное, можно сделать вывод о необходимости повышения эффективности существующих моделей, методов и программных средств для построения систем дикторонезависимого распознавания слитной русской речи, способных адаптироваться под любую предметную область.

**Цель диссертационного исследования** – повышение эффективности дикторонезависимой системы автоматического распознавания слитной русской речи за счет модернизации алгоритмов системного анализа, обработки и распознавания речевой информации, а также автоматической обработки текстовых данных.

Для достижения цели в работе решены следующие задачи:



1) осуществлен выбор методов построения акустической (АМ) и языковой моделей (ЯМ), транскриптора, классификатора фонем и декодера, исходя из анализа современных технологий распознавания слитной русской речи;

2) собраны и обработаны речевые и текстовые данные, находящиеся в открытом доступе, с целью создания аннотированного речевого корпуса для обучения АМ и ЯМ;

3) разработаны методы автоматического построения словаря транскрипций, позволяющие определять позицию ударения, генерировать транскрипцию для слов-исключений и осуществлять практическую транслитерацию с учетом орфоэпических норм русского языка;

4) разработаны методы получения робастных акустических признаков и обучения АМ на основе глубоких нейронных сетей;

5) разработан классификатор для распознавания фонем;

6) оценено качество построенных АМ и классификатора фонем с целью обоснования предложенного подхода их использования в системах дикторонезависимого распознавания слитной русской речи;

7) построена система автоматического распознавания слитной русской речи (Automatic Speech Recognition, ASR) на основе предложенных методов и моделей, проведена оценка качества ее работы по сравнению с российскими и зарубежными системами.

**Объект исследования** – процессы анализа, обработки и классификации речевого сигнала в системах автоматического распознавания речи.

**Предмет исследования** – методы и алгоритмы построения АМ и ЯМ, методы распознавания речевого сигнала.

**Научная новизна** полученных результатов заключается в следующем:

1) получили дальнейшее развитие нейросетевые методы автоматического определения позиции ударения в слове за счет модернизации архитектуры нейросети типа Transformer, которая заключается в увеличении количества слоёв, использовании методов градиентного отсечения и teacher forcing для оптимизации

параметра скорости обучения, что позволило повысить точность определения позиции удара на 10% по сравнению со стандартной моделью Transformer;

2) усовершенствована seq2seq модель для генерации практических транскрипций англоязычных слов и слов-исключений за счет применения механизма обучения с подкреплением и метода beam-search для выбора наиболее вероятной последовательности символов, что позволило повысить точность модели по критерию количества ошибочно сгенерированных символов на 0,8% и 3%, по критерию неправильно сгенерированных слов на 0,6% и 9% соответственно.

3) предложена модель нейросетевой параметризации, основанная на объединении ансамбля нейронных сетей с «узким горлом» и архитектуры ResNet-50. Использование данной модели позволяет повысить точность распознавания на 2,7% по сравнению с моделью, извлекающей стандартные bottleneck-признаки;

4) получили дальнейшее развитие методы нейросетевой классификации фонем за счет использования механизма внимания в последнем скрытом слое сети, включающей в себя нейросеть с временными задержками и двунаправленную нейросеть с долгой кратковременной памятью, что позволило сохранять высокую точность на относительно небольшом обучающем наборе аудиоданных, свойственную системам, для обучения которых требуется речевая база длительностью в десятки тысяч часов.

**Теоретическая значимость** научных результатов, полученных в ходе диссертационного исследования, определяется созданием нового подхода нейросетевой параметризации речевого сигнала для обучения АМ. В частности, предложенная технология извлечения робастных признаков из скрытых слоев иерархической мультимодульной ИНС вносит свой вклад в теорию понимания кодирования сигнала на основе глубоких нейросетей, которые на сегодняшний день рассматриваются как «черный ящик».

**Практическое значение работы.** Материалы исследований могут быть использованы при разработке методов автоматического формирования аннотированных речевых баз данных; методов автоматического построения словаря транскрипций системах синтеза речи; методов получения робастных

акустических признаков и обучения АМ, а также при разработке классификатора для распознавания фонем в системах голосового управления и поиска по голосовому запросу, а также в системах диктовки с приемлемым уровнем ошибок.

Результаты и выводы диссертационной работы нашли применение в Институте проблем искусственного интеллекта, что подтверждается справкой о внедрении (справка №347/01-01 от 01.12.2020).

**Методология и методы исследования.** Для решения поставленных задач использовались следующие методы:

– прикладной лингвистики для анализа закономерностей синтаксиса, морфологии и фонетического состава русского языка, анализа структуры существующих словарей;

– методы математической статистики для оценки эффективности разработанных моделей;

– методы цифровой обработки сигналов для получения акустических характеристик речевых сигналов;

– методы машинного обучения для построения АМ и ЯМ и классификации фонем.

#### **Положения, выносимые на защиту.**

1. Доказано, что модификация seq2seq модели для генерации практических транскрипций англоязычных слов и слов-исключений на базе архитектуры Transformer за счет применения механизма обучения с подкреплением обеспечивает повышение точности генерации транскрипции по критерию количества ошибочно сгенерированных символов и по критерию неправильно сгенерированных слов.

2. Установлено, что усовершенствование метода акустического моделирования за счет аугментации и модификации признаков для получения адаптивных и дискриминативных характеристик повышает робастность акустических признаков, обеспечивая тем самым их инвариантность к смене диктора и акустической обстановке и повышение точности распознавания.

3. Доказано, что предложенный метод нейросетевой параметризации речевого сигнала на основе иерархической мультимодульной архитектуры MultiBN и



архитектуры ResNet-50 позволяет извлечь из скрытых слоев информативные акустические признаки, устойчивые по отношению к темпу речи, акустической среде и междикторской вариативности, что приводит к повышению точности распознавания на по сравнению с моделью, извлекающей стандартные bottleneck-признаки.

По направлению исследований, содержанию научных положений и выводов, существу полученных результатов диссертационная работа соответствует паспорту специальности 05.13.01 – Системный анализ, управление и обработка информации (по отраслям) (технические науки) по областям исследований: п.5 «Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений и обработки информации»; п.12 «Визуализация, трансформация и анализ информации на основе компьютерных методов обработки информации».

Степень достоверности и апробация результатов обеспечивается полнотой анализа теоретических и практических исследований, положительной оценкой на научных конференциях и семинарах, выполненными публикациями.

**Апробация результатов работы.** Основные научные положения и результаты диссертационной работы доложены, обговорены и приняты на конференциях: «Искусственный интеллект: теоретические аспекты и практическое применение» (г. Донецк, 2020); Международная научная конференция студентов и молодых учёных «Донецкие чтения» (г. Донецк, 2019, 2018, 2017); Международная научно-техническая конференция «Информатика, управляющие системы, математическое и компьютерное моделирование» (г. Донецк, 2019, 2016); XII Мультиконференция по проблемам управления (г. Геленджик, Дивноморское, 2019); Международная научно-техническая конференция «Интеллектуальные технологии и проблемы математического моделирования» (г. Геленджик, Дивноморское, 2018); VIII Международная конференция по когнитивной науке (г. Светлогорск, 2018).

**Личный вклад автора.** Соискателем лично решены задачи диссертации. В работах, опубликованных в соавторстве, личный вклад автора заключается в выполнении аналитических расчётов, практических экспериментов, реализации

программных решений и статистическом анализе полученных результатов. Все выносимые на защиту положения получены автором лично.

**Публикации.** Основные научные результаты диссертации опубликованы в 17 научных работах, в том числе в 5 научных статьях в изданиях, рекомендуемых ВАК для публикации трудов на соискание ученых степеней.

**Объем и структура диссертации.** Диссертация изложена на 180 страницах машинописного текста и состоит из списка сокращений, введения, пяти глав, выводов, заключения, списка литературы, 3 приложений. Работа иллюстрирована 47 рисунками, содержит 11 таблиц. Список литературы включает 186 наименований.

## ГЛАВА 1

## СОВРЕМЕННЫЕ ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ СЛИТНОЙ РЕЧИ

## 1.1 Архитектура современных систем автоматического распознавания слитной речи

Архитектура современных систем автоматического распознавания речи представлена на рисунке 1.1.

Векторы акустических признаков извлекаются из фреймов сигнала, как правило, длиной 25 миллисекунд (на этих участках речевой сигнал можно считать квазистационарным) с шагом 10 миллисекунд, который подаётся на вход распознавателя, активная речь отделяется от фонового шума или тишины. Полученные признаки проходят адаптацию к диктору и окружению.

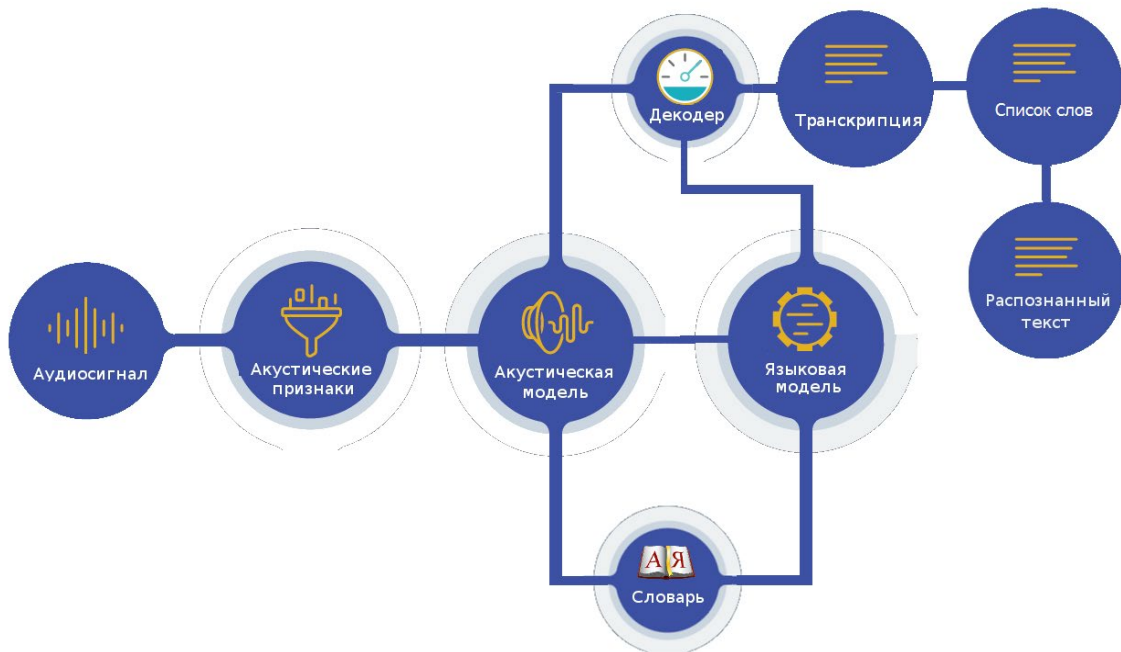


Рисунок 1.1 – Типовая структура автоматической системы распознавания речи

Акустическая модель (АМ) описывает плотность распределения вероятностей акустических классов (например, фонем). Для каждого звука изначально строится сложная статистическая модель, которая описывает произнесение этого звука в речи. Каждый фрейм пропускается через акустическую



модель, при этом он сопоставляется не с одной фонемой, а с несколькими, подходящими с разной степенью вероятности. Кроме того, система учитывает вероятности переходов, то есть определяет, какие фреймы могут идти следом за конкретной фонемой. Система распознавания сопоставляет входящий речевой сигнал с фонемами, а затем подбирает варианты слов. Точность результатов напрямую зависит от полноты фонетического алфавита системы. Например, фонетический алфавит Яндекса состоит из 4000 элементарных единиц, которые включают фонемы, их части и сочетания.

Языковая модель (ЯМ) описывает вероятность появления слова в контексте других слов. Для этого составляются лингвистические модели, подкрепляющие распознавание фонем правилами образования слов (морфология), словосочетаний и предложений (синтаксис), смысловых связей (семантика). С помощью ЯМ система определяет вероятный порядок слов и при необходимости восстанавливает нераспознанные слова по смыслу, исходя из контекста и имеющейся статистики.

Словарь транскрипций устанавливает связь между последовательностями акустических классов, описываемых АМ, и словами, описываемыми ЯМ.

Декодер анализирует вероятности, генерируемые АМ и ЯМ, совмещая данные от этих моделей, и на основании их объединения выдает конечный результат в виде наиболее вероятной последовательности слов.

Рассмотрим методы, используемые для разработки вышеуказанных компонентов ASR, более подробно.

## 1.2 Технологии извлечения акустических признаков

Наряду с Герцами для частотных параметров звука используются альтернативные системы измерения, основанные на физиологическом (Барки) и психофизиологическом (Мелы) механизме его восприятия.

### 1.2.1 Параметризация речевого сигнала

Критическая полоса – это минимальная полоса частот, которая возбуждает одну и ту же часть базилярной мембраны. В частотном промежутке от 0 до 16 кГц опытным путем определены 24 критические полосы, приведенные в таблице А1 (Приложение А). Если же звуковой сигнал переходит из одной критической полосы в другую, то слуховые ощущения в момент перехода заметно изменяются, потому что мозг анализирует информацию, полученную из разных критических полос, отдельно. Для удобства работы с критическими полосами существует специальная единица измерения частоты – Барк.

На мел-шкале равное изменение частоты в Мелах соответствует равному изменению ощущения высоты тона. Шкалы барков и мелов приблизительно совпадают, хотя некоторые расхождения наблюдаются в области средних частот (Рисунок А.1).

Одними из наиболее часто используемых в современных системах распознавания речи признаков являются Мел-частотные кепстральные коэффициенты (Mel-Frequency Cepstral Coefficients, MFCC) [3]. Алгоритм вычисления MFCC признаков подробно описан в [4–6] и состоит из следующих этапов: предсказание – фильтрация входного сигнала, применяемая для спектрального выравнивания сигнала; дискретное преобразование Фурье для каждого кадра; построение набора из  $M$  треугольных фильтров, равномерно расположенных на Мел-шкале; вычисление логарифмов энергии спектра для построенного набора треугольных фильтров (Mel-frequency filterbank log energies, FBANK); дискретное косинусное преобразование для вычисленных на предыдущем шаге логарифмов энергии. В качестве итоговых MFCC признаков берутся первые несколько, как правило 13, компонентов кепстрального вектора.

Помимо MFCC-признаков используются коэффициенты линейного предсказания (Linear Predictive Codes, LPC) и перцепционные коэффициенты линейного предсказания (Perceptual Linear Prediction, PLP) [7]. Основной идеей методов LPC является возможность аппроксимации текущего отсчета речевого

сигнала с помощью линейной комбинации  $p$  предшествующих отсчетов с коэффициентами линейного предсказания  $\alpha_k$ .

PLP отличается от LPC тем, что пытается учесть особенности восприятия различных частот человеком: перед нахождением коэффициентов линейного предсказания речевой сигнал пропускается через фильтры, полосы пропускания которых изменяются в соответствии с барк-шкалой.

Анализ состояния области распознавания речи показал, что на сегодняшний день MFCC применяется наиболее широко.

Помимо указанных методов существует ряд техник для модификации извлечённых акустических признаков с целью получения шумоустойчивых, а также адаптивных и дискриминативных характеристик.

Для снижения степени вариативности речевого сигнала применяется нормализация среднего кепстра (Cepstral Mean Normalization, CMN) и дисперсии (Cepstral Mean and Variance Normalization, CMVN) [8], а также включение временного контекста в кепстральные кадры для моделирования динамики речевого сигнала. Общая идея этого метода заключается в вычислении скорости и ускорения MFCC-коэффициентов (дельта и дельта-дельта коэффициентов) для соседних кадров внутри окна, обычно, из 4 кадров. Эти коэффициенты добавляются к статическому кепстру для формирования окончательного вектора признаков [9].

Для лучшей разделимости фонетических классов применяют метод линейной матрицы проекций, отображающей вектор, полученный путем объединения последовательных кадров в пространстве низкой размерности. Разделители обычно вычисляются с помощью критерия линейного дискриминантного анализа (Linear Discriminant Analysis, LDA) [10]. Чтобы сделать гипотезу моделирования диагональной ковариации более допустимой, пространственные признаки LDA «переворачиваются» посредством преобразования полупривязанной ковариации (Semi-Tied Covariance, STC) и линейного преобразования признаков, максимизирующего среднее правдоподобие (Maximum Likelihood Linear Transformation, MLLT) [11], целью которых является сведение к минимуму вероятности потери между полной и диагональной ковариацией Гаусса.

Для борьбы с нестационарными шумами используют «стерео кусочно-линейный компенсатор внешних воздействий» (Stereo-based Piecewise Linear Compensation for Environments, SPLICE) [12]. Методика SPLICE заключается в улучшении характеристик путем замены помех в зашумленном речевом сигнале наиболее вероятным вектором коррекции, который является ожидаемой разницей между чистой и зашумленной речью.

Для компенсации несоответствия распределения обучающих и тестовых речевых данных используют уравнивание на основе квантилей (quantile equalization, QE) [13], которое заключается в параметризации компенсационной функции за счёт оценки минимизации квадрата расстояния между текущими квантилями и обучающими квантилями из фильтрационного банка тоновых частот.

Для борьбы с внедикторскими вариациями применяют методы нормализации диктора: деформации оси частот, чтобы соответствовать длине вокального тракта контрольного громкоговорителя (Vocal Tract Length Normalization, VTLN) [14]; аффинное преобразование функции максимизации вероятности как пространство характеристик максимального правдоподобия линейной регрессии (Feature space Maximum Likelihood Linear Regression, fMLLR) [15]. Применение этих методов позволяет сократить ошибку распознавания на 5–30%.

Широко используется техника адаптивного обучения дикторов (Speaker Adaptive Training, SAT) – это метод максимизации вероятности получения данных обучения с учетом моделей, адаптированных к fMLLR. Данный метод предназначен для коррекции параметров акустической модели для улучшения качества ее работы в условиях, отличных от условий обучения. SAT представляет собой максимизацию квадратичной целевой функции для каждого среднего гауссова значения.

Метод пространственно-характеристического дискриминативного обучения использует область характеристик минимальных фоновых ошибок (fMPE) [16]. fMPE – преобразование, обеспечивающее независимые от времени сдвиги регулярных характеристических векторов, путем линейного проецирования из пространства высокой размерности гауссовских распределений. Проекция

обучается таким образом, чтобы повысить уровень распознавания между верными и некорректными последовательностями слов.

На рисунке 1.2 приведена классификация методов параметризации [17].

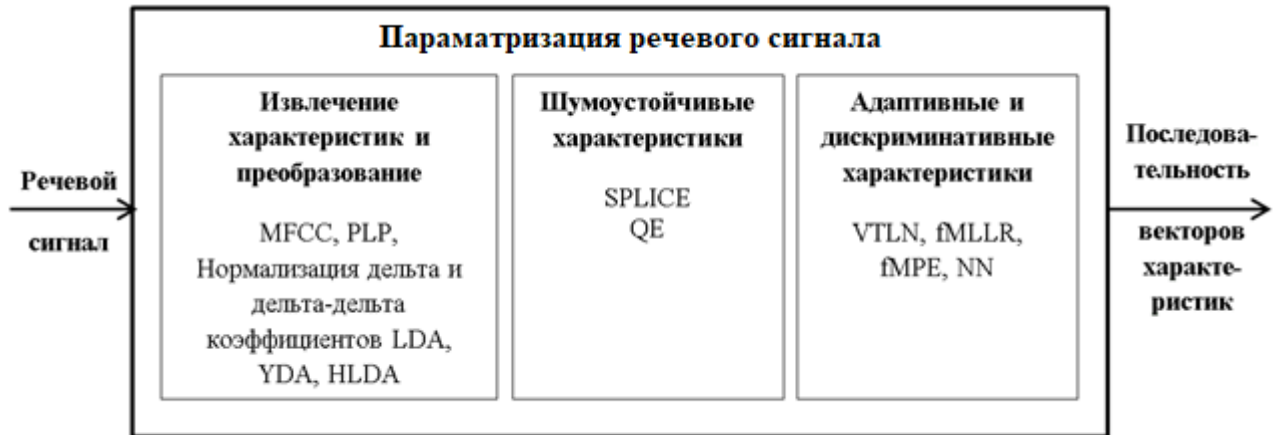


Рисунок 1.2 – Классификация методов параметризации речевого сигнала

В настоящее время получила широкое распространение нейросетевая (Neural Networks, NN) параметризация речевого сигнала. Вектора акустических признаков подаются вход нейросети (Artificial Neural Networks, ANN), вычисляющей распределение вероятностей целевых классов, которое дополняет вектор акустических признаков. Для уменьшения размерности полученного вектора применяют метод LDA [18] или другой нейросетевой подход – обработку с узким горлом [19]. Более детально нейросетевые методы параметризации речевого сигнала и акустического моделирования будут описаны в пункте 1.3.4.

### 1.2.2 Акустическое моделирование

Большинство современных систем автоматического распознавания речи используют скрытые марковские модели (Hidden Markov Models, HMM) для учета временной вариативности речевого сигнала [20–24]. HMM – модель, состоящая из  $N$  состояний, в каждом из которых некоторая система может принимать одно из  $M$  значений какого-либо параметра. Процесс называется марковским, если для

каждого момента времени вероятность любого состояния системы в следующий момент зависит только от состояния системы в настоящий момент и не зависит от того, каким образом система пришла в это состояние.

Вероятности переходов между состояниями задается матрицей вероятностей  $A = \{a_{ij}\}$ , где  $a_{ij}$  – вероятность перехода из  $i$ -го в  $j$ -е состояние. Вероятности выпадения каждого из  $M$  значений параметра в каждом из  $N$  состояний задается вектором  $B = \{b_j(k)\}$ , где  $b_j(k)$  – вероятность выпадения  $k$ -го значения параметра в  $j$ -м состоянии. Вероятность наступления начального состояния задается вектором  $\pi = \{\pi_i\}$ , где  $\pi_i$  – вероятность того, что в начальный момент система окажется в  $i$ -м состоянии. Таким образом, скрытой марковской моделью называется тройка  $\lambda = \{A, B, \pi\}$ .

Таким образом, НММ представляет собой конечный автомат, изменяющий свое состояние в каждый дискретный момент времени  $t$  (Рисунок 1.3). Переход из состояния  $s_i$  в состояние  $s_j$  осуществляется случайным образом с вероятностью  $a_{ij}$ . В каждый дискретный момент времени модель порождает вектор наблюдений  $o_t$  (который в конкретной задаче является вектором признаков, полученным в преобразователе сигнала) с вероятностью  $b_j(o_t)$ . Каждая такая модель обозначает один из звуков речи или отсутствие звука (одна из моделей).

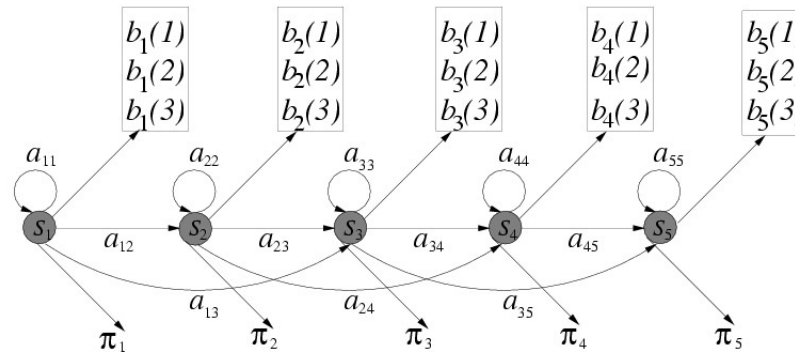


Рисунок 1.3 – Скрытая Марковская модель

НММ представляет спектральные свойства речи с помощью параметрического случайного процесса. Каждому моделируемому речевому объекту – фразе, слову, слогу, фонеме или аллофону – сопоставляется своя НММ. НММ фразы представляют собой конкатенацию НММ слов, которые представляются конкатенацией НММ более мелких элементов.



Рассмотрим основные задачи, решаемые в рамках НММ [25].

1. Задача оценки: для данной модели  $\lambda(A, B, \pi)$  и последовательности наблюдений  $O = \{o_1, o_2, \dots, o_\tau\}$  вычислить вероятность  $P(O|\lambda)$ , то есть вероятность порождения последовательности  $O$  моделью  $\lambda$ . Решается алгоритмом «Вперёд-Назад» (Forward-Backward).

2. Задача распознавания: для данной последовательности наблюдений  $O = \{o_1, o_2, \dots, o_\tau\}$  и модели  $\lambda(A, B, \pi)$  вычислить оптимальную, в некотором смысле, последовательность состояний  $Q = \{q_1, q_2, \dots, q_\tau\}$ , принадлежащих модели  $\lambda$ . Решается алгоритмом Витерби.

3. Задача обучения: для данной последовательности наблюдений  $O = \{o_1, o_2, \dots, o_\tau\}$  и модели  $\lambda(A, B, \pi)$  подстроить параметры модели так, чтобы максимизировать  $P(O|\lambda)$ . Решается алгоритмом Баума-Уэлша.

В отличие от НММ, гауссовы смешанные модели (Gaussian Mixture Models – GMM) игнорируют временную информацию об акустической наблюдаемой последовательности и содержат состояния, отражающие разные акустические классы.

Для каждой фонемы создается модель, представленная на рисунке 1.4, которая определяет вероятность принадлежности фрейма этой фонеме.

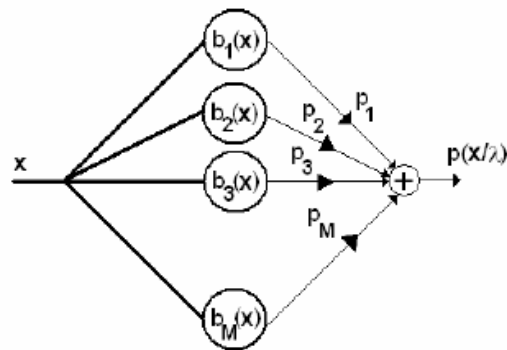


Рисунок 1.4 – Гауссова смешанная модель для одной фонемы

Одна из основных моделей распознавания речи – это модель, комбинирующая модель GMM и НММ. НММ моделируют фонемы языка, при этом выходные вероятности моделируются с помощью GMM [26].

Кроме вышеописанных методик для повышения точности распознавания для GMM-НММ систем применяется ряд следующих методик.

1) Использование контекстно-зависимых фонем. НММ моделируют не изолированные фонемы, а трифоны, т.е. контекст из одного звука слева и справа от моделируемой фонемы. Очевидно, что количество возможных трифонов очень велико, и многие из них могут не встретиться в обучающей выборке. Для решения этой проблемы вместо состояний трифонов используют связанные состояния, или сеноны – состояния трифонов объединяются в группы, каждая из которых получает общий набор параметров гауссовых смесей.

2) Дискриминативное обучение [27]. После стандартного обучения GMM-НММ система дополнительно обучается таким образом, чтобы увеличить правдоподобие истинной гипотезы относительно альтернативных гипотез, что гарантирует «оптимальность» в точности классификации. Дискриминативное обучение эффективнее, чем оценка критерия ML, которая гарантирует «оптимальность» в распределении для порождающей модели. Выделяют ряд методик дискриминативного обучения:

- оценка дискриминативной модели по минимизации частоты ошибок классификации (Minimum Classification Error, MCE) [28];

- обучение в соответствии с критерием максимизации взаимной информации (Maximum Mutual Information, MMI) [29], который выражается как взаимная информация между данными наблюдения  $X$  и последовательностью эталонных слов  $W_r$ ;

- увеличение целевой функции в MMI (VMMI) за счет введения параметра масштабирования и увеличения коэффициента внутри целевой функции MMI [30];

- пространственно-характеристическое и пространственно-модельное VMMI обучение (fVMMI+VMMI) [31], которое в настоящее время является лучшей дискриминативной обучающей схемой для распознавания слитной речи;

- модификация критерия максимизации взаимной информации (Boosted MMI, VMMI) [32].

Стоит также выделить такой метод как подпространство моделей гауссовых смесей (Subspace Gaussian Model Mixtures, SGMM), использование которого улучшило качество распознавания [33]. В SGMM все состояния НММ используют

одну и ту же структуру GMM, с таким же количеством гауссиан в каждом положении, что делает модель компактной.

Преимуществом GMM-HMM перед остальными методами является естественное встраивание времени в модель  $\lambda$ , что позволяет учесть вариативность произнесений по длине и скорости, а также перейти к распознаванию слитной речи.

Однако, несмотря на широкую распространенность в системах распознавания речи, акустические модели на основе GMM-HMM обладают рядом существенных недостатков [34].

1. Они статистически неэффективны для моделирования данных, лежащих близко к границам или на границах нелинейных многообразий.

2. Для повышения скорости распознавания и обучения моделей в GMM-HMM применяются преимущественно смеси с диагональной матрицей ковариации, что влечет за собой необходимость использования некоррелированных признаков. Это не позволяет эффективно учитывать информацию от смежных кадров.

### 1.3 Нейросетевой подход к распознаванию речи

Базовый алгоритм, модель GMM-HMM, была предельно оптимизирована, поэтому новым системам превзойти ее длительное время было невозможно [35]. Однако, вычислительные мощности современных компьютеров сделали возможным использование многослойных нейросетей сложных архитектур с выходным слоем, состоящим из нескольких тысяч нейронов, соответствующих трифонам [36], что повысило точность распознавания.

#### 1.3.1 Основные принципы функционирования глубоких нейросетей

Минимальная вычислительная единица персептрона – искусственный нейрон (Рисунок 1.5а) [37], определяется как линейная функция с весами  $w_j = (w_{0j}, w_{1j}, \dots, w_{Nj})$  от  $N$  аргументов, на которые подается речевой образ  $O =$

$(o_1, o_2, \dots, o_N)$  (Рисунок 1.5б), каждый выходной нейрон сопоставляется с одним из  $\underline{i}$  классов:

$$y_j = w_{0j} + \sum_{i=1}^N w_{ij} o_i = w_j x^T = g_j(x), \quad (1.20)$$

где  $x = (1, o_1, o_2, \dots, o_N)$ .

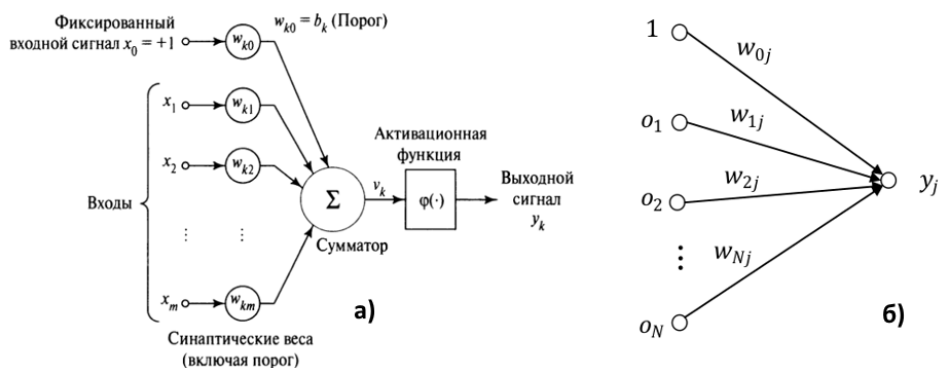


Рисунок 1.5 – Изображение нелинейной модели нейрона (а) и схемы искусственного нейрона однослойного персептрона (б)

Глубокой нейронной сетью (Deep Neural Network, DNN) принято называть ANN с двумя или более скрытыми слоями [38].

У нейронной сети с  $N$  слоями обозначим входной слой как слой 0, выходной слой как слой  $N$ . Для скрытых слоев выполняется:

$$v^n = f(z^n) = f(W^n v^{n-1} + b^n), 0 < n < N, \quad (1.21)$$

где  $z^n = W^n v^{n-1} + b^n \in R^{L^n}$  – вектор индуцированного локального поля;  $v^n \in R^{L^n}$  – вектор активации;  $W^n \in R^{L^n \times L^{n-1}}$  – матрица весов;  $b^n \in R^{L^n}$  – вектор смещения;  $L^n \in R$  – количество нейронов для слоя  $n$ ;  $v$  – вектор признаков;  $L_0 = D$  – размерность вектора признаков;  $f(\cdot): R^{L^n} \rightarrow R^{L^n}$  – функция активации, применяемая поэлементно к вектору индуцированного локального поля. Функция активации для выходного слоя выбирается в зависимости от задачи.

Для задач классификации каждый выходной нейрон отвечает за класс  $i \in \{1, 2, \dots, C\}$ , где  $C = L_N$  – число классов. В этих задачах значение выходного  $i$ -го нейрона обычно вычисляется по формуле:

$$v_i^N = P_{dnn}(i|o) = \text{soft max}_i(z^N) = \frac{e^{z_i^N}}{\sum_{j=1}^C e^{z_j^N}} \quad (1.22)$$

и интерпретируется как вероятность того, что наблюдение  $o$  принадлежит классу  $i$ .

Имея вектор наблюдений  $o$ , выход DNN, определяемой набором параметров  $\theta = \{W, b\} = \{W^n, b^n | 0 < n \leq N\}$ , может быть вычислен посредством последовательного вычисления векторов активации согласно (1.21), начиная со слоя 1 и заканчивая слоем  $N - 1$ , и далее с помощью (1.22) для задач для задач классификации. Этот процесс называют прямым проходом (forward pass).

Процесс обучения DNN называется настройкой параметров  $\theta = \{W, b\}$  по имеющимся обучающим примерам:

$$S = \{(o^m, y^m) | 0 \leq m < M\}, \quad (1.23)$$

где  $M$  – количество примеров;  $o^m, y^m$  – векторы наблюдений и желаемый выходной вектор для  $m$  примера.

### 1.3.2 Технологии обучения нейронных сетей

Процесс обучения характеризуется критерием обучения и обучающим алгоритмом.

В задачах классификации  $y$  представляет собой распределение вероятностей акустических классов, и часто используется критерий минимизации взаимной энтропии (Cross-Entropy, CE):

$$J_{CE}(W, b, S) = \frac{1}{M} \sum_{m=1}^M J_{CE}(W, b, o^m, y^m), \quad (1.24)$$

где

$$J_{CE}(W, b, o^m, y^m) = - \sum_{i=1}^C y_i \log(v_i^N), \quad (1.25)$$

где  $y_i = P_{emp}(i|o)$  – вероятность того, что наблюдение  $o$  принадлежит классу  $i$ ;  $v_i^N$  – вероятность того, что наблюдение  $o$  принадлежит классу  $i$ , вычисленная посредством DNN.

$$y_i = \begin{cases} 1, & c = i, \\ 0, & c \neq i, \end{cases} \quad (1.26)$$

где  $c$  – метка класса в обучающих данных для наблюдения  $o$  (1.25) преобразуется в отрицательный логарифм правдоподобия (Negative Log-Likelihood, NLL):

$$J_{CE}(W, b, o, y) = - \log v_c^N \quad (1.27)$$

При имеющемся обучающем критерии параметры модели  $\{W, b\}$  могут быть обучены с помощью широко известного алгоритма обратного распространения ошибки (Error Backpropagation, BP) [39], заключающегося в использовании правила дифференцирования сложной функции для вычисления градиента.

Для обучения DNN необходимо решить ряд вопросов.

1. Нормализация входных признаков с целью приведения входных данных к близкому диапазону численных значений, что позволяет использовать одну и ту же скорость обучения для всех весов.

2. Инициализация параметров модели. Параметры инициализируются случайным образом, поскольку в противном случае различные нейроны будут определять одни и те же шаблоны признаков на нижних слоях. Согласно [40, 41], для нейронных сетей со скрытыми слоями размера 1000–2000 нейронов, обычно используемых в распознавании речи, эффективно работает инициализация матриц весов гауссовым распределением с нулевым средним и дисперсией 0.05, либо равномерным распределением в диапазоне  $[-0,05, \dots, 0,05]$ . Векторы смещений можно инициализировать нулями.

3. Регуляризация путём добавления регуляризующего слагаемого  $R(W)$  к критерию обучения. Применяется для борьбы с переобучением (overfitting) – явлением, при котором построенная модель хорошо объясняет примеры из обучающей выборки, но плохо работает на примерах, не участвовавших в обучении, что актуально при маленьких размерах обучающей выборки.

4. Выбор размера обучающей порции влияет и на скорость сходимости, и на качество обучения. Простейший способ – брать в качестве обучающей порции все обучающие данные (full-batch training), в этом случае вычисляется точный градиент по обучающим данным. Однако это приводит, во-первых, к низкой скорости обучения, и, во-вторых, к склонности к попаданию в плохой локальный минимум. Альтернативой является метод стохастического градиентного спуска (Stochastic Gradient Descent, SGD) [42], при котором обновление параметров модели происходит после каждого обучающего примера. Неточная оценка градиента в этом случае является преимуществом, а не недостатком, поскольку позволяет избежать

плохих локальных минимумов и переобучения. К недостаткам этого метода можно отнести трудности в распараллеливании и невозможность достижения полной сходимости. Компромиссом между full-batch training и SGD является оценка градиента и обновление параметров модели по малой порции данных, случайным образом выбранной из обучающих примеров (minibatch training). Размер порции, используемый в задачах распознавания речи, обычно составляет 128–1024 примера. Кроме того существуют другие методы улучшения устойчивости и сходимости градиентного спуска: ускоренный градиент Нестерова (Nesterov Accelerated Gradient, NAG) [43]; Adam [44]; RMSProp [45], AdaGrad [46]; AdaBound [47]; Ranger [48].

6. Скорость обучения оказывает существенное влияние на качество обучения нейронной сети [49]. В задачах распознавания речи распространён алгоритм «newbob», который заключается в осуществлении нескольких полных проходов обучения по всем данным (эпох обучения) с постоянной скоростью. Как только абсолютное уменьшение ошибки классификации кадров на кросс-валидационной выборке окажется менее определенного порога, скорость для каждой последующей эпохи уменьшается в несколько раз. Обучение останавливается, как только абсолютное уменьшение ошибки классификации кадров окажется достаточно малым. Другая простая и эффективная техника состоит в уменьшении скорости обучения для следующей эпохи в несколько раз (например, в два раза), если относительное улучшение критерия обучения на кросс-валидационной выборке после текущей эпохи оказалось менее определенного порогового значения.

7. Выбор архитектуры DNN сильно влияет на эффективность ее работы. В задачах распознавания речи [41] обычно применяются глубокие нейронные сети, имеющие 5–7 скрытых слоев по 1000–2000 нейронов в каждом.

Перечислим основные технологии обучения глубоких нейросетей.

1) Для предобучения DNN используются ограниченные машины Больцмана (Restricted Boltzmann Machine, RBM) [50], при этом каждый уровень сети учится как отдельная RBM. Эта технология называется «глубокая сеть доверия» (Deep Belief Network, DBN) [51].



3) Дискриминативное предобучение (discriminative pretraining, DPT) [52], осуществляющееся послойно с использованием размеченных обучающих данных. Целью дискриминативного предобучения является приведение весов модели к хорошему локальному экстремуму. При этом регуляризующий эффект генеративного предобучения отсутствует, поэтому дискриминативное предобучение лучше всего работает на больших объемах обучающих данных.

4) Обучение акустических моделей на основе DNN с использованием критериев разделения последовательностей (Sequence-Discriminative Training, ST). Применение данной методики позволяет достичь 3–17% относительного уменьшения ошибки распознавания по сравнению с DNN, обученными по критерию минимизации взаимной энтропии [53].

Для выделения и настройки подмножества параметров нейронной сети используют следующие техники.

1. Адаптация слоев: линейного скрытого [55], линейного выходного [56] и линейного входного [54]. При адаптации входного слоя параметры всех слоев дикторонезависимой нейронной сети, кроме первого, фиксируются и на данных определенного диктора осуществляется настройка параметров первого слоя.

2. Дискриминативная линейная регрессия в пространстве признаков (feature Discriminant Linear Regression, fDLR) [57]. По сравнению с методом адаптации линейного входного слоя, данный метод имеет меньшее количество настраиваемых параметров, поэтому он меньше подвержен переобучению и демонстрирует лучшее качество работы при наличии небольшого количества адаптационных данных.

3. Использование дикторозависимого слоя. Наиболее чувствительные к междикторской вариативности параметры нейронной сети локализованы в определенном её слое. В работе [58] показано, что наибольшей чувствительностью обладают параметры второго слоя. Схема адаптации состоит из трех этапов. На первом этапе обучается дикторонезависимая нейронная сеть, на втором – выполняется обучение, но для каждого диктора используется индивидуальный набор параметров дикторозависимого слоя (второго). На третьем этапе фиксируются

параметры всех дикторонезависимых слоев, полученные на втором этапе, и по данным целевого диктора настраиваются параметры дикторозависимого слоя.

4. Факторизация параметров нейронной сети и последующее выделение дикторозависимого фактора. Одним из способов для сокращения количества параметров DNN является сингулярное преобразование. При помощи сингулярного преобразования матрицы весов представляются в виде произведения двух матриц, имеющих существенно меньшую размерность по сравнению с исходной – формируется т. н. «узкое горло» (bottleneck). Полученная после факторизации сеть заново обучается. Подобный способ обучения зачастую позволяет не только не ухудшить точность распознавания, но и немного улучшить [59].

Для настройки параметров DNN с использованием в целевой функции дополнительного регуляризирующего слагаемого используют L2-штраф на изменение параметров модели [60] и дивергенцию Кульбака-Лейблера выходного распределения сенонов [61].

Предоставление нейронной сети дополнительной информации о фонограмме или ее участках также является одним из путей к адаптации DNN-HMM. Используют следующие методы.

1. Использование дикторских кодов [62] для быстрой адаптации к диктору. Специально обучаемый дикторский код представляет собой малоразмерный вектор дикторских характеристик, подающийся наряду с акустическими признаками на вход каждого слоя дополнительной входной сети, включая первый. При этом адаптационная сеть учится по всем обучающим данным и не меняется в зависимости от диктора, а дикторские коды обучаются для каждого диктора только по его данным. Такая адаптация осуществляется в режиме работы с учителем, т. е. предполагается наличие текстовых расшифровок и разметки выборки на дикторов.

2. Адаптация при помощи  $i$ -векторов [63], которые содержат канальную и дикторскую информацию.  $i$ -вектор вычисляют по фрагменту фонограммы, соответствующему определенному диктору, и добавляют к вектору акустических признаков [64]. Таким образом, осуществляется адаптация как к диктору, так и к акустической обстановке [65].

3. Использование акустических факторов [66]. Суть метода заключается в выделении из речевого сигнала факторов, характеризующих акустическую обстановку, и добавлении этих факторов на вход выходного слоя нейронной сети.

4. Использование признаков, адаптированных при помощи GMM-НММ моделей [67].

Существенным недостатком алгоритмов выделения и настройки подмножества параметров DNN, а также использования дикторских кодов является то, что они демонстрируют хорошее качество работы только в условиях адаптации с учителем, т. е. при наличии эталонного текста. В реальных задачах это требование часто не выполняется, и применяется адаптация без учителя. А для использования признаков, адаптированных при помощи GMM-НММ моделей, необходимо выполнить предварительный проход распознавания, что приводит к значительному снижению скорости работы системы. Адаптация при помощи  $i$ -векторов работает без учителя и не оказывает существенного влияния на быстродействие, поэтому можно сделать вывод о перспективности этого подхода для разработки системы распознавания слитной русскоязычной речи.

### 1.3.3 Архитектуры глубоких нейронных сетей, используемых для распознавания речи

Существует множество архитектур DNN, выделим основные из них.

1) Контекстно зависимые нейронные сети (Context-Dependent Deep Neural Networks, CD-DNN). DNN обучается таким образом, чтобы предсказывать на каждом кадре признаков апостериорные вероятности. Вход нейронной сети состоит из объединенных векторов признаков нескольких кадров в окне контекста. Такое объединение позволяет улучшить точность классификации за счет использования более широкого временного контекста.

Количество весов и нейронов в скрытых слоях должно дать модели достаточную степень свободы для моделирования речи. Модель в [68] имеет 5 слоёв с 2048 нейронами. Последний слой осуществляет softmax функцию и даёт выходные постериорные вероятности для различных состояний НММ, которые являются сенонами (Рисунок 1.6).

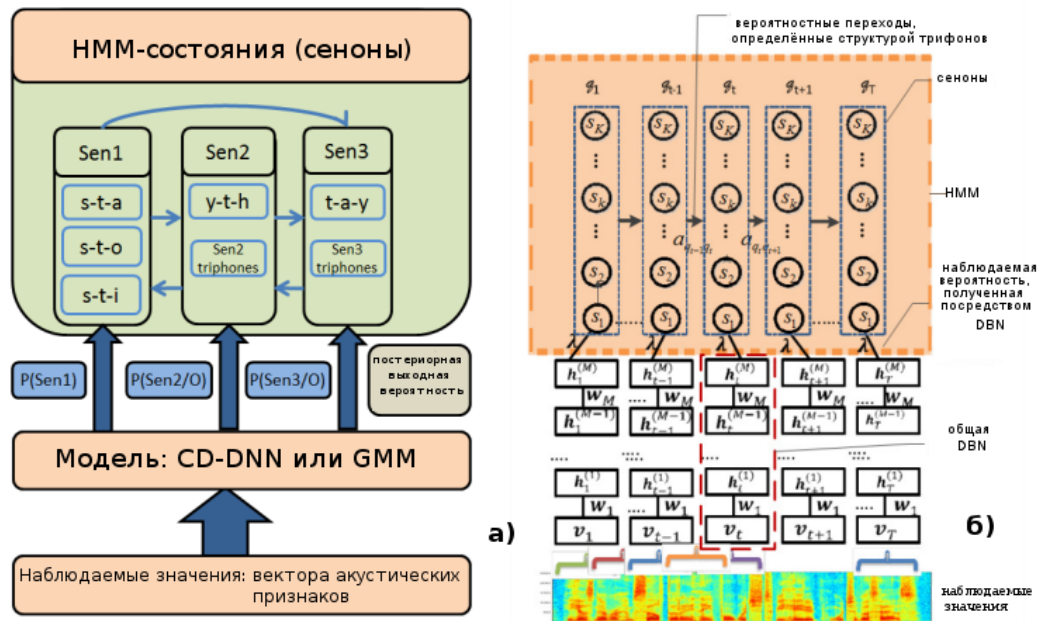


Рисунок 1.6 – Общая (а) и расширенная (б) схемы CD-DNN-HMM

В зависимости от задачи, применение DNN позволяет получить по сравнению с GMM относительное улучшение точности распознавания до 5% (Таблица 1.1).

Таблица 1.1 – Результаты распознавания английской слитной речи для различных речевых корпусов

Тестовая выборка	Кол-во речевых данных, ч	WER(CNN-HMM), %	WER (DNN-HMM), %	WER (GMM-HMM), %
Switchboard (rt03 FSH)	309	14,5	14,9	17
Switchboard (rt03 SWB)	309	22,1	23,5	25,2
English Broadcast News	50	12,0	13,4	13,8
Google Voice Search	5870	-	12,3	16,0
Bing Voice Search	18	33,4	35,4	-

2) Свёрточные нейронные сети (Convolutional Neural Networks, CNN) успешно применяются для многих задач обработки изображений. В [69] концепция CNN используется в частотной области для нормализации дисперсии дикторов для достижения более высокой производительности распознавания речи для нескольких дикторов. Экспериментальные результаты (Таблица 1.1) показывают, что CNN позволяет достичь относительного уменьшения ошибок в независимых

наборах тестов при сравнении с обычным CD-DNN (Рисунок 1.7), с использованием того же количества скрытых слоев и весов.

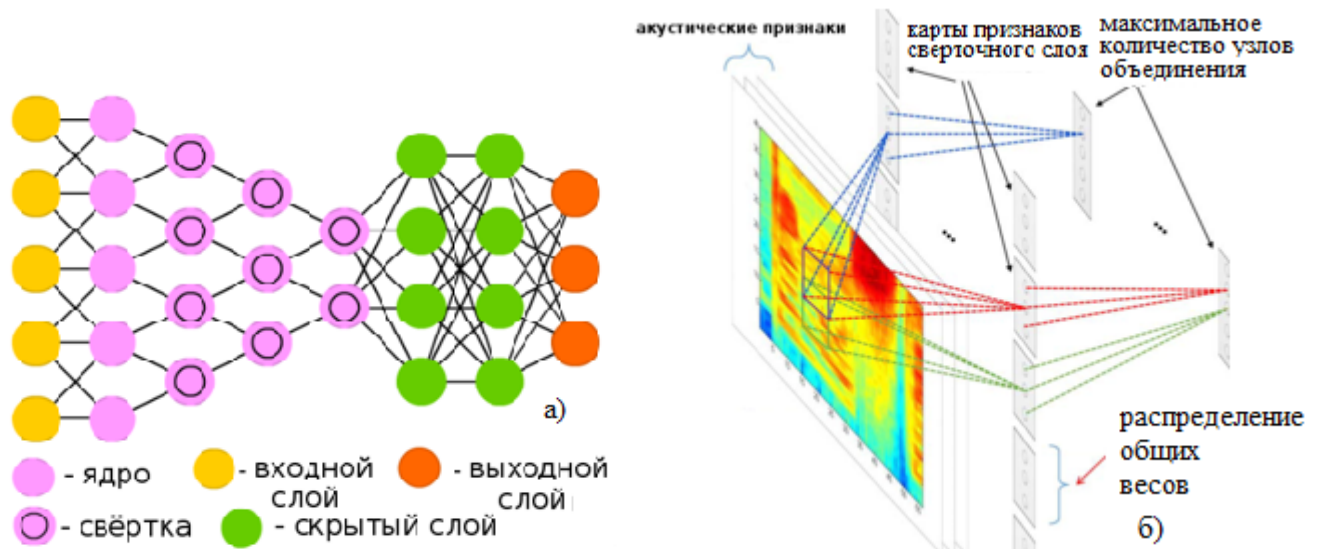


Рисунок 1.7 – Общая схема CNN (а) и схема CD-CNN-HMM (б)

3) Рекуррентные нейронные сети (Recurrent Neural Networks, RNN) содержат соединения с предыдущими слоями (или другими перцептронами того же слоя). Это позволяет использовать итерации для моделирования последовательности входных данных. RNN реализуются путем добавления «блоков памяти», таким образом, состояния предыдущих циклов (вычисленные значения перцептронов) учитываются в более поздних циклах в произвольных положениях. Наличие обратной связи наделяет RNN памятью, благодаря чему появляется возможность моделировать динамические процессы (Рисунок 1.8). Однако обучение RNN является сложной задачей из-за проблем со исчезающими и взрывными градиентами. Это приводит к тому, что долгосрочная информация (из предыдущих циклов) растет экспоненциально, затирая кратковременную информацию, или исчезает.

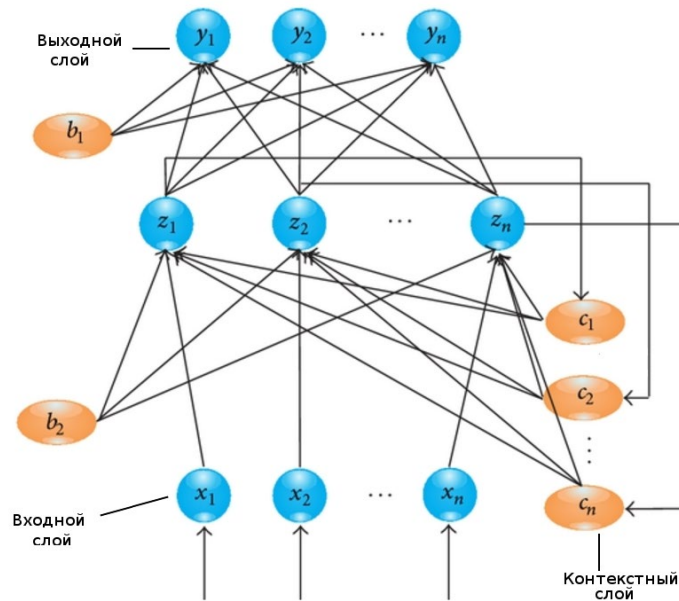


Рисунок 1.8 – Схема рекуррентной нейронной сети

Для разрешения проблемы с исчезающими и взрывными градиентами была предложена архитектура RNN, названная нейросетью с долгой краткосрочной памятью (Long Short-Term-Memory, LSTM) [71], содержащая специальные элементы – блоки памяти. Блоки памяти содержат ячейки, которые хранят временное состояние сети, а также мультипликативные элементы, называемые гейтами (gates), управляющие потоком информации. Каждый блок памяти содержит входной и выходной гейты, а также гейт забывания (Рисунок 1.9).

Базовый принцип LSTM состоит в том, что сигналы ошибки сохраняются внутри блока памяти, называемого «карусель с постоянной ошибкой» (Constant Error Carousel, CEC). Это простой нейрон, который соединён с весом 1, т. е. константой. В этом случае для сигналов ошибки может быть применено обратное распространение ошибки. А гейты «наблюдают» за всеми ячейками памяти и могут принимать решения на каждом этапе:

- входной гейт: каким образом для CEC будет изменен входной слой;
- выходной гейт: каким образом CEC влияет на выходной слой;
- гейт забывания: CEC будет удалён путем обнуления веса CEC.

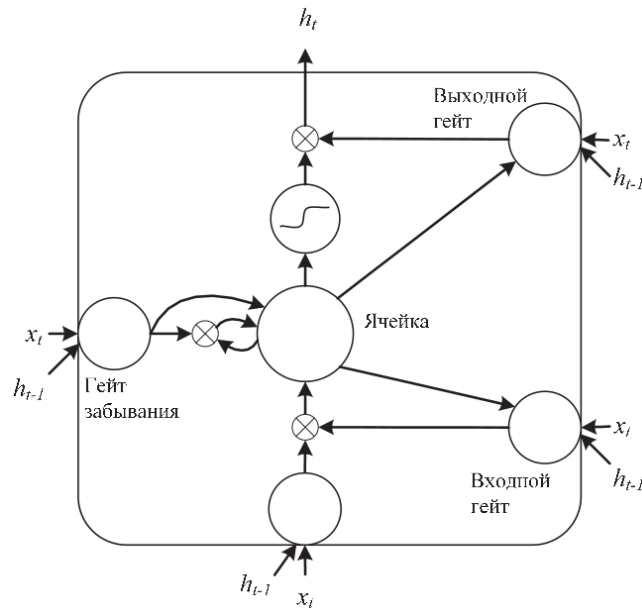


Рисунок 1.9 – Структура блока памяти сети LSTM

Ячейка сети LSTM может рассматриваться как элемент сети, способный запоминать информацию на длительное время. Проблема LSTM заключается в том, что только входные гейты могут изменять СЕС, который зависит только от входных значений. Таким образом, нет гарантии, что конкретная СЕС перестанет влиять на выходной слой. Применение LSTM совместно с GMM-HMM позволяет снизить ошибку распознавания слов по сравнению со стандартной DNN [72].

5) Для акустического моделирования также применяются нейросети с временными задержками (Time Delay Neural Network, TDNN), представляющими собой DNN, узлы которой модифицированы введением временных задержек [73].

6) Главной проблемой DNN является сложность процедуры обучения. Данная проблема называется проблемой недетерминированного полиномиального времени (nondeterministic polynomial time problem). Чтобы обойти данное ограничение была разработана архитектура «глубокой выпуклой нейросети» (Deep Convex Network, DCN) [74], которая состоит из небольших подсетей с одним скрытым слоем. Вывод любой подсети может быть передан на более высокий слой для создания архитектуры сверточного слоя. Причём, каждая подсеть может быть обучаться как индивидуально, так и параллельно, так как ни один входной слой не зависит от



вывода подсетей более низкого слоя. Данная особенность обеспечивает высокий параллелизм в обучении и, следовательно, оптимизирует скорость обучения.

На основе DCN была разработана архитектура «глубинной нейросети с тензором» (Tensor Deep Stacking Network, T-DSN) [75]. Основной особенностью T-DSN является расположение скрытых слоев: можно разделить скрытый слой, чтобы параллельно работать с двумя скрытыми слоями; затем они снова объединяются трехмерным тензором вместо двумерной весовой матрицы. Возможность конкатенации входных данных (или вывода большего количества T-DSN) в разных точках дает высокую степень мощности моделирования, а также возможность для дальнейшей оптимизации.

В последнее время возникают попытки построения end-to-end систем распознавания речи, использующих только DNN, без использования GMM-HMM. End-to-end системы состоят из двух модулей: кодировщика и декодера. Кодировщик читает входной сигнал, вычисляет признаки сигнала и преобразует его в промежуточное параметрическое представление. Декодер преобразует параметрическое представление сигнала в последовательность символов. Обычно end-to-end системы строятся на базе CNN и метода нейросетевой темпоральной классификации (Connectionist Temporal Classification; CTC)[76, 77], а в качестве признаков используют образы спектрограмм. WER при использовании данного подхода составляет 18,2%. В работе [78] совместно с CTC применяют сети LSTM: без использования лингвистической информации WER = 27,3%; при применении словаря WER = 21,9%, при применении 3-gram модели языка WER = 8,2%.

Главными преимуществами нейросетей является: способность к выделению характерных черт образа и обобщению; более высокой по сравнению с GMM-HMM точности распознавания слитной речи; возможность применения обучения без учителя; оптимальная адаптация акустических признаков как под дикторов, так и под окружение. Минусы нейросетевых подходов: необходимость в высокой вычислительной мощности на стадии обучения и размеченных речевых корпусов длительностью в десятки тысяч часов; применение нейросетевых подходов существенно осложняет анализ и тестирование системы распознавания речи.

### 1.3.4 Технологии обучения акустических моделей на основе глубоких нейросетей

ANN является сложной моделью, состоящей из каскада нелинейных преобразований входных признаков и линейного классификатора. Каждый скрытый слой, как правило, реализует простое нелинейное преобразование, а стек таких преобразований может моделировать различные недетерминированные закономерности. Таким образом, комбинация скрытых слоев глубокой нейронной сети является неким блоком для извлечения информативных признаков.

Скрытые слои глубокой нейронной сети извлекают из неструктурированных входных данных внутренние представления, которые затем преобразуются в такое расположение точек в гиперпространстве, чтобы выходной линейный слой смог их легко разделить на классы. Информативные признаки, как правило, хранятся на последнем (или предпоследнем) скрытом слое ANN. Эти признаки должны определять шаблоны, оказывающие большое влияние на модель, поскольку обеспечивают хорошую разделимость классов. Эти шаблоны, как правило, непонятны и неразличимы человеком. При этом информативные признаки, полученные с помощью ANN, инвариантны к изменениям входных признаков [79].

После ряда нелинейных преобразований в  $N-1$  скрытых слоях ANN признаки становятся более устойчивыми по отношению к темпу речи, акустической среде, междикторской вариативности и т. п. За счет этого внутренние представления, извлекаемые скрытыми слоями глубокой нейронной сети из входных признаков, становятся менее чувствительными к малым возмущениям входного сигнала с ростом числа скрытых слоев.

Учитывая вышесказанное, наиболее перспективным способом обеспечения робастности акустической модели является применение информативных признаков, извлечённых из скрытых слоёв.

Первые упоминания использования высокоуровневых признаков изложены в работе [80]. В данной работе предложен подход, именуемый тандемным, который комбинирует нейросетевую параметризацию и GMM-HMM. Его суть состоит в том,

что высокоуровневые признаки, полученные после обучения ANN, подаются как входной вектор для обучения GMM-HMM АМ. Вектора акустических признаков подаются вход нейросети, в результате чего вычисляется распределение вероятностей целевых классов. Это распределение вероятности дополняет вектор акустических признаков, а при помощи метода линейного дискриминативного анализа уменьшается размерность расширенного вектора акустических признаков. Недостатком данного подхода является тот факт, что получаемые вектора обладают слишком большой размерностью.

В качестве альтернативы тандемному подходу в работе [19] предложен метод извлечения bottleneck-признаков – признаков, извлекающихся из «узкого горла» (скрытого слоя небольшой размерности с линейной функцией активации, расположенной в середине или возле последних скрытых слоёв).

В настоящее время ряд авторов [81–83] исследовали влияние совместного применения обычных акустических признаков, таких как MFCC и  $i$ -вектора [63]. В этих работах для совмещённых признаков используется понижение размерности и декорреляция, после чего полученные признаки вновь используются в процессе обучения GMM-HMM АМ (Рисунок Б.1).

Помимо вышеописанных подходов, существует подход использования bottleneck-признаков второго уровня [84]. При этом подходе используются bottleneck-признаки, извлечённые с некоторым контекстом (шагом) для обучения bottleneck-ANN второго уровня.

#### 1.4 Языковое моделирование

Задачей языкового моделирования является определение вероятности последовательности слов  $w = (w_1, w_2, \dots, w_m)$ . Наиболее распространённым подходом к языковому моделированию являются статистические модели на основе  $n$ -грамм [21], представляющих собой последовательности из  $n$  слов. Т.е. необходимо предсказать следующее слово по известным  $n-1$  предыдущим словам. Причём для получения надежных оценок распределений параметр  $n$  должен быть

достаточно мал: униграммы (1-gram), биграмммы (2-gram) или триграммы (3-gram) соответственно. N-граммные модели определяют вероятность появления цепочки слов при помощи правила Байеса, для вычисления вероятности n-gram используется оценка ML [85].

Методики глубокого обучения и применение рекуррентных нейросетей для обработки текстов существенно улучшают качество ЯМ за счет учета контекста и отсутствия ограничений на использование только  $n$  предыдущих слов.

В RNN [88] скрытый слой хранит всю предыдущую историю, таким образом, размер контекста неограничен. Проблема исчезающего градиента в RNN решается с помощью модификации – LSTM. Сети LSTM нашли широкое применение для построения ЯМ [89].

ЯМ, базирующаяся на архитектуре Transformer, использует полносвязные слои в качестве стандартных архитектур для энкодера и декодера, а также и механизм многослойного обучающего внимания (multi-head attention) в энкодере [91]. Multi-head attention – новый слой, который дает возможность каждому входному вектору взаимодействовать с другими через механизм внимания, вместо передачи скрытого состояния как в RNN.

На основе Transformer была разработана GPT2 [92], которая использует в качестве входных данных не токены, а части токенов. Кроме того, в GPT-2 каждый слой обучающего внимания хранит соответствующие значения векторов для каждого токена. Полносвязный слой в GPT2 состоит из двух уровней (слоёв). Первый уровень имеет размерность, которая в четыре раза больше размера модели. Данная размерность позволяет модели иметь достаточную репрезентативную способность. Второй уровень проецирует результат из первого уровня обратно в размер модели. Основной недостаток GPT2 – огромное число параметров.

## 1.5 Генерация транскрипций слов

Выделяют два подхода к решению задачи фонемного транскрибирования: традиционный на основе знаний; статистический на основе данных. Методы традиционного подхода используют словарь или набор лингвистических правил

[93, 94], сформированные экспертом-лингвистом. Методы статистического подхода [95, 96] заключаются в обучении алгоритма транскрибирования по словарю, содержащем буквенные и фонемные формы представления слов. Недостаток первого подхода заключается в ограниченности словаря и необходимости ручного составления набора правил, требующих периодического пересмотра и обновления. Недостатком второго подхода является зависимость качества транскрибирования от обучающих данных.

Как правило, в качестве формата представления транскрипций используют формат международного фонетического алфавита (International Phonetic Alphabet, IPA) [97].

Общедоступные системы автоматического формирования транскрипций не учитывают всех особенностей русского языка, в том числе произношение иностранных слов, слов с апострофами, слов-исключений. В случае, если слова нет в словаре транскрипций, невозможно точно сгенерировать транскрипцию.

Среди алгоритмов фонемного транскрибирования, относящиеся к группе статистического моделирования, наилучшие результаты показывает нейросетевой подход, подразумевающий наличие извлечённых из набора обучающих данных статистических зависимостей. В качестве обучающих данных обычно выступает словарь слов с их фонемными транскрипциями. На основе обучающего словаря происходит сопоставление букв с фонемами одного слова (задача графемно-фонемного выравнивания).

В качестве одной из основных нейросетевых архитектур, использующихся для задачи графемно-фонемного выравнивания, выделяют seq2seq (sequence-to-sequence) [98], базирующаяся на архитектуре RNN. Seq2seq (Рисунок 1.10) состоит из двух RNN: энкодер для обработки входных данных и декодер для генерации выходного значения. Энкодер преобразует входную последовательность данных  $X$  в свое непрерывное представление  $Y$ , которое, в свою очередь, используется декодером для генерации вывода, по одному символу за раз.

Конечным состоянием кодировщика является контекстный вектор (embedding) фиксированного размера, который должен кодировать входной

документ, используя предобученную модель. Декодер использует полученный контекстный вектор для генерации выходных данных.

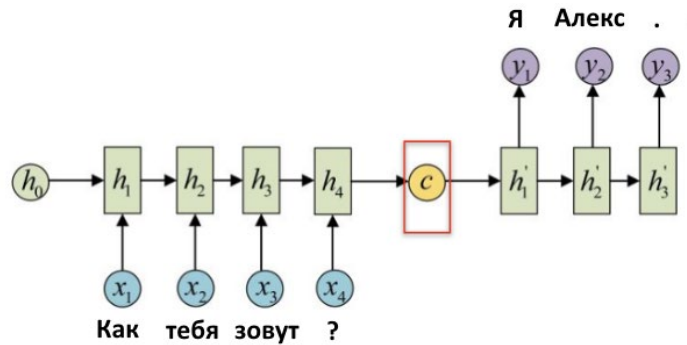


Рисунок 1.10 – Схема работы seq2seq модели

Основным недостатком модели seq2seq является неспособность запоминать длинные предложения. Эта проблема была решена за счёт использования модели с вниманием [99], создающей контекстный вектор для каждого входного слова.

Однако, модель с вниманием имеет ряд недостатков: контекстный вектор вычисляется через скрытое состояние между исходным и целевым документом, не учитывая контекст внутри исходного и целевого предложения; модель ресурснозатратна, сложна в распараллеливании.

Наряду с моделью seq2seq для обработки последовательностей применяют архитектуру Transformer, которая также использует слои multi-head attention и работает не хуже рекуррентных сетей [90]. Кроме того, модель распределяет свое внимание равномерно между частями последовательности, в отличие от LSTM, обращающей больше внимания на непосредственно предшествующие части. Помимо этого, Transformer лучше работает в многопоточном режиме, т.е. обучение с использованием графических процессоров будет быстрее.

## 1.6 Декодирование

Декодер обрабатывает вероятности, генерируемые АМ и ЯМ, и выдает в качестве результата распознавания последовательность слов:

$$\hat{w} = \arg \max_w (w|x) = \arg \max_w \frac{P(x|w)}{P(x)} = \arg \max_w P(x|w)P(w), \quad (1.32)$$

где  $w$  – цепочка возможных слов;  $x = [x_1, x_2, \dots, x_T]$  – набор векторов признаков распознаваемого сигнала для GMM-НММ или для DNN-НММ;  $P(w)$  – генерируемая языковой моделью вероятность цепочки слов;  $P(x|w)$  – генерируемая акустической моделью вероятность:

$$P(x|w) = \sum_q P(x|q, w) \approx \max_{q|w} \pi(q_0) \prod_{t=1}^T a_{q_{t-1}q_t} \prod_{t=0}^T P(x_t|q_t). \quad (1.33)$$

Для поиска максимально правдоподобной последовательности состояний для НММ используется алгоритм Витерби.

В DNN-НММ системах нейросеть вычисляет апостериорную вероятность  $P(q_t|x_t)$  вместо требуемого декодеру правдоподобия  $P(x_t|q_t)$ , применяя теорему Байеса.

Для нормирования вероятностей, генерируемых АМ и ЯМ, обычно используется языковой вес  $\lambda$ , и добавляется штраф за вход в слово  $w_{penalty}$ , что позволяет избежать разбиения длинных слов на большое количество коротких и таким образом способствует повышению точности распознавания речи. При этом результат работы декодера – последовательность слов  $\hat{w}$ , называемая лучшей гипотезой (1-best):

$$\hat{w} = \arg \max_w [\ln P(x|w) + \lambda \ln P(w) - w_{pen} n(w)], \quad (1.37)$$

где  $n(w)$  – количество слов в цепочке  $w$ .

Результатом работы декодера также может быть список из  $n$  гипотез с наибольшими значениями  $\hat{w}$  ( $n$ -best), или словная сеть (word lattice) – направленный ациклический граф с одной начальной точкой, каждое ребро которого содержит слово, а также его акустическую, языковую и итоговую вероятность (в этом случае лучшая гипотеза представляет собой путь в графе с наибольшей итоговой вероятностью).

Для достижения приемлемой скорости работы декодера применяются различные методики ограничения пространства поиска гипотез. Одна из них, называемая ограничением луча поиска, на каждом кадре выбрасывает из рассмотрения все гипотезы, значение логарифма правдоподобия которых меньше соответствующего значения для лучшей на данный момент гипотезы более чем на



постоянную величину, называемую шириной луча поиска. Согласно другой методике, на каждом кадре декодером рассматривается только  $n$  гипотез, имеющих максимальное значение логарифма правдоподобия. Помимо этого, применяется техника, в которой расширение сети интегрировано в процесс декодирования. Построенное на старте распознавания начальное дерево расширяется, используя виртуальные узлы и временные структуры, содержащие только информацию, необходимую для текущей гипотезы.

Так как GMM-HMM или DNN-HMM получается огромной (она содержит лексикон, ЯМ, фонетическое дерево решений, топологию фонем HMM), поэтому обычно применяют подходы к статическому декодированию, основанные на конечных автоматах, оптимизированных с помощью: преобразователей с конечным числом состояний или взвешенных преобразователей с конечным числом состояний [100]. Последние имеют преимущество – их использование для декодирования позволяет значительно снизить WER, не изменяя ЯМ и АМ [101].

## 1.7 Выводы к главе 1

Анализ основных технологий разработки составных частей современных ASR-систем позволил сделать следующие выводы:

1. «Узким» местом современных систем распознавания русской слитной речи является обеспечение устойчивости по отношению к акустической вариативности и смене диктора. Кроме того, общедоступные системы автоматического формирования транскрипций не учитывают всех особенностей русского языка.

2. На данный момент не существует достаточного количества аннотированных русскоязычных речевых и нормализованных текстовых корпусов, находящихся в открытом доступе. Поэтому необходима разработка методики формирования речевого и текстового корпусов для обучения АМ и ЯМ.

3. Для построения АМ, наиболее перспективным представляется использование нейросетевой параметризации речевого сигнала, что повысит

робастность системы распознавания, компенсируя несоответствие условий обучения и эксплуатации. Помимо того, с целью повышения точности распознавания и робастности следует использовать техники модификации акустических признаков, а также адаптивные и дискриминативные характеристики.

4. Для уменьшения признакового пространства в АМ необходимо использовать LDA и bottleneck-признаки – признаки, которые извлекаются из скрытого слоя небольшой размерности, расположенного в середине или возле последних скрытых слоёв.

5. Современные методы языкового моделирования для построения ЯМ используют нейросети с архитектурой LSTM и Transformer, позволяющие учитывать контекст без ограничений на использование только *n* предыдущих слов, а для декодирования – подходы, основанные на конечных автоматах, оптимизированных с помощью преобразователей с конечным числом состояний.

6. Разработка новых методов и моделей для систем дикторонезависимого распознавания слитной русской речи, позволяющих учитывать особенности русского языка, адаптирующихся под любую предметную область и обеспечивающих инвариантность АМ к смене диктора и акустической обстановке, позволит повысить точность распознавания речи.

## ГЛАВА 2

# ПОДГОТОВКА ДАННЫХ ДЛЯ ОБУЧЕНИЯ АКУСТИЧЕСКОЙ И ЯЗЫКОВОЙ МОДЕЛЕЙ

Любая ASR-система работает в двух режимах: обучение и распознавание. В режиме обучения создаются модели акустических единиц речи, модель языка, а также словарь транскрипций и словоформ, которые далее будут использоваться распознавателем. Для обучения акустических моделей используются аннотированные речевые базы, а модель языка создается по текстовому корпусу. Одна из основных проблем создания качественной АМ является отсутствие достаточного количества аннотированного речевого материала для русского языка, нормализованных текстовых данных для статистического моделирования языка также недостаточно. Возникает необходимость сбора речевого и текстового материала и предварительной обработки этих данных для качественного обучения АМ и ЯМ. В данной главе описан алгоритм проверки соответствия текстовых расшифровок и аудио, позволяющий автоматизировать процесс создания речевого корпуса русской речи, а также приведено описание методов аугментации, нормализации русскоязычного текста, используемых для обучения ЯМ, и системы автоматического исправления ошибок.

### 2.1 Описание речевых и текстовых данных для обучения акустической и языковой моделей

#### 2.1.1 Описание речевых данных

Речевые данные имеют следующие характеристики: wav-PCM, 16 кГц, 16 бит. База речевых данных состоит из следующих корпусов: корпус аудиокниг; корпус радиозаписей; корпус выступлений (подкастов); дикторский корпус. Общая характеристика речевых данных приведена в таблице 2.1.

*Корпус аудиокниг* состоит из аннотированных речевых данных, сформированных из аудиокниг. Данный корпус содержит эмоциональную речь, в записях наблюдается смена тональности голоса и темпа речи. Записи сделаны в различных условиях с использованием большого диапазона техники (микрофоны, звуковые карты).

Таблица 2.1 – Общая характеристика речевых данных

	Диктор-мужчина	Диктор-женщина
Общее количество записей	231	91
Общая продолжительность, мин	1143	637

*Корпус радиозаписей* состоит из аннотированных речевых данных, сформированных из записей радиоэфиров. В данном корпусе содержится речь разных дикторов с эмоциональной окраской и сменой тональности.

*Корпус выступлений (подкастов)* состоит из аннотированных речевых данных, извлечённых из записей выступлений, подкастов, стриминговых (поточковых) платформ. Стоит отметить, что данный корпус заранее оцифрован потоковыми сервисами. Данный корпус содержит записи с различными шумами, сделанные в разных помещениях. Т.е. условия записи отличаются от эталонных (запись в закрытом помещении, без наличия посторонних звуков и т.п.).

*Дикторский корпус* состоит из аннотированных данных, сформированных из непрофессиональных дикторских баз. Данный корпус содержит речь различных дикторов, записанную с помощью обширного диапазона звукозаписывающего оборудования.

Использование вышеуказанных корпусов для обучения АМ позволит обеспечить её дикторнезависимость и адаптировать АМ под шумы, вызванные различными каналами записи и акустическим окружением.

## 2.1.2 Описание текстовых корпусов

Для обучения языковых моделей составлена текстовая база нормализованных текстов, предназначенных для работы со слитной речью, объемом около 18 Гб. Ниже перечислены ресурсы, на основе которых сформирована текстовая база.

**Корпус новостных лент.** Состоит из текстов, полученных из сайтов, содержащих новостные ленты. Данный корпус играет важную роль при формировании языковой модели. Для данного корпуса выявлены следующие проблемы: отсутствие букв «ё» и «й» (вместо них употребляются символы «е» и «и»); цифробуквенные комплексы, аббревиатуры, сокращения; строки, написанные полностью или частично на языке отличном от русского.

**Литературный корпус.** Данный корпус основан на текстах, полученных из литературных баз (книги, журналы и т.п.). Во время составления текстов для данной выборки возник ряд проблем: наличие книг на разных языках; наличие книг, не несущих контекстной информации (например, книги с иллюстрациями картин); наличие книг разных жанров; отсутствие букв «ё» и «й»; повторяющиеся книги; ошибки в словах; цифробуквенные комплексы, аббревиатуры, сокращения; строки, написанные полностью или частично на языке, отличном от русского.

**Корпус соцсетей.** Является одним из основных корпусов для составления ЯМ. Данный корпус состоит из сообщений между пользователями известных соцсетей, для извлечения которых использовались их собственные API (VkAPI, Facebook API и т.п.). При обработке данного корпуса выявлены следующие проблемы: строки, не несущие контекстной информации (email, ссылки на интернет-ресурсы; хештеги; упоминание ником пользователей и т.п.); большое количество опечаток; отсутствие «ё» и «й»; цифробуквенные комплексы, аббревиатуры, сокращения; наличие строк, написанных полностью или частично на языке отличном от русского. Для решения вышеописанных проблем использовались регулярные выражения, а также ANN для определения языка, описанная в пункте 2.4.1. Целесообразно также разработать ANN-модель для автоматического определения и исправления опечаток [102, 103].

**Корпус субтитров.** Является одним из основных корпусов для составления ЯМ. Данный корпус состоит из текстов, извлечённых из файлов субтитров, хранящихся на интернет-ресурсах. При формировании данного корпуса выявлены следующие проблемы: наличие строк, не несущих контекстной информации (обозначение таких действий, как смех, и т. п.; название фильмов, серий; электронная почта; имя составителей субтитров); опечатки в текстах; отсутствие «ё» и «й»; повторяющиеся субтитры; наличие цифробуквенных комплексов, аббревиатур сокращений; наличие строк, написанных полностью или частично на языке отличном от русского.

**Корпус текстовых расшифровок.** Данный корпус является основным для формирования ЯМ. Он основан на текстовых расшифровках выступлений, радиопередач и т. п. Для данного корпуса выявлены следующие проблемы: отсутствие «ё» и «й», наличие строк, не несущих контекстной информации; наличие цифробуквенных комплексов, аббревиатур сокращений.

Таблица 2.2 – Общая характеристика речевых данных

Общее количество слов	Общее количество предложений
115442870	9057388

Основными проблемами, которые встречаются в текстовых корпусах являются: отсутствие символов «ё» (проблема ёфикации) и «й» (проблема йфикации); наличие цифробуквенных комплексов, аббревиатур, сокращений; наличие иностранных слов; наличие неконтекстных слов/строк.

Для обучения АМ необходимо создать аннотированный речевой корпус из имеющихся речевых данных с расшифровкой. В связи с этим разработан алгоритм проверки соответствия аудио и текстовых расшифровок.

## 2.2 Модификация алгоритма Смита-Ватермана для проверки соответствия текстовых расшифровок и аудио

Алгоритм Смита-Уотермана [104], являющийся алгоритмом парного выравнивания (sequence alignment), как правило, применяется в сфере биоинформатики для нахождения сходств (локальное выравнивание) между ДНК. Помимо задач из сферы биоинформатики данный алгоритм можно применять для нахождения последовательностей, которые, как предполагается, имеют сходство, при этом природа элементов последовательностей может быть различной. Алгоритм Смита-Уотермана находит локальные паттерны с высоким уровнем сходства. В рамках автоматического распознавания речи алгоритм Смита-Уотермана применяют для сравнения строк – эталонной и строки текста, полученной в результате распознавания устной речи, что позволяет исправлять ошибки при распознавании.

В рамках алгоритма Смита-Уотермана нахождение оптимального выравнивания сводится к решению задачи динамического программирования. Входные данные: последовательности символов  $q1$  (эталон),  $q2$  (последовательность, подаваемая на сравнение после распознавания), длиной  $n$  и  $m$  соответственно;  $score\_match$  – размер «поощрения», добавляемое каждый раз при совпадении элементов последовательностей  $q1$  и  $q2$ ;  $score\_pen$  – размер штрафа, добавляемый за несовпадение элементов последовательностей  $q1$  и  $q2$ . Промежуточные переменные – матрица оценок  $M = \{m(i, j)\}_{i=1, j=1}^{m+1, n+1}$ , элементы которой соответствуют накопленным очкам, добавляемым в результате поощрения или штрафа в ходе сравнения последовательностей  $q1$  и  $q2$ .

Классический алгоритм Смита-Уотермана состоит из следующих этапов.

1. Инициализация: элементам первой строки и первого столбца матрицы  $M$  присваивается нулевое значение, определяются значения поощрений и штрафов:

$$m(1, j) = 0 \text{ для } j = 1, \dots, n+1;$$

$$m(i, 1) = 0 \text{ для } i = 1, \dots, m+1.$$

$$score\_match = 2; score\_pen = -1.$$

2. Вычисляются элементы матрицы  $M$ :

$$m(i, j) = m(i-1, j-1) + score, \quad (2.1)$$

$$\text{где } score = \begin{cases} score\_match, & \text{если } q1(j) = q2(i) \\ score\_pen, & \text{если } q1(j) \neq q2(i) \end{cases}.$$

3. Обратный ход для поиска локальных паттернов с высоким уровнем сходства (трассировка). Трассировку нужно начинать с  $m(i_1^*, j_1^*)$  элемента матрицы  $M$  с максимальным значением. Итеративно получаем последовательность индексов элементов матрицы  $M$ . На итерации  $l$  ищется максимальный элемент среди соседних элементов с  $m(i_l^*, j_l^*)$ :

$$(i_l^*, j_l^*) = \operatorname{argmax} \left( m(i_{l-1}^*, j_l^* - 1), m(i_{l-1}^* - 1, j_{l-1}^*), m(i_{l-1}^* - 1, j_{l-1}^* - 1) \right). \quad (2.2)$$

4. Последней является итерация  $k$ , если  $m(i_k^*, j_k^*) = 0$ .

Выходными данными алгоритма (результатом выравнивания) является подпоследовательность эталонной последовательности  $q1$ , содержащаяся в последовательности  $q2$ :

$$q1(j_k^*), q1(j_{k-1}^*), \dots, q1(j_1^*). \quad (2.3)$$

Рисунок 2.1 демонстрирует результат обратного хода.

-		К	Н	И	Г
-	0	0	0	0	0
К	0	2	1	0	0
Н	0	1	4	3	2
И	0	0	3	6	5
Г	0	0	2	5	8
А	0	0	1	4	7

Рисунок 2.1 – Результат обратного хода для  $q2$  «книга» и  $q1$  «книг»

Недостатком классического алгоритма Смита-Уотермана является тот факт, что для ряда задач, таких как выравнивание результатов распознавания речи, результатом выравнивания может стать не слово, а лишь его часть, в то время как модифицированный алгоритм позволяет этого избежать. Для вышеописанной задачи эталон  $q1$ , состоящий из единой строки, образуется посредством считывания множества массивов текстовых объектов, отделённых друг от друга в строке знаком



пробела (« », *delimiter*). Таким образом, известно, что начало и конец последовательностей строк отделяются символом пробела.

Модифицированный алгоритм Смита-Уотермана отличаются следующим. Используем алгоритм трассировки из классического алгоритма, но запоминаем начало (*beg\_query*) и конец (*end\_query*) совпадения в исходных данных, а также ищем не одно максимальное, а  $n$  максимальных, в том случае, если есть несколько вхождений с одинаковыми максимальными оценками.

$$beg\_query=pos(q1(j_k^*)), end\_query=pos(q1(j_1^*)). \quad (2.4)$$

Если *beg\_query* не является первым символом строки  $q1$ , то проверяем является ли *beg\_query*-1 символом *delimiter*. Если да, то началом результата выравнивания является *beg\_query*. Иначе – сдвигаем начало результата выравнивания, на предыдущий символ, пока не встретим *delimiter* или не попадём на первый символ. Если *end\_query* не является концом строки  $q1$ , то проверяем является ли *end\_query*+1 символом *delimiter*. Если да, то концом результата выравнивания является *end\_query*. Иначе – сдвигаем конец результата выравнивания на следующий символ, пока не достигнем *delimiter* или конца строки  $q1$ . В случае, если в результате выравнивания имеются несколько вхождений с одинаковыми максимальными оценками – выполняем для каждого паттерна шаги 2, 3. Затем выбираем вхождение с наименьшим количеством символов.

С целью оценки эффективности предложенной модификации алгоритма проведен ряд экспериментов выравнивания для результатов распознавания речи. Используемая ASR-система основана на статистическом моделировании: акустическая модель обучена с использованием GMM-HMM (гауссовых смесей на основе скрытых Марковских моделей) подхода с применением дискриминационного обучения, а также глубоких нейронных сетей. Языковая модель построена с использованием 3-gram (общее количество 1-gram: 410191; 2-gram: 5347632; 3-gram: 3899721) на текстах, извлечённых из дампа Википедии, а также из новостных лент; словарь использовался на основе 1-gram из языковой модели. В качестве обучающих данных использовался исправленный корпус VoxForge [105], общая продолжительность аудио составляет около 15 ч. В качестве

тестовых данных использовались аннотированные записи 2 мужских дикторов (по 200 аудиозаписей для каждого диктора; текст один и тот же). Результаты распознавания, а также выравнивания их результатов отображены в таблице 2.3, где WER – процент верно распознанных слов; SER – процент верно распознанных предложений (фраз); sw\_o – правка результатов распознавания при помощи классического алгоритма Смита-Уотермана; sw\_m – правка результатов распознавания при помощи модифицированного алгоритма Смита-Уотермана; Dict1, Dict2 – 1-й и 2-ой дикторы. Помимо этого, в таблицах 2.4 и 2.5 приведены примеры распознавания и соответствующих правок, где source\_right – оригинальная последовательность слов (что распознавалось); target – результат распознавания.

Таблица 2.3 – Результаты распознавания и их выравниваний

	WER(target)	SER(target)	WER(sw_o)	SER(sw_o)	WER(sw_m)	SER(sw_m)
Dict1	37,26 %	77 %	14,25 %	34 %	4,62 %	15 %
Dict2	40,89 %	83,5 %	21,19 %	47 %	6,73 %	25

Таблица 2.4 – Примеры результата распознавания и последующих выравниваний для диктора 1

Source_right	Dict1(target)	Dict1(sw_o)	Dict1(sw_m)
молох	Запах	задач	задачах
об интеллекте	Интеллекте	интеллекте	интеллекте
автоматическое распознавание речи	автоматического распознавания речи	автоматическое распознавание речи	автоматическое распознавание речи

Модифицированный алгоритм Смита-Уотермана не ограничен длиной строки, а также показывает более высокую точность выравнивания по сравнению с классическим; из недостатков можно выделить тот факт, что данный алгоритм обладает меньшим быстродействием, по сравнению с оригинальным, из-за наличия дополнительных операций.

Таблица 2.5 – Примеры результата распознавания и последующих выравниваний для диктора 2

Source_right	Dict2(target)	Dict2(sw_o)	Dict2(sw_m)
Молох	Мало	моло	Молох
Об интеллекте	пол интеллекте	об интеллекте	об интеллекте
Автоматическое распознавание речи	Автоматического распознавания речи	Автоматическое распознавание речи	Автоматическое распознавание речи

Среди общих недостатков, которые присущи как классическому, так и модифицированному алгоритму Смита-Уотермана, можно выделить следующие.

1) В случае, если в target количество несоответствий с source (последовательность исходных данных) составляет  $\geq 70\%$ , то результат выравнивания не даст улучшения.

2) В случае, если в source есть сложное слово (более одного корня), то в target вместо одного сложного слова находятся два слова, одно из которых не соответствует одному из корней. Тогда, если в множестве source есть похожее вхождение одного из двух слов – результат выравнивания может быть неверным.

3) Проблемы с падежами, родом, временем и т. п. В том случае, если в source встречается последовательность слов, отличающаяся вышеописанными признаками, при условии, что target задан только этой проблемной последовательностью, то выравнивание не даст улучшения. Пример: source\_right – «автоматическое распознавание речи»; source\_false (неверный вариант из последовательности source) – «методы автоматического распознавания речи»; sw\_m – «автоматического распознавания речи».

4) В случае, если падеж, род, время target не соответствуют source\_right и такая последовательность встречается в других вхождениях source, то выравнивание не даст улучшения. Например: source\_right – «нейронные сети»; source\_false – «подход к определению параметров нейронной сети»; sw\_m – «нейронной сети».

5) Если в результате распознавания пропущено слово, как правило, являющееся коротким словом, то при выравнивании это слово не будет

учитываться. Пример: `source_right` – «об интеллекте»; `target` – «интеллекте»; `sw_m` – «интеллекте». Стоит отметить, что недостатки, описанные в пунктах 3, 4, 5 могут быть устранены за счёт внедрения меток начала и конца (например, «<» и «>») с соответствующими изменениями в модифицированном алгоритме Смита-Уотермана.

Для качественного обучения АМ необходим большой объем речевых данных, имеющегося объема недостаточно. Для увеличения выборки используют метод создания дополнительных обучающих данных из имеющихся путем их модификации. Этот важный этап предварительной обработки данных для обучения моделей называется аугментацией данных.

### 2.3 Разработка метода аугментации речевых данных

Предлагаемая техника аугментации использует модификацию имеющихся речевых данных путем зашумления обучающей выборки, что позволяет сделать АМ более устойчивой к различным шумам. Так как процесс обучения вычислительно сложен, а количество различных шумов, присутствующих в речи имеет большую разнообразность, поэтому необходимо используя акустические модели, полученные в каких-то фиксированных условиях (обычно с минимальным шумом), научиться распознавать речевой сигнал, полученный в других шумовых условиях [106]. Для этих целей и применяется аугментация.

Стоит отметить, что в работе не рассматривались методы для аугментации спектрограмм, такие как `SpecAugment` [107], т. к. они применяются в `end-to-end` акустических моделях, процесс обучения которых требует большого объема речевого материала и временных ресурсов. Используемая стратегия аугментации [108] состоит из следующих техник.

1. Использование масок микрофонных решеток. Для данного метода аугментации использован метод из работы [109], который заключается в записи микрофонных решёток с величиной отношения сигнал/шум 10-20 дБ, наложенных поверх оригинальной аудиозаписи. Также если длина шума ( $t_s$ ) меньше длины

оригинальной дорожки ( $t_{orig}$ ), то шум дублируется до достижения  $t_{orig}$ . Если  $t_s > t_{orig}$ , то шум обрезается до  $t_{orig}$ . Данный подход применяется на большинстве этапов аугментации, кроме этапов 6–9.

2. Наложение аудиозаписей шумов (шум автобуса и т.п.).
3. Использование аддитивного шума, присутствующего в обычных помещениях и реверберации (для небольших, средних и больших помещений).
4. Наложение музыки.
5. Наложение голоса другого диктора.
6. Использование вокодера для смены высоты тона речи (с коэффициентом,  $\beta$ , 0,9 и 1,1).

Для реализации вокодера для смены высоты тона речи применён алгоритм синхронного накладывающегося окна с равномерным шагом (Time-Domain Pitch Synchronous Overlap-Add, TD-PSOLA) [110], заключающийся в следующем. Допустим у нас есть сигнал  $y_{P_0}(t)$ , где  $P_0$  – его период:

$$y_{P_0}(t) = \sum_{k=0}^{\infty} h(t - kP_0), \quad (2.5)$$

где  $h()$  – функция фильтра.

Задача состоит в том, чтобы изменить сигнал  $y_{P_1}(t)$  посредством изменения его периода  $P_1$ . Для этого выполняются следующие шаги.

1. Извлекается импульсная характеристика сигнала из  $y_{P_0}(t)$ .
2. Находится местоположение каждого периода основного тона, путем нахождения пиков в сигнале, приближённых к  $P_0$ .

3. Вычисляется новый период  $P_1$ :

$$P_1 = F_s / F_{new}, \quad (2.6)$$

где  $F_s$  – текущая высота тона,  $F_{new}$  – новая высота тона:

$$F_{new} = \beta F_s. \quad (2.7)$$

4. Для каждого нового шага  $i = \{P_1, 2P_1, 3P_1, \dots\}$ : найти ближайший шаг в исходном сигнале; аппроксимировать импульсный отклик, применив окно Хеннинга длиной  $2P_0 + 1$ ; центрировать исходный шаг; перекрыть его и добавить оконный шаг, полученный при помощи окна Хеннинга, в новый буфер с центром в индексе  $i$ .

8) Применение вокодера с тональной маской другого диктора (отличие от пункта 6 состоит в том, что извлекается импульсная характеристика из аудиозаписи другого диктора).

9) Изменение скорости речи (с коэффициентом деформации,  $\beta$ , 0,9 и 1,1).

Для исследования эффективности использования стратегии аугментации проведен эксперимент над акустическими моделями, обученными с применением машинного обучения (скрытых марковских моделей и гауссовых смесей). Обучающая выборка состояла из 8\*14 часов аудиозаписей (8 часов оригинальных и 8\*13 часов аугментированных), тестовая выборка состояла из 2 часов. В качестве акустических признаков использовались мел-скептральные коэффициенты (размерностью 40) и коэффициенты перцептивного линейного предсказания (размерностью 3). ЯМ обучена на основе триграмм на текстовых данных, извлечённых из новостных лент и книг. Словарь сформирован из 500 тыс. наиболее встречаемых слов, извлечённых из языковой модели. Обучена монофонная (mono) и трифонная (tri1) акустические модели без применения аугментации (WER\_wo) и с её применением (WER\_with) (Таблица 2.6).

Таблица 2.6 – Результаты обучения АМ

Модель	WER_wo	WER_with
Mono	72,31	74,01
tri1	41,47	40,33

Можно сделать вывод, что применение аугментации с целью улучшения качества (повышения робастности акустической модели) распознавания речи является перспективным методом. Т. к. уже на этапе tri1 можно увидеть уменьшение WER на 1,14%. Стоит также отметить, что применение аугментации влечёт за собой значительное увеличение обучающей выборки, что влияет на количество времени и вычислительные ресурсы, необходимые для обучения акустической модели. Поэтому целесообразно использовать аугментацию при наличии небольшой обучающей базы, а также при использовании больших вычислительных ресурсов.

Перспективным подходом для аугментации речевых данных является использование в качестве обучающего материала, полученного при помощи синтеза речи [111–113], однако в рамках данной работы он не использовался.

Предварительная обработка речевых данных необходима для обучения АМ, направлена на повышение робастности и дикторонезависимости АМ. Для обучения ЯМ необходимо из имеющихся текстовых данных сгенерировать более естественные, максимально грамматически и семантически точные тексты. Эта задача решается с помощью нормализации текста.

#### 2.4 Разработка методов нормализации текста

В общем случае задача нормализации текстов представляет собой приведение слов и выражений (лексем) естественного языка к единой, общепринятой форме. В качестве выражения могут выступать устойчивые числовые, словесные или число-словесные выражения, например, условные записи дат, телефонных номеров, аббревиатуры. Под нормализацией текста в рамках данной работы понимается процесс трансформации исходного текста посредством удаления неконтекстных символов, т.е. без потери смысла исходного текста, а также преобразования символов, вносящих «шум» для понимания смысла (цифробуквенные комплексы, сокращения и т. п.). При нормализации текста для задачи автоматического распознавания речи целесообразно преобразовывать аббревиатуры и вставки на латинице, встречающиеся в русскоязычных текстах, а также проводить коррекцию ошибок.

Разработанный блок нормализации текста состоит из нескольких модулей (Рисунок 2.2).

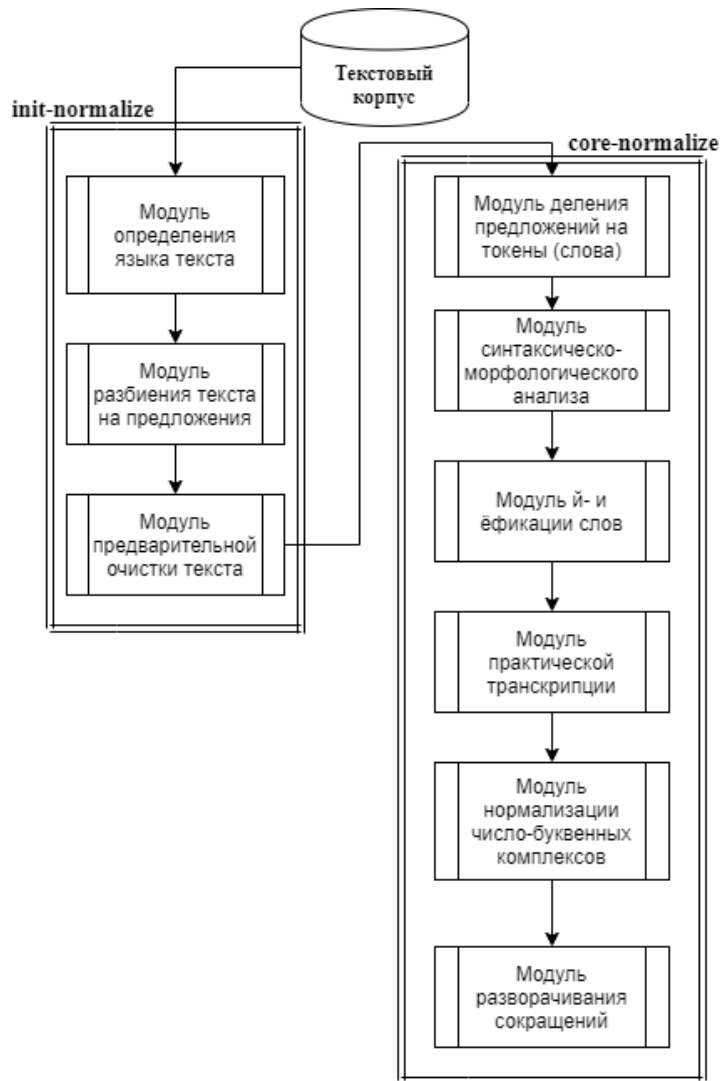


Рисунок 2.2 – Схема блока нормализации текста

#### 2.4.1 Общая схема нормализации текста

Предложенный алгоритм нормализации текста состоит из двух этапов: предварительная обработка (init-normalize) и основная обработка (core-normalize)

Основные шаги алгоритма init-normalize состоят в следующем.

1. Считывается массив текстов  $T = \{t_i\}$ , где  $i$  – общее количество текстов.
2. Для каждого  $t_i$  проводится проверка на принадлежность к русскому языку, для этого используется предобученная ANN-модель для классификации, описанная в выше. В качестве входного значения используется текст с количеством символов  $l_{min} = \max(l, 100)$ , где  $l$  – общее количество символов во входном тексте. В случае,



если язык текста отличен от русского – используется следующий  $t_i$ . Иначе – переходим на следующий шаг.

3. Проверяется текст с количеством символов  $l_{min}$  на наличие букв «ё» и «й». Если нет ни единого символа «ё» устанавливаем значение  $bool\_jo=False$ , нет ни единого «й» –  $bool\_j=False$ .

4.  $t_i$  разбивается по символу окончания строки. Формируется множество  $N=\{n_j\}$ , где  $j$  – количество полученных абзацев.

5.  $n_j$  разбивается на предложения при помощи модуля разделения текста на предложения, использующего модель, обученную на русских текстах (дампы Wikipedia), что позволяет точно определять границы предложения (например, игнорируя знаки «.» перед сокращениями, как «г.», «в.»). Формируется множество  $S=\{s_k\}$ , где  $k$  – общее количество предложений.

6. Для  $s_k$ :

a. производится очистка от тэгов, «неконтекстных символов», нестроковых значений, удаление текстовых последовательностей внутри «[]» и «()», удаление более 4 повторяющихся символов из слов, удаление предложений с верхним регистром и т. п., используя регулярные выражения для удаления или замены;

b. удаляются строки интернет-ссылок, электронных почт, слова, содержащий символ «#», а также слова, не несущие контекстной информации;

c. удаляются строки, в конце которых отсутствуют знаки, а также длина которых составляет менее 7 символов;

d. производится замена римских цифр на арабские.

7. Производится корректировка строк с неверными наращиваниями (200летний, 2005г., \$3), используя регулярные выражения.

8. Полученные строки записываются в файл формата csv. Дополнительно вносится информация о языке текста, наличия символа «ё» и «й».

После прохода `init-normalize` используем алгоритм `core-normalize`, который состоит в следующем.

1. Считываем информацию из файла с информацией. Считываем массив предложений. Если нет букв «ё» – подключаем модуль ёфикации, если нет букв «й» – подключаем модуль йфикации, описанные в пункте 2.4.5.

2. Предложение делим на массив слов, используя набор правил, реализованных при помощи регулярных выражений, исключая указатели деления («+», «-», «/»), а также символы пунктуации).

3. Извлекаем синтаксическо-морфологическую информацию.

4. Используем модуль расшифровки сокращений (пункт 2.4.4).

5. Используем регулярное выражение для определения является ли слово английским или русским, или содержит численную часть строки. Для английских слов производится практическая транскрипция на русский текст, используя алгоритм, описанный в следующей главе (пункт 3.4). Если содержит численную часть строки, то используется функция согласования чисел, полученная в результате применения алгоритма синтаксического анализа (пункт 2.4.3).

6. Если слово полностью состоит из букв верхнего регистра – проводится проверка на количество слогов, если содержит один слог – каждая буква нормализуется в соответствии со словарём («АНБ» - «а эн бэ»), иначе слово приводится к нижнему регистру.

7. Корректируются слова, содержащие знаки препинания в конце, а также состоящие из одних знаков препинаний. Знаки препинания удаляются.

8. Слова записываются через пробел, отделяя знаком конца строки, когда достигнут конец предложения. Полученный текст записывается в текстовый файл.

Первым этапом автоматической нормализации текста является определение его языка.

## 2.4.2 Разработка нейросетевой модели определения языка текста

Для определения языка текста предложения разработана нейросетевая модель на основе свёрточных слоёв [114].

Для обучения нейросети необходимо получить векторное представление символа. Алгоритм получения векторного представления символа состоял в следующем:

- создать общий словарь встречаемых символов;
- обучить char2vec – нейросетевую модель для представления символа в виде вектора, размерностью 100, используя обучающий набор;
- используя обученную char2vec модель, векторизовать набор обучающих данных;
- используя набор векторизованных данных с соответствующим классом (языком), обучить ANN для определения языка текста (Рисунок 2.3).

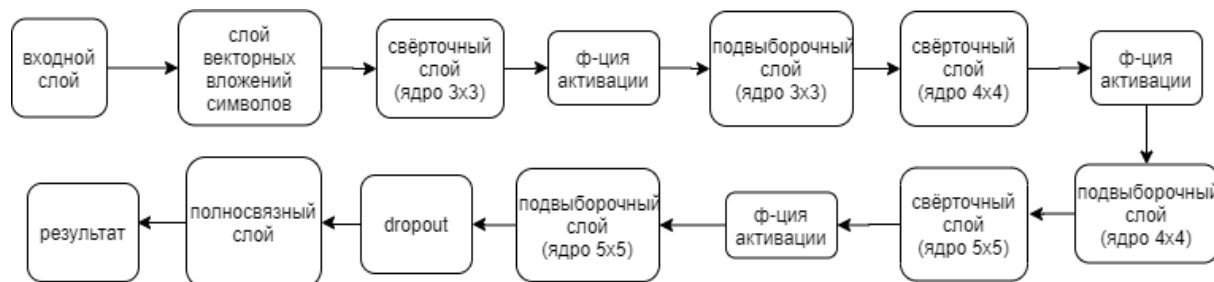


Рисунок 2.3 – Архитектура ANN для определения языка текста

В качестве обучающего материала использованы текстовые расшифровки для видеовыступлений из проекта Ted-Talks [115] для языков, содержащих символы кириллицы (белорусский (bel), болгарский (bol), киргизский, казахский, русский, сербский, таджикский, македонский, украинский, азербайджанский), а также для наиболее распространённых языков, содержащих символы латиницы (английский, немецкий, испанский, французский).

Для тестирования использовалось по 100 случайно выбранных предложений для каждого языка. Для обучения использовалось от 2 000 до 10 000 предложений для каждого языка.

При обучении модели классификации использованы следующие гиперпараметры:

- размер батча: 128 (данный датасет объёмный, поэтому данные делим на пакеты или батчи, что позволяет оптимизировать процесс обучения модели);
- размер векторных представлений для символов: 100;
- коэффициент скорости обучения: 0.01;
- количество эпох обучения: 50;
- количество скрытых слоёв: 100;
- размер скрытых слоёв: 100;
- метод оптимизации: Adam;
- функция активации скрытых слоёв: Rectified Linear Unit (ReLU) [116].

В качестве метрик использовались стандартные метрики потерь и точности классификации (Рисунок 2.4), а также F-мера (F1-score) (Рисунок 2.5).

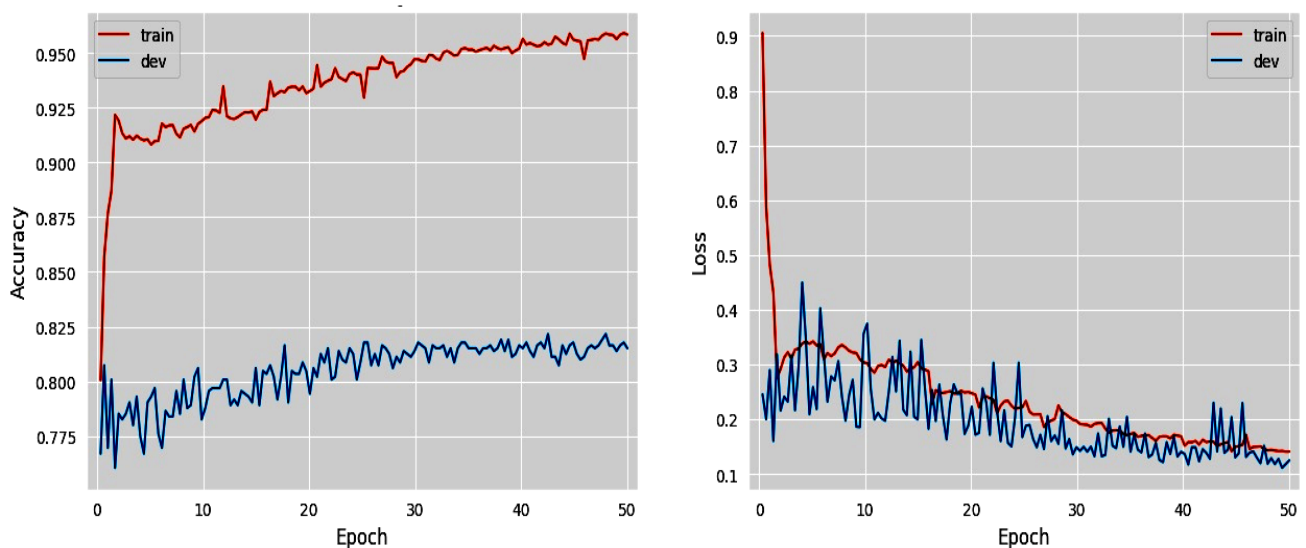


Рисунок 2.4 – Зависимость точности (ассигасу) и потерь (loss) для обучающей (train) и для тестовой (dev) выборок от количества эпох при обучении модели классификации языка по тексту

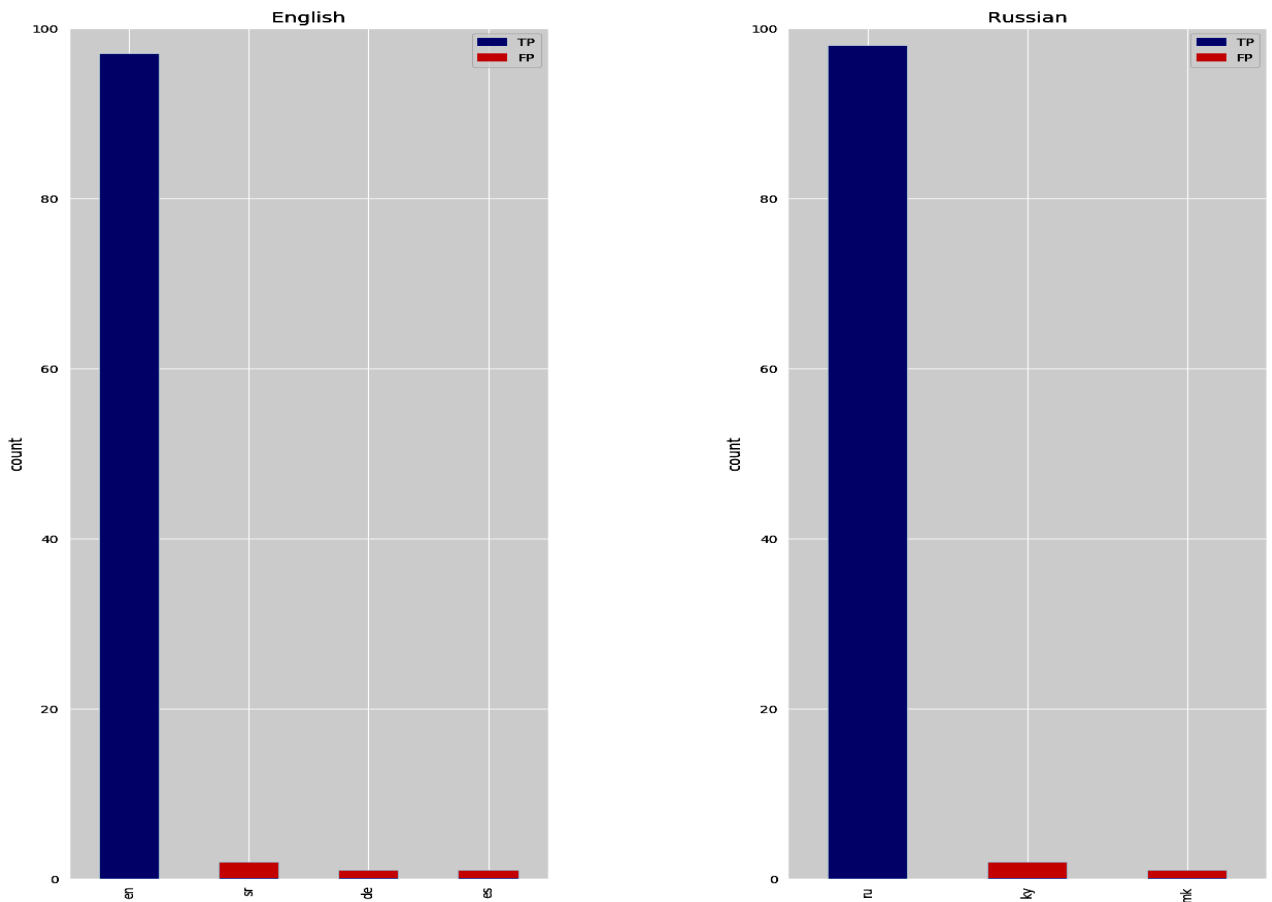


Рисунок 2.5 – Диаграммы F-меры для классификации английского (слева) и русского (справа) языков

Проведенные исследования показали, что точность модели определения языка текста составила на тестовой выборке 82.5%, на обучающей – более 95%.

Для корректной расшифровки цифро-буквенных комплексов необходимо провести согласование чисел методами синтаксического анализа.

### 2.4.3 Разработка системы синтаксического анализа

Для задачи нормализации текстов разработан синтаксический анализатор (парсер) с применением глубоких нейронных сетей.

В качестве основных метрик для парсера используются оценка немаркированной принадлежности (Unlabelled Attachment Score, UAS) и оценка маркированной принадлежности (Labelled Attachment Score, LAS). UAS это процент слов, чья синтаксическая позиция правильно определена; LAS это процент

слов, чья синтаксическая позиция правильно определена с помощью соответствующей метки зависимостей (subject, object) [117, 118]. UAS и LAS используют как на уровне слов ( $UAS_w$ ), так и на уровне предложений ( $UAS_S$ ).

При анализе доступных решений были найдены готовые модели для русского языка (по состоянию на 2018 г.): Parsey Universal или SyntaxNet ( $UAS_S$ : 81,75%;  $LAS_S$ : 77,71%) [119]; UDPipe ( $UAS_S$ : 89,8%;  $LAS_S$ : 87.2%) [120]. Но они обладают недостаточной точностью, используют медленный алгоритм парсера, кроме того, для этих моделей нельзя применить ускорение на уровне графических микропроцессоров.

Для применения ANN-метода необходима большая база обучающих примеров. При обучении парсера в настоящее время используют данные стандарта Universal Dependencies treebank (UDT) [121]. В Сети были найдены следующие корпуса для обучения парсера русского языка формата UDT: Russian GSD, Parallel UDT, SynTagRus UDT, Taiga UDT [122].

ANN-модель, используемая для разработки данного синтаксического анализатора основана на свёрточных слоях, т. к. синтаксический анализатор будет использоваться для задач нормализации текста, а данная ANN-архитектура является оптимальной для этой цели [123]. В связи с этим использована ANN, состоящая из 3-х свёрточных ANN, с использованием слоя внимания (attention) для формирования контекстных векторов – 3-CNN-attention.

Механизмы внимания могут быть использованы для повышения производительности нейронных сетей. В модели с вниманием (Рисунок 2.6) вместо построения одного контекстного вектора из последнего скрытого состояния сети создаётся контекстный вектор для каждого входного слова. Т.е., если в исходном документе  $N$  уникальных слов, то должно быть создано  $N$  контекстных векторов, а не один. Преимущество применения данного подхода состоит в том, что закодированная информация хорошо декодируется моделью.

Формула контекстного вектора для модели внимания имеет вид:

$$c_t = \sum_{j=1}^{T_x} a_{tj} h_j,$$

где  $a_{ij}$  – веса для каждого скрытого состояния  $h_j$  (оценка внимания):

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{T_x}^{k=1} \exp(e_{ik})},$$

где  $e_{ij}$  – модель вложения документа:

$$e_{ij} = a(s_{t-1}, h_j),$$

где  $s_t$  – скрытое состояние сети:

$$s_t = f(s_{t-1}, y_{t-1}, c_t).$$

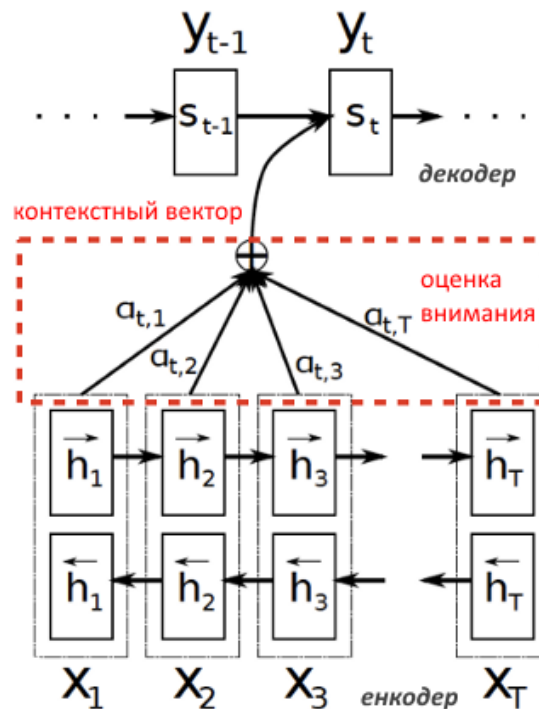


Рисунок 2.6 – Архитектура модели с вниманием

Архитектура используемой для парсера сети изображена на рисунке 2.7. Каждый свёрточный слой содержит 200 слоёв в глубину, а данные, подающиеся на вход ANN, представлены вектором размерностью в 64 единиц (200\*64 признаков).

Главная идея данной ANN заключается в формировании контекстного вектора для каждого слова в предложении. 3-CNN выполняет как роль парсера, так и морфоанализатора (определяет часть речи и лемму). Одновременное использование как синтаксической, так и морфологической информации в качестве входных данных позволяет добиться улучшения в точности. В качестве входной

информации используются слова с соответствующей морфологической информацией (слово: мам; лемма: мама; часть речи: NOUN), а в качестве соответствующих меток (классов) – синтаксическая позиция и метка (позиция: 1; метка: nsubj).

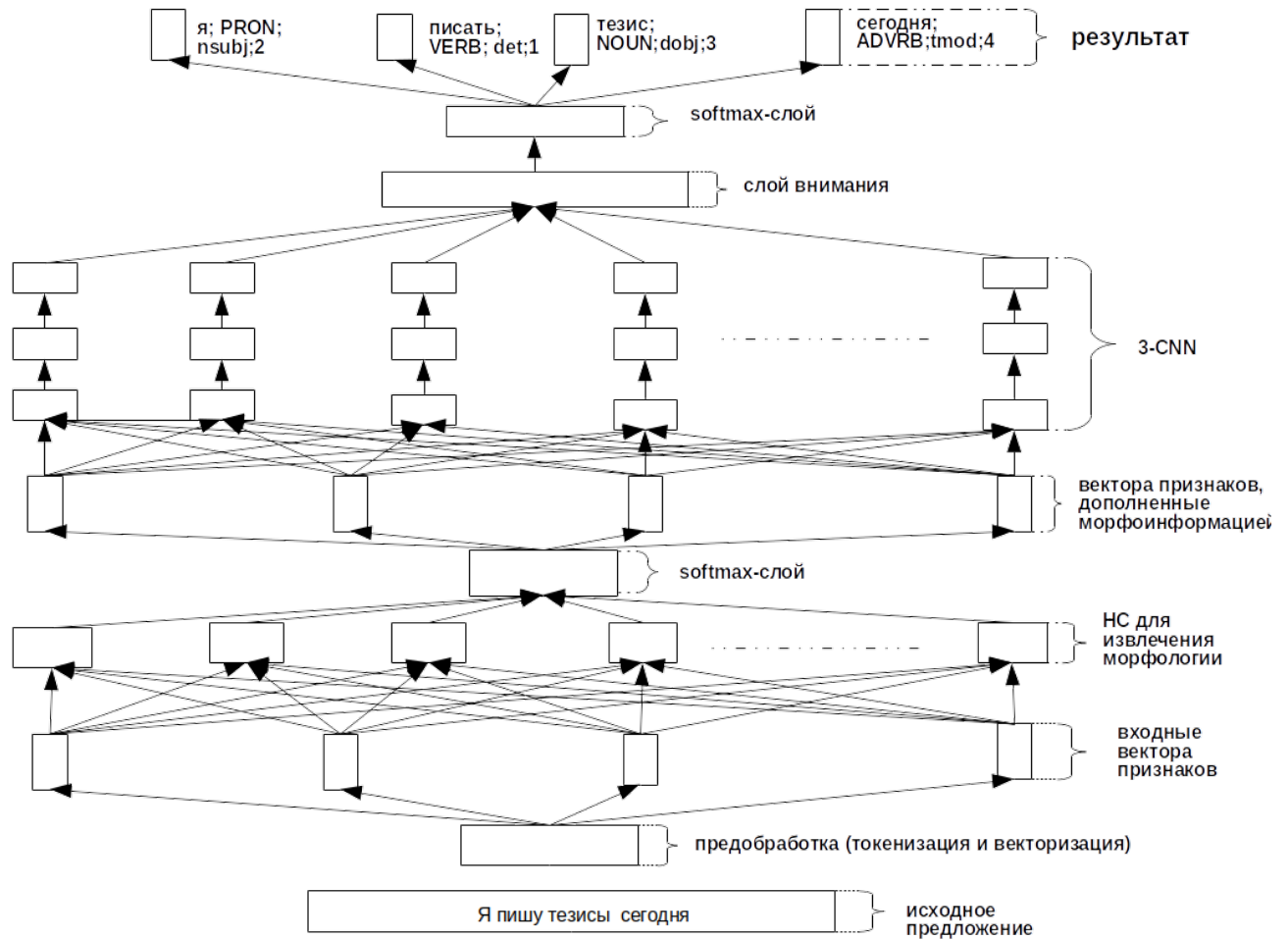


Рисунок 2.7 – Нейросетевая архитектура системы синтаксического анализа

На рисунке 2.8 отображены результаты обучения ANN-модели синтаксическо-морфологического анализатора по критериям LAS и UAS (на уровне предложений).



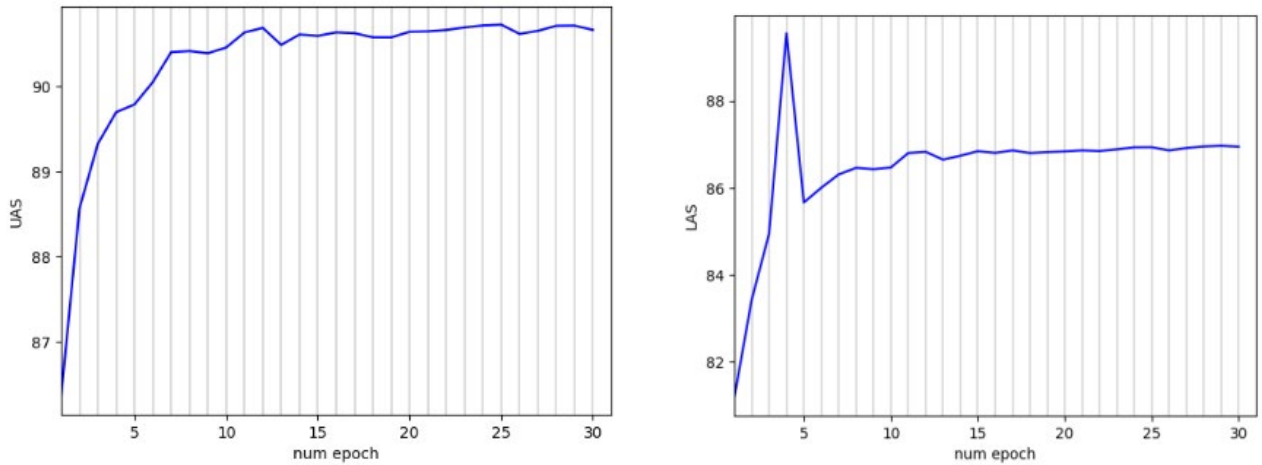


Рисунок 2.8 – Зависимость величины критериев UAS (слева) и LAS (справа) от количества эпох обучения системы синтаксического анализа

Предложенная модель синтаксического анализатора дает качество по критерию UAS – 94,3%, по критерию LAS – 90,2%.

#### 2.4.4 Разработка метода обработки цифробуквенных комплексов и аббревиатур

Модуль согласования чисел предназначен для автоматической нормализации цифробуквенных комплексов и аббревиатур (трансформации в буквенные аналоги). Алгоритм модуля согласования чисел состоит в следующем.

1. В предложении ищется слово, состоящее из числа или имеющее численную часть.

2. В случае, если такое вхождение слово найдено – определяется патерн (шаблон) слова:

- патерн наращивания («10-й», «10й», «10-летний», «10летний»);
- патерн аббревиатура и цифра («А1Б», «ТУ-104», «104-ТУ»);
- патерн времени и счёта («12:30», «12-30», «12:30:30»);
- патерн дат («01.05.2018», «01/05/2018», «2018-05-01»);
- патерн телефонных номеров;

- патерн дробей и адресов («3/4»);
- патерн чисел с плавающей точкой («0.5», «0,5»);
- патерн простого числа, который дополнительно подразделяется на «количественное числительное» + «сокращение» («5 кг»); «порядковое числительное» + «сокращение» («5-ый кг»); «количественное числительное» («5»).

Стоит отметить, что для определения типа шаблона простого числа использовались словари с наиболее известными сочетаниями пар. Данную задачу можно решить с использованием sequence-to-sequence (seq2seq) моделей [124], но этот подход усложнит вычислительный процесс. Подход с использованием словарей позволяет определить шаблон (патерн) для большинства известных случаев.

### 3. В соответствии с паттерном происходит нормализация текста:

- для патерна наращивания используется таблица окончаний, в соответствии с которой склоняется слово; в случае «10-летний» – численная часть приводится в дательный падеж «десятилетний»;
- для патерна аббревиатур цифровая и буквенная части разделяются «А1Б» – «А», «1», «Б». Затем буквы соотносятся со словарём («А1Б» – «а один бэ»);
- для патерна времени и счёта используется ANN для синтаксического анализа, берутся 2 синтаксически связанных слова с вхождением патерна (ведётся поиск предлога и самостоятельных частей речи). В случае предлога используем словарь, для остальных – извлекаем падеж из самостоятельной речи, склоняем оба числа. Стоит отметить, что существуют так называемые «двойные падежи», которые отличаются от стандартных шаблонов формирования падежей, эта проблема частично решается составлением словаря или набора правил для проблемных сокращений (как правило, в качестве определителя падежа используются предлоги).
  - нормализация патерна дат проводится аналогично патерну времени и счёта;
  - при нормализации патерна телефонных номеров учитывается два типа этого патерна:

а) патерн для телефонных номеров, записанных в виде «361-72-72». Для этого патерна разделяются числа, используя разделитель «-». Полученные числительные приводятся в текстовый вид как начальные формы количественных числительных;

б) патерн телефонных номеров, записанных в виде «+380714648734». Последние 9 цифр разделяются как «714», «64», «87», «34». Все остальные символы разделяются посимвольно.

– нормализация патерна дробей и адресов производится аналогично патерну времени и счёта, но изменяется лишь второе число; для адресов (просматривается начинается ли предыдущее слово с большой буквы) – преобразуем каждое число в количественное числительное;

– для обработки патерна чисел с плавающей точкой делим числа по разделителю и смотрим на длину второго числа (удаляя нули, если они находятся перед цифрами: «0.005») в соответствии с длиной используем словарь и добавляем слово в конец, дополнительно используем подход из патерна аббревиатуры и счёта для добавленного слова в конец;

– для патерна простого числа ищем синтаксически связанные слова, если из списка сокращений слово находится на расстоянии 0...2 – по сокращению определяем вид числительного; далее ищем соседние синтаксические слова (местоимение, предлог, имя прилагательное). Число переводим в строковую форму, а сокращение в полную форму, изменяя его, используя морфологическую информацию соседнего синтаксического слова («в 5 в.» – «в пятом веке»).

В русскоязычных текстах, как правило, буква «ё» заменяется на «е», благодаря чему упрощается и ускоряется распознавание слов в ходе морфологического анализа для большинства текстов. Для построения качественной языковой модели необходимо провести обратную операцию – заменить «е» на «ё» в соответствии с правилами русского языка там, где «ё» должна употребляться.

Еще одной проблемой при обработке русскоязычного текста является наличие опечаток, где «й» заменяется на «и». В связи с этим для проведения

нормализации необходимо разработать и реализовать алгоритмы «ёфикации» и «йфикации» слов.

#### 2.4.5 Разработка методов «ёфикации» и «йфикации» слов

Методы «ё-» и «йфикации» предназначены для автоматического исправления ошибок в словах, а именно неверного использования символов «е» (или «и») и «ё» (или «й»). Эти модули основаны на 2-х типах словарей, сформированных автором:

- 1) словарь, содержащий парадигмы слов с символами «ё» («й»).
- 2) словарь, содержащий парадигмы слов, в которых в одной и той же позиции может использоваться как символ «ё» («й»), так и символ «е» («и»): «небо» – «нёбо»; «мои» – «мой».

Принцип работы *методов «ёфикации» и «йфикации»* следующий.

1. Инициализируется слово.
2. Производится проверка на наличие в слове символов «ё» и «е» (или «й» и «и»). Если данные символы присутствуют, то переходим к следующему шагу. Иначе возвращаем исходное слово.
3. В слове символы «ё» (или «й») заменяются на символы «е» (или «и»). Производится поиск слова в исходном словаре, который содержит парадигмы слов с символами «ё» (или «й»), и извлекается его найденное значение в случае, если слово есть в данном словаре («елка»: «ёлка» или «иод»: «йод»; данный словарь состоит из пар слов). Иначе переходим к следующему шагу.
4. Слово ищется в словаре, который содержит парадигмы слов, в которых в одной и той же позиции может использоваться как символ «ё» (или «й»), так и символ «е» (или «и»), если такое вхождение слова есть, то извлекается его найденное значение. Иначе переходим к следующему шагу.
5. Если слова нет ни в одном из словарей, то по умолчанию символы «ё» (или «й») заменяются на «е» (или «и»), т. е. используется «ефицированное» (или «ифицированное») значение слова, т. к. это слово с большой вероятностью относится к словам, содержащих символ «е» (или «и»). Существует вероятность

того, что это слово внесловарное («Тёёлё» или «днровский»), но данную проблему практически невозможно решить ни применением шаблонов, ни применением машинного обучения (т. к. большинство текстов «ефицировано» или «ифицировано»). Данная проблема решается лишь за счёт расширения существующих словарей, поэтому исходное слово дополнительно выводится в отдельный текстовый файл для дальнейшего рассмотрения.

Для метода «йфикации» дополнительно создан алгоритм с учётом контекста в предложении.

Работа *метода «йфикации» с учётом контекста в предложении* состоит в следующем.

1. Взять исходное слово, а также предложение, где встречается данное слово.
2. Проверить слово на наличие символов «и» или «й». Если данные символы есть – переходим к следующему шагу. Иначе возвращаем исходное слово.
3. Использовать предобученную ANN-модель синтаксического анализа, описание которой приводится в пункте 2.4.3, для данного предложения, т. е. построить синтаксическое дерево связей.
4. Используя синтаксическое дерево связей, взять 4 синтаксически близких слова с искомым (2 слева и 2 справа).
5. Т. к. ANN-модель синтаксического анализа позволяет извлекать информацию только о части речи, то дополнительно используется ANN-модель морфологического анализа, RNNMorph [125], позволяющая извлечь расширенную морфологическую информацию для каждого слова в исходном предложении.
6. Найти ближайшее слово, являющееся прилагательным, местоимением, существительным, глаголом (т. е. содержащее информацию о числе). Если такое слово найдено – переходим к следующему шагу, иначе используем алгоритм «йфикации» без учёта контекста.
7. Если ближайшее слово имеет множественное число, то и «йфицированное» слово имеет множественное число (по аналогии определяется единственное число).

8. Разбиваем слово на 3 части, используя алгоритм стемматизации (описанный в пункте 3.3.2): приставку, корень и суффикс с окончанием.

9. Применяем запрограммированные правила «йфикации» к корню; производится проверка на наличие символов в части с окончанием, если присутствуют символы, то производится проверка по морфологической информации, а именно числу, «йфицированного» слова; если число множественное – в данном слове будет использоваться символ «и»; единственное – символ «й». Если часть с окончанием пуста, то далее используется алгоритм «йфикации» без учёта контекста.

Стоит отметить неточности в правилах орфографии относительно написания «й», а именно то, что после согласных буква «й» не пишется» [126]. Это не соответствует действительности, примером чему могут служить слова «безйотовый», «безйодовый». Данная погрешность устранена в [127].

## 2.6 Выводы к главе 2

1. Для обучения АМ из различных открытых источников подготовлен речевой корпус общей длительностью 29 часов 40 минут, содержащий записи 231 мужского и 91 женского голосов. Для обучения языковых моделей подготовлена база текстов объемом 15,2 Гб. Предложена модификация алгоритма Смита-Уотермана для проверки соответствия текстовых расшифровок и аудио, что позволило повысить точность формирования текстовых расшифровок в среднем на 10,5%. Предложена техника аугментации речевых данных для повышения робастности АМ, что позволило уменьшить WER на 1,14% для трифонной АМ и на 1,7% для монофонной. Для создания качественной языковой и акустической моделей разработаны алгоритмы нормализации текстов, нормализующие текстовые расшифровки к аудио и текстовый корпус. Эти алгоритмы позволяют определять язык текста отдельного предложения с помощью, разработанной нейросетевой модели на основе свёрточных слоёв, которая дает точность на тестовой выборке 82,5%;

– генерировать синтаксическую структуру текста для корректной расшифровки цифро-буквенных комплексов с помощью морфосинтаксического парсера, оценка точности которого составила по критерию UAS – 94,3%, по критерию LAS – 90,2%;

– трансформировать цифробуквенные комплексы и аббревиатуры в буквенные аналоги с помощью разработанного алгоритма, использующего морфосинтаксический парсер;

– автоматически исправлять неверное использование символов «е»/«ё» и «и»/«й» с помощью предложенных декларативных методов «йфикации» и «ёфикации» слов, опирающихся на созданные словари, которые содержат парадигмы слов с символами «ё»/«й».

## ГЛАВА 3

### РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ТРАНСКРИПЦИЙ CYR2TRANS

Как указано в первой главе, моделирование фонем и генерация транскрипции слов являются одними из важных шагов при разработке ASR-системы. Имеющиеся на сегодняшний день системы генерации транскрипций русского языка не учитывают всех особенностей произношения, в частности положения гласного относительно ударного слога, произношения иностранных слов, а также слов с апострофами. В связи с этим возникла необходимость разработки ряда дополнительных правил, которые будут реализованы в фонетическом модуле ASR-системы, именуемом далее PhonemCyr, учитывающих особенности фонетики русского языка.

#### 3.1 Особенности фонетики русского языка

В отличие от остальных систем формирования автоматической транскрипции в PhonemCyr, предусмотрен ряд дополнений, учитывающих особенности фонетики русского языка. Эти дополнения сформированы на основе информации, полученной из работ [128–132].

##### 3.1.1 Особенности произношения гласных

Приведем несколько особенностей произношения гласной в зависимости от ее положения относительно ударного слога.

***Предударные гласные (слоги).*** Ударения для гласных осуществляются следующим образом: алгоритмом определяется позиция ударения; все гласные, стоящие перед ударной гласной являются предударными гласными (слогами) и обозначаются как «А\_» (самая ближняя предударная обозначается как «А\_», в свою очередь остальные предударные обозначаются как «А\_\_»). Стоит отметить, что во



втором и третьем предударных слогах гласные подвергаются более значительной редукции, чем в первом слоге. В формате IPA предударный звук пишется как [ʌ]. Данный подход позволяет избавиться от неоднозначности для предударных слогов в следующих случаях:

- после твердых согласных на месте букв *a*, *o*, *e* произносится звук средний между [ы] и [а]: «выдал», «выпал».
- в не полностью освоенных или нарочито произнесенных иностранных словах: «поэт», «бомонд».
- в словах с сочетаниями [oá], [áo], [ио], [ио]: «боá», «oáзис», «хаос», «какао», «период», «радио».
- в безударных слогах при произношении иноязычных слов, пишущихся с э: «экран», «эволюция», «фаэтон».
- перед мягкими согласными на месте букв *e*, *я* при ударных [o], [a], [e], [и] произносится звук средний между [и] и [е]: «несла», «река», «мясной».
- на месте буквы *e* после шипящих [ж], [ш] и после [ц] произносится звук средний между [ы] и [э]: «желток», «жесток», «шесток», «шестой», «цена».
- после мягких согласных на месте букв *o*, *a*, *я*, *e* произносится звук, средний между [и] и [е]: «часы», «часок», «лесок».
- на месте букв *a*, *o* в начале слова произносится звук средний между [o] и [a]: агент, осока;
- буквы *я*, *e* в начале слова обозначают два звука [ja], [je], из которых первый является мягким согласным [j], поэтому на месте этих букв произносятся звук средний между [и] и [е]: «ярмо», «еда»;
- после твердых согласных на месте букв *a*, *o*, *e*, находящихся не в конце слова, произносится звук средний между [ы] и [а]: «видел», «пальцем».

**Зударные гласные (слоги).** Гласные, стоящие после ударной гласной, являются заударными и обозначаются как «А\*», в свою очередь самая ближняя обозначается как «А\*», а самая дальняя как «А\*\*». В формате IPA заударный звук обозначается как [ʔ]. Произношение гласных в заударных слогах, в большинстве случаев, аналогично произношению гласных во всех предударных слогах, кроме

первого. Однако, произношение заударных гласных отличается рядом частных особенностей, касающихся произношения гласных в составе различных морфем:

- после твёрдых согласных на месте букв *а, о* произносится редуцированный звук в окончаниях именительного падежа (им. п.) единственного числа (ед. ч.) существительных женского и среднего рода; родительного падежа (род. п.) единственного числа существительных мужского и среднего рода; именительного падежа множественного числа (мн. ч.) существительных мужского и среднего рода: «сталино», «ведомства».

- после мягких согласных на месте букв *а, я* произносится редуцированный звук в окончаниях им. п. ед. ч. существительных женского и среднего рода; род. п. ед. ч. существительных мужского и среднего рода; им. п. мн. ч. существительных мужского и среднего рода; в суффиксе деепричастия несовершенного вида; им. п. ед. ч. прилагательных женского рода: «воля», «туча», «роща». А также в конечном слоге, если он не представляет собой окончание, перед мягким согласным: «память», «площадь», «поняли».

- после мягких согласных на месте буквы *е* произносится звук [ь] в окончаниях: 1) дательного, творительного, предложного падежей единственного числа существительных женского рода: «туче»; 2) предложного падежа единственного числа существительных мужского и среднего рода: «олене»; 3) родительного падежа множественного числа существительных с окончанием *ей*: «олений»; 4) им. п. мн. ч. существительных на *ан(е)*: «римляне»; 5) в неконечном и конечном слогах, если они не составляют окончания.

**Побочное ударение.** Многие сложносоставные слова (имеющие более одного корня) кроме основного ударения могут иметь побочное (или побочные). При наличии двух ударений в слове побочным, как правило, объявляется ударение, находящееся ближе к началу слова, а основным ударением объявляется ударение, находящиеся ближе к концу слова. Побочное ударение характеризует свободный стиль речи («общежитие», «девятьсот»). Помимо сложносоставных слов, побочное ударение могут иметь и сложносокращённые слова («Донгормаш»). Также побочное ударение могут иметь приставки в словах («чрезмерный»). С побочным

ударением обычно произносятся слова иноязычного происхождения («постскриптум»). Если в сложносоставном слове три основы, то оно может иметь три ударения – 2 побочных и 1 основное («авиаметеослужба»).

### 3.1.2 Особенности произношения иноязычных слов

В силу длительных экономических, политических, культурных, военных и иных связей русского народа с другими в его язык проникло довольно значительное количество иноязычных слов, которые имеют различную степень ассимиляции и неограниченную или ограниченную сферу употребления. В русской лексикологической традиции выделяются: слова, давно усвоенные и используемые наравне с русскими («стул», «лампа», «школа» и т.д.); слова, не всем понятные, но необходимые, так как они обозначают понятия науки, техники, культуры и т.п. («фонема», «морфема», «дагностицизм» и т.п.); слова, которые могут быть заменены исконно русскими без всякого ущерба для смысла и выразительности высказывания («эпатировать», «эпатаж», «апологет», «акцентировать», «визуальный» и т.п.). Сейчас значительная часть таких слов по своему произношению ничем не отличается от слов исконно русских. Но некоторые из них – слова из разных областей техники, науки, культуры, политики и в особенности иноязычные собственные имена, – выделяются среди других слов русского литературного языка своим произношением, нарушая правила.

Опишем особенности иноязычных слов на основе информации из работ [133–142]. На фонетическом уровне таковыми являются:

- *а, э, ит, ип* – в начале слова;
- звук и буква *ф*, сочетание *дж, уа, уэ, ау, оу*;
- сочитания *бю, пю, дю, тю, сю, зю, ню, мю, рю, лю*;
- сочетания *ей, ой, ай* + согласный звук;
- сочетание гласных *а, о* + носовой согласный *н* или *м*;
- слоговой сингармонизм;
- удвоенные согласные;

- произнесение *о* в безударных слогах;
- фиксированное ударение на последнем слоге.

К словообразовательным особенностям относятся:

- приставки: *дис, а, квази, пан, интер, де, ре, ир* и др.
- суффиксы: *аж, ёр, ист* и др;
- «инговое» окончание;
- корни: *зоо, авиа, био, агро* и др;

К морфологическим особенностям относится несклоняемость слова.

При произношении слов иноязычного происхождения следует учитывать следующие особенности.

1. Звук [о] в безударных слогах. В первом и во втором предударном слоге, в абсолютном начале слова, а также в заударных слогах в абсолютном конце слова после согласных или гласных на месте буквы *о* произносится гласный [о] без характерной для русских слов редукции: «**бо**а», «**до**сье», «**бо**рдо», «**ко**нсо**ме**», «**мо**дерато». Безударный гласный нередко сохраняется в иноязычных собственных именах: «**Бо**длер», «**Зо**ля», «**До**лорес», «**Ро**ден».

2. В некоторых малоупотребительных именах собственных в предударных слогах сочетания букв *ао, оа, оо, оу* и *уо* произносятся так, как пишутся, т.е. без редукции: «**Ао**гасима», «**Оа**хака», «**Мо**орéа», «**Ло**урива́л», «**лу**ораветла́ны», «**ко**коро».

3. В иноязычных не русифицировавшихся словах согласные перед *е* не смягчаются, как в исконно русских. Это относится, прежде всего, к зубным согласным (кроме *л*) [т], [д], [с], [з], [н], [р]. Но для ряда широко распространённых слов иноязычного происхождения согласные перед *е* смягчаются: **про**фессор, **а**грессор, **бе**рет и т. д.

4. Твёрдый [т] произносится в таких словах, как «атеизм», «ателье», «стенд», «эстетика». Сохраняется твердый [т] и в иноязычной приставке «интер» («интервью»), а также в ряде географических названий и других собственных именах: «Амстердам», «Данте».

5. Звук [д] не смягчается в словах «кодекс», «модель», «модерн» и др., а также в таких географических названиях как «Дели», «Родезия» и фамилиях «Декарт», «Мендельсон».

6. Звуки [з] и [с] произносятся твердо лишь в немногих словах: «сентенция», «Морзе». Также твердые [з] и [с] встречаются в именах и фамилиях, таких, как «Жозеф», «Сенека».

7. Звук [н] также остается твердым в именах и фамилиях («Рене», «Нельсон»). Большинство слов произносится с твердым [н], но появляются случаи, когда [н] перед е смягчается: «неолит», «неологизм».

8. В заимствованных словах, начинающихся с приставки «де», перед гласными «дез», а также в первой части сложных слов, начинающихся с «нео», при общей тенденции к смягчению наблюдаются колебания в произношении мягкого и твердого [д] и [н], например: «девальвация», «деидеологизация», «дезинформация», «дезодорант», «неоглобализм», «неоколониализм», «неофашизм».

9. Твердое произнесение согласных перед е рекомендуется в иноязычных собственных именах: «Белла», «Жорес», «Кармен», «Мери», «Ионеско», и др.

10. В заимствованных словах с двумя (и более) е нередко один из согласных произносится мягко, а другой сохраняет твердость перед е: «бретелька», «генезис», «реле», «генетика», «кафетерий», «пенсне», «реноме», «секретер», «этногенез».

11. В сравнительно немногих иноязычных по происхождению словах наблюдаются колебания в произношении согласного перед е, например: при нормативном произношении твердого согласного перед е в словах «бизнесмен», «аннексия» допустимо произношение с мягким согласным; в словах «декан», «претензия» нормой является мягкое произношение, но допускается и твердое [дэ] и [тэ]; в слове «сессия» варианты твердого и мягкого произношения равноправны. Ненормативным является смягчение согласных перед е в профессиональной речи представителей технической интеллигенции в словах «лазер», «компьютер», а

также в просторечном произношении слов «бизнес», «бутерброд», «интенсивный», «интервал».

12. Стилистические колебания в произношении твердого и мягкого согласного перед *e* наблюдаются также в некоторых иноязычных именах собственных: «Берта», «Декамерон», «Рейган», «Крамер», «Грегори».

13. В некоторых заимствованных словах в литературном произношении после гласных и в начале слова звучит достаточно отчетливо безударное [э]: «дуэлянт», «муэдзин», «поэтический», «эгида», «энергия», «энциклопедия», «эпиграф».

14. Твердый [ш] произносится в словах, как правило, заимствованных из французского языка «парашют», «брошюра». В слове «жюри» произносится мягкий шипящий [ж']. Твёрдый [ж] произносятся в именах «Жюльен», «Жюль».

15. Очень часто заимствованные слова искажаются: осуществляется неправомерное выпадение, вставка или замена звуков. Так, признается верным произношение: *инициатива* (не *инциатива*), *инициалы* (не *инциалы*), *дерматин* (не *дермантин*), *констатировать* (не *константировать*), *компрометировать* (не *компроментировать*), *мармелад* (не *мармалад*), *инцидент* (не *инциндент*) и др.

16. В словах иноязычного происхождения, не вошедших в широкое употребление, наблюдаются специфические особенности произношения. Например, в словах из разных областей науки, техники, политики, культуры, а также именах собственных возможно отсутствие качественной редукции безударных гласных.

17. Некоторые слова допускают два варианта произношения согласного. Однако однозначных правил произношения твердых-мягких согласных перед *e* привести нельзя, каждый случай следует проверять по словарю и запоминать.

### 3.1.3 Произношение слов с апострофом

Апостроф (') – письменный/печатный знак в виде запятой на верхней линии строки. Апостроф является так называемым небуквенным орфографическим

знаком. При этом в ряде слов апостроф логически делит слово на подслова («д'Ареццо» = «д» + «Ареццо»), также может входить в фонетическую основу слова (Word'a = «ворда»). Применяется апостроф при написании определённых слов, а именно:

- апостроф пишется во французских фамилиях, именах, прозвищах с начальной частицей *d'*: «д'Артаньян», «д'Онуа», «д'Арк»;

- апостроф применяется для записи британских фамилий с начальным *O'* (что означает «внук»): *O'Кейси*, *O'Коннор*;

- в словах итальянского происхождения («д'Ареццо»);

- в названиях различных географических объектов: «Кот д'Ивуар», «Кот-д'Ор», «Кот-д'Армор», «Л'Умо», «Пон-л'Эвек», «Л'Иль-сюр-ла-Сорг», «Моррод'Оро», «Л'Алькерия-д'Аснар» «Ка'Дарио», «Ка'д'Оро»;

- апостроф употребляется в русском языке для отделения основы собственных имён, записанных латиницей, от последующего окончания (обычно падежного), которое, по общим правилам, всегда записывается кириллицей. Примеры: «Word'a», «Microsoft'y», «Firefox'ом».

- в словах, относящихся к художественной литературе 19-го и начала 20-го веков: «Да'с»;

- в некоторых сокращённых словах: «O'кей».

### 3.1.4 Правила образования сложносоставных слов

Большую трудность вызывают сложные слова, состоящие из двух и более основ. Эта трудность вызвана наличием более одного ударения в слове, в связи с чем стандартные правила транскрипции не применимы. Явления, когда в образовании сложного слова используется более двух корней, достаточно редки («веломотодром»). Следует отличать сложные слова от простых. Так, в слове «электрификация» всего один корень «электри-», а все, стоящее за ним – это суффикс и окончание. Сложные слова образуются следующим образом.

1. Сложение полных основ. Образовываться эти слова могут при помощи сочинительных и подчинительных связей:

- сложные слова (существительные и прилагательные) со значением сторон света («западно-европейский», «юго-восточный»);

- слова, между частями которых легко можно поставить союз "и" («мясо» и «молоко» – «мясо-молочный»);

- сложные слова, образованные с помощью подчинительной связи - «лесоперерабатывающий» («перерабатывать лес»);

- слова, передающие оттенки цветов: «малиново-золотой», «серо-буро-коричневый», «светло-зеленый», «пурпурно-синий».;

- если слово образовано от имени собственного: («лев-толстовский», «вальтер-скоттовские», «нью-йоркская»). Исключение составляют географические названия, образованные из словосочетания существительное и прилагательное: («Великие Луки» – «великолукский», «Сергиев Посад» – «сергиевopосадский», «Старая Русь» – «старорусский»);

- научно-технические термины: «динамо-машина», «вакуум-сушилка», «дизель-электрод», «стоп-кран», «фильтр-пресс»;

- обозначения политических партий и течений: «вице-мэр», «либерально-демократический», «социал-демократ», «национал-социалистический».

- слово, имеющее в первой части оценочное суждение: «горе-жена», «рубаха-парень», «лапочка-дочка», «пайныка-сыночек»;

- если первая производящая основа – обозначение какой-либо латинской буквы: «альфа-самец», «бета-каротин», «гамма-излучение».

2. Сложение усеченных основ. Усекаться могут как обе основы («юннат»), так и какая-либо одна («турфирма»).

3. Можно образовать сложное слово, используя соединительные гласные *о* и *е*. Соединительная гласная *о* используется после основ на твердый согласный (кроме [ж], [ш] и [ц]), а также в немногих сложных словах согласный звук первой основы отвердевает, поэтому пишется соединительная гласная *о*. Буква *о* пишется также после гласной *и* или *е* в ряде сложных слов с первой частью – основой слов



на *-ия* или на *-ей, -ея*, «*бактерионоситель*» («*бактерия*»), «*историография*» («*история*»). Соединительная гласная *e* пишется после основ на мягкий согласный, на *й*, на шипящий звук и *ц*.

4. Иногда соединительные *o* и *e* в сложных словах не используются: их заменяют части производных основ:

- слово образовано из сочетания наречия с именем прилагательным («*малоисследованный*», «*зловеще-гордый*»);

- первая часть – глагол в повелительном наклонении («*перекати-поле*»);

- слово – оттенок цвета. Соответственно, для связи основ используется суффикс («*изжелта-красный*», «*иссиня-черный*»);

- если первая производящая основа – имя числительное в форме родительного падежа: «*десятилетка*», «*семимесячный*», «*тридцатитомный*».

- в ряде случаев слово образовано без данных гласных просто с помощью сложения основ («*психастения*»);

- иногда первая производящая основа – начальная форма имени существительного («*пламяизвергающий*»);

- первая производящая основа может иметь форму какого-либо падежа («*умалишенный*», «*сумасшедший*»);

- слова иноязычного происхождения: *авиа-*, *авто-*, *мото*, *фото-*, *электро-*, *квази-* и другие. Здесь вне зависимости от твердости/мягкости предшествующего согласного остается первоначальный гласный;

- в прилагательных, имеющих отношение к имени собственному: «*бабы-дусин*», «*анны-петровнин*»;

- без соединительной гласной образованы термины типа «*азотсодержащий*», «*вперёдсмотрящий*» и т. п.

В результате изучения особенностей фонетики русского языка сформирована база правил, на основе которой сгенерирован размеченный словарь слов русского языка объёмом более 5 млн. слов с пометами ударения, редукации, смягчения согласных, что позволило использовать ее для обучения нейросети, генерирующей

транскрипцию слов русского языка в формате международного фонетического алфавита.

### 3.2 Общая схема работы системы автоматической генерации транскрипций Cyr2Trans

Для генерации транскрипции слова на русском языке достаточно знать позицию ударения. Исключением из этого правила являются иноязычные и заимствованные слова, а также слова-исключения, т. к. их алгоритм получения транскрипции отличается. Поэтому необходимо иметь модели для построения транскрипции как для простых слов, так и для слов-исключений.

Система автоматической генерации транскрипций, именуемая далее Cyr2Trans, является гибридной, т. к. использует процедурно-декларативный подход с применением словарей транскрипций и нейросетей.

Словари транскрипций, используемые в системе Cyr2Trans, основаны на механизме конечных автоматов – направленных ациклических графов слов (directed acyclic word graphs, DAWG) [143]. Словари выглядят следующим образом: <мама><t><ма+ма>. Словарь омографов отличается от других словарей: <замок><t><за+мок,замо+к>. Также отличается словарь практических транскрипций: <english><t><и+нглиш>.

Системой используются следующие словари транскрипций:

1. Общий словарь транскрипций. Составлен из общедоступных словарей транскрипций (словарь Хагена, Зализняка, АОТ и т. п.), а также дополнен парадигмами входящих в них слов с помощью морфологического анализатора `rumorphy2`. Общее количество пар слов – 4932715.

2. Словарь иностранных слов современного русского языка. Составлен вручную из [144–146], а также дополнен материалом из Сети. Общее количество пар слов – 56916.

3. Словарь практических транскрипций. Составлен из источников [135–142], а также дополнен материалом, находящимся в открытом доступе. Общее количество пар слов – 132640.

4. Словарь омографов. Составлен из источников [147, 148], а также дополнен материалом, находящимся в открытом доступе. В данном словаре используются пары слов из предыдущих словарей (слова должны быть одинаковыми, а позиции ударения – разными). Общее количество пар слов – 11242.

Помимо словарей используются нейронные сети для определения положения ударения (DetAccentNN, пункт 3.3.3), генерации транскрипций для слов-исключений (PhonExcNN, пункт 3.5), а также для практической транслитерации (PractTransNN, пункт 3.4).

Общая схема работы Cyr2Trans приведена на рисунке 3.1.

Функционирование Cyr2Trans заключается в последовательном выполнении следующих действий.

1. Инициализация словарей транскрипций и нейронных сетей.
2. На вход системы подаётся слово (список слов).
3. Разделение слов на подслова (данный этап будет подробно описан ниже).
4. Проверка каждого подслова на наличие символов, входящих в английский алфавит. Если таковых нет – переход на шаг 6.
5. Если подслово латинское, то используется PractTransNN [149]. Результат записывается в массив вариантов транскрипции слов  $T_{\text{sub}} = [t_{\text{sub}0}, \dots, t_{\text{sub}N}]$ , где  $N$  – количество вариантов транскрипции подслова.
6. Проверка на наличие символов только из русского алфавита, а также проводится ёфикация и йфикация, описанные в пункте 2.4.5.
7. Проверка при помощи DAWG на наличие слова в словаре транскрипций (словарь омографов – содержит слова, имеющие одинаковое написание, но разное произношение; словарь исключений – содержит слова, произношение которых отличается от фонетических норм русского языка; словарь практической транскрипции – словарь транскрипций англоязычных слов; общий словарь).
8. Поиск транскрипции слова в словарях. Если слово найдено, переходим на шаг 11, иначе – на шаг 9.
9. В случае, если слово не найдено, то используется DetAccentNN, позволяющая определять позицию ударения в слове.

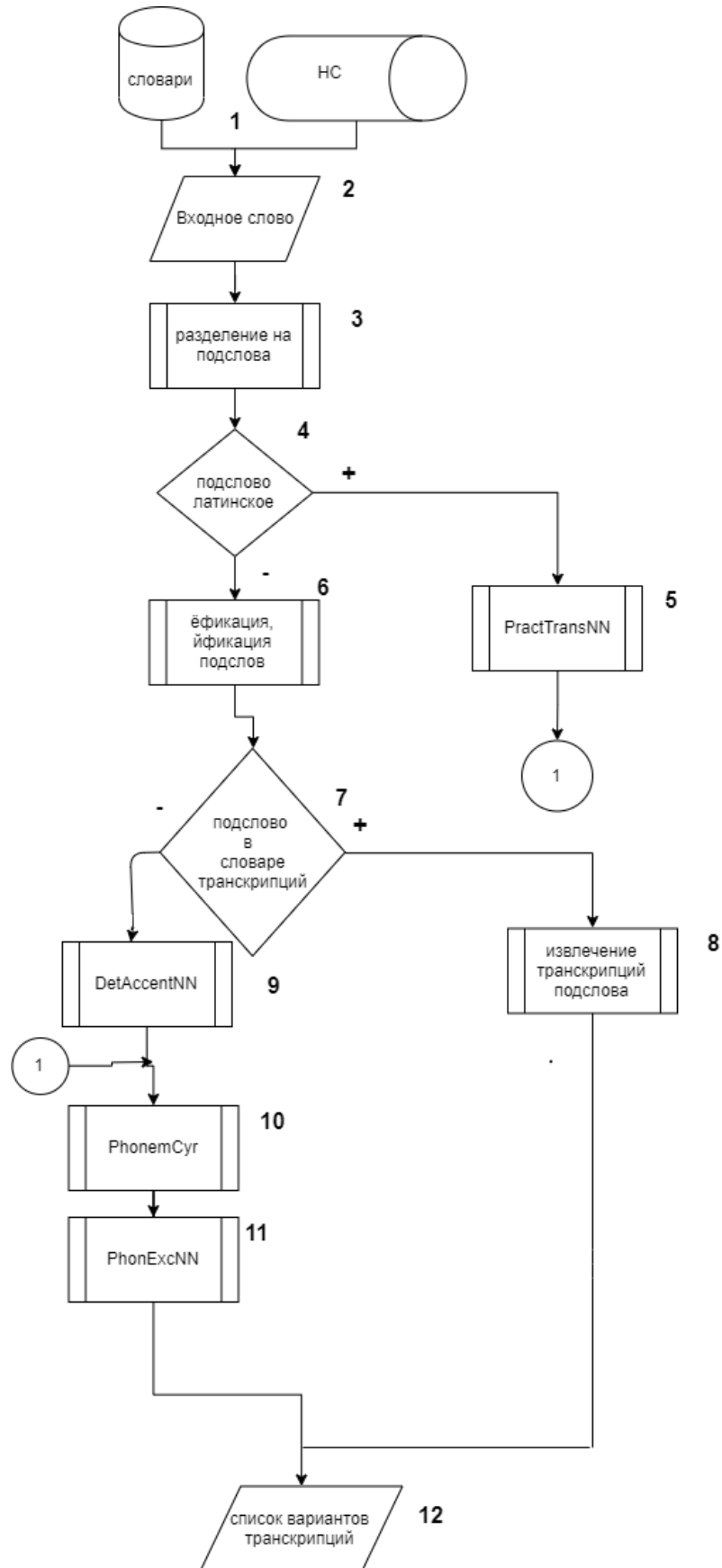


Рисунок 3.1 – Блок-схема работы системы автоматической генерации транскрипций Cyr2Trans

10. Используем PhonExcNN, которая позволяет сгенерировать транскрипцию.  
Переход на шаг 12.

11. Используя разработанную систему PhonemCyr, получаем транскрипцию.

12. Транскрипции для подслов объединяются в транскрипции слов, получаем M-вариантов транскрипции слов ( $T = [t_0, \dots, T_m]$ ).

Например, результатом работы системы Cyr2Trans для слова «высокогорный» является транскрипция, записанная в формате IPA: [V Y\*\*\* S A\*\* K O\* G O! R N Y^ J].

Дальнейшие пункты посвящены детальному описанию каждого этапа работы системы Cyr2Trans.

### 3.3 Разработка метода разделения слов на подслова

#### 3.3.1 Общая схема работы метода разделения слова на подслова

На рисунке 3.2 изображена схема разделения слова на подслова, состоящая из следующих шагов.

1. Загрузка словарей постфиксов, окончаний, суффиксов (для разных частей речи), а также загрузка словарей приставок и корней лемм слов.
2. Инициализация слова или списка слов для дальнейшего разделения.
3. Разделение слова на подслова на основе дефисов (SplitDef)
4. «Ёфикация» и «йфикация» подслов (автоматическое исправление ошибок в словах, связанных с неверным употреблением символов «е» («и») и «ё» («й»), YO\_J).
5. Стемматизация подслов (Stem), извлечение корня (ROOT), суффикса (SUF), постфикса (POST) и приставки
6. Разделение корня слова с использованием корней лемм слов (SplitStemEnd)
7. Формирование двух временных подслов: part1e; part2e.

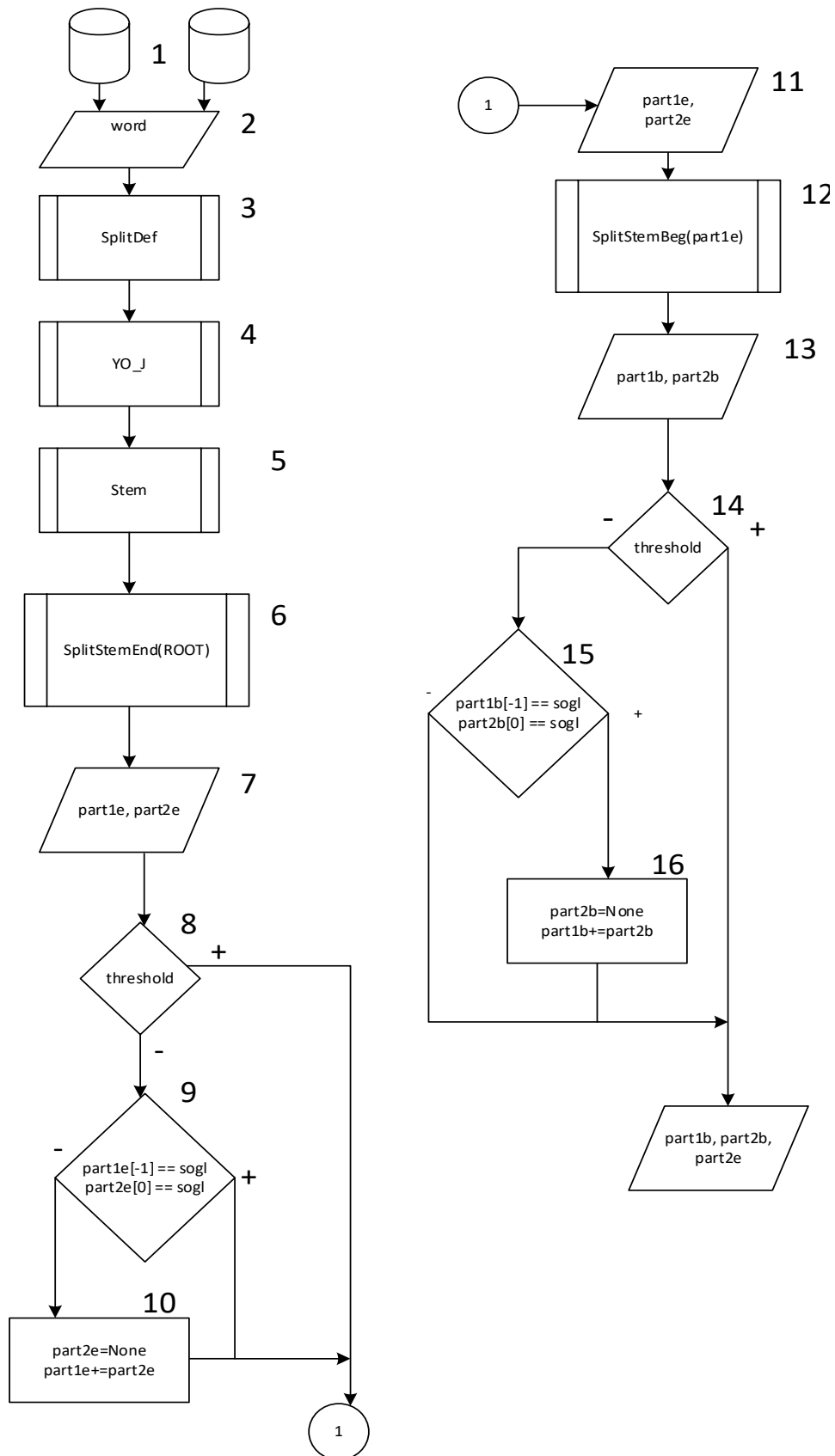


Рисунок 3.2 – Блок-схема алгоритма разделения слов на подслова

8. Проверка подслов на минимальную длину (`threshold`, подслово должно состоять минимум из трёх символов).
9. Проверка, является ли последний символ `part1e` и первый символ `part2e` согласным.
10. Изменение `part1e` и `part2e`.
11. Извлечение `part1e` и `part2e`.
12. Разделение `part1e` с использованием корней лемм слов (`SplitStemEnd`).
13. Формирование двух временных подслов: `part1b`; `part2b`.
14. Проверка подслов на минимальную длину (`threshold`, подслово должно состоять минимум из трёх символов).
15. Проверка является ли последний символ `part1b` и первый символ `part2b` согласным.
16. Изменение `part1b` и `part2b`.
17. Извлечение `part1b`, `part2b`, `part1e`, `part2e`, `PRE`, `SUF`, `POST`, `END`.

Подслово, разделённое дефисом, можно разделить ещё максимум на 4 подслова: подслово#1 (`PRE` + `part1b`); подслово#2 (`part2b`); подслово#3 (`part1e`); подслово#4 (`part2` + `SUF` + `END` + `POST`). Т. е. рассматриваемое слово максимум можно разделить на  $N \times 4$  подслов, где  $N$  – количество подслов, полученных при разделении слова через дефис.

### 3.3.2 Модификация алгоритма стемматизации `SnowballStemmer`

В качестве основы модуля разбиения слов на подслова используется модифицированный алгоритм `SnowballStemmer` [150, 151] – `RusStemmer`. Для этого собраны словари окончаний, приставок, суффиксов и постфиксов для самостоятельных частей речи. Блок-схема алгоритма изображена на рисунке 3.3.

Отличие предложенного алгоритма стемматизации состоит в использовании отдельных словарей суффиксов, постфиксов, окончаний для каждой самостоятельной части речи. Морфоанализатор позволяет выявить, что данное слово может относиться к нескольким частям речи, например, «стали» как глагол и «стали» как существительное. Кроме того, используется общий словарь приставок.

Таким образом, данный алгоритм позволяет разделить любое слово на составные части. Стоит отметить, что при использовании данного алгоритма есть ряд слов-исключений (например, при стемматизации имени «Василий» *-ий-* определяется как суффикс, т. к. для ряда существительных (например, «иридий») имеется такой суффикс).

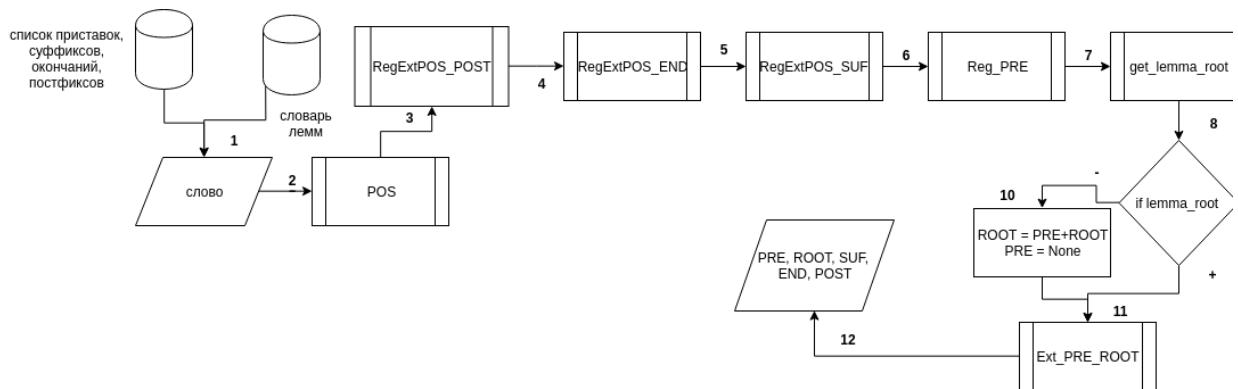


Рисунок 3.3 – Блок-схема алгоритма RusStemmer

Алгоритм RusStemmer работает следующим образом.

1. Загрузка словарей постфиксов, окончаний, суффиксов (для разных частей речи), а также загрузка словарей приставок и лемм слов.
2. Подача слова (массивов слов) на стемматизацию.
3. Определения списка возможных частей речи для слова (POS).
4. Извлечения постфиксов из слова (RegExtPOS\_POST).
5. Извлечение окончаний из слова (RegExtPOS\_END).
6. Извлечение суффиксов из слова (RegExtPOS\_SUF).
7. Определение возможных приставок в слове (Reg\_PRE).
8. Определение леммы для корня слова (get\_lemma\_root).
9. Поиск леммы в словаре лемм.
10. Изменение информации о корне и приставке.
11. Извлечение приставки и корня (Ext\_PRE\_ROOT).
12. Извлечение списка вариантов стеммов.



### 3.4 Разработка нейросетевой модели для определения позиции ударения

Для определения позиции ударения в слове разработана нейросеть DetAccentNN, на вход которой подаётся слово, а на выходе – позиция ударения. DetAccentNN имеет архитектуру Transformer (Рисунок 3.4).

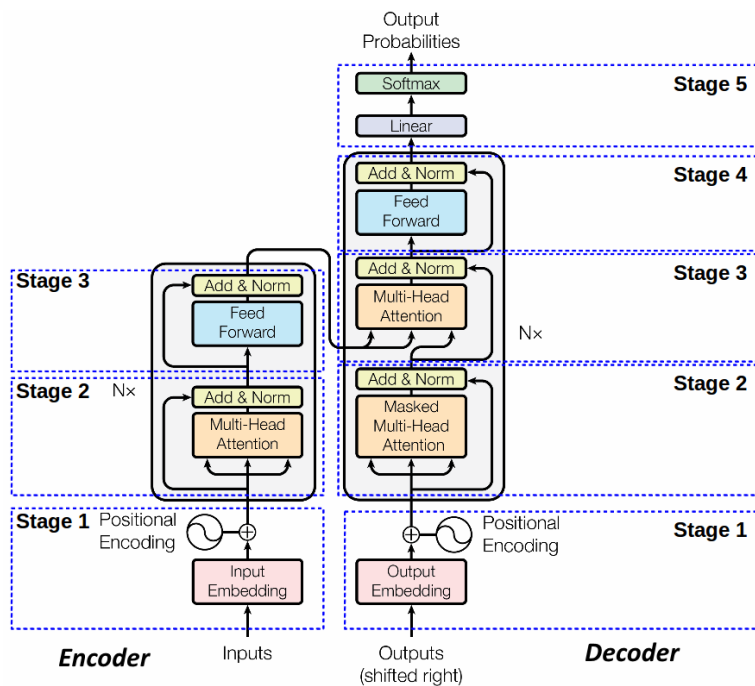


Рисунок 3.4 – Архитектура Transformer

Transformer [90] использует отдельные модели энкодера и декодера. В качестве стандартных архитектур для энкодера и декодера Transformer использует полносвязные слои. Transformer уменьшает количество последовательных операций для привязки двух символов из последовательностей ввода-вывода. Это достигается при помощи механизма многозадачности, который позволяет моделировать зависимости независимо от их расстояния во входном или выходном предложении.

В Transformer энкодер и декодер состоят из стека одинаковых слоёв. Каждый из этих слоёв состоит из двух общих типов подслоёв:

- механизма многослойного обучающего внимания (multi-head attention);
- позиционной полносвязной NN прямого распространения (feed forward).

Главное отличие декодера от энкодера в Transformer – использование слоя с механизмом маскирующего многослойного внимания (masked multi-head attention), который позволяет «обращать внимание» на специфичные сегменты из энкодера [91]. Это возможно благодаря тому, что masked multi-head attention маскирует будущие токены посредством блокирования информации токенов, которые находятся справа от вычисляемой позиции.

DetAccentNN имеет следующие параметры:

- количество скрытых слоёв: 512;
- размер входных векторов для энкодера: 31;
- максимальное кол-во символов в слове: 32;
- размер батча: 128;
- количество блоков в энкодере: 5,
- количество блоков в декодере: 3;
- количество заголовков обучающегося внимания: 4;
- функция активации для скрытых слоёв: rectified linear unit;
- функция потерь: разреженная кросс-энтропия;
- коэффициент dropout-регуляризации: 0.2;
- функция регуляризации: L2-регуляризация;
- оптимизатор для градиентного спуска: AdamBound;
- коэффициент скорости обучения: 0.0001;
- количество эпох: 100 тыс.

Для улучшения производительности архитектура Transformer модифицирована за счёт:

- применения градиентного отсечения (clip gradient) [152]. Это общепринятый метод, направленный на решение проблемы «взрывных градиентов» (vanishing gradients). По сути, обрезая градиенты или устанавливая пороговые значения до максимального значения, мы предотвращаем экспоненциальный рост градиентов и переполнение (равенство градиентов нулю), или превышение крутых обрывов в функции оценивания (Рисунок 3.5);

- увеличения количества блоков в энкодере.

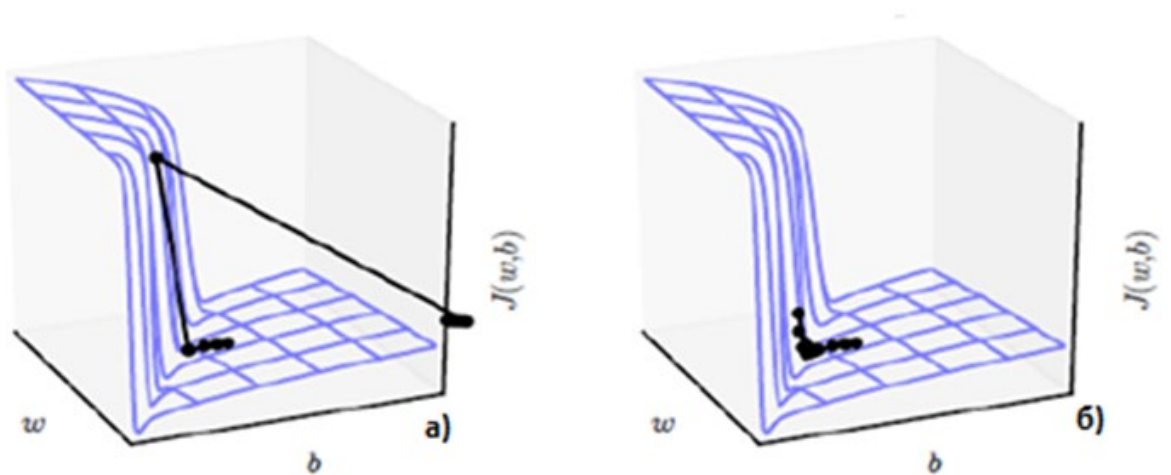


Рисунок 3.5 – Значение функции потерь  $J(w, b)$  без градиентного отсечения (а), с градиентным отсечением (б)

В качестве обучающих данных использован составленный набор пар (слово и позиция ударения), извлечённых из найденных в Сети фонетических словарей. Общий набор слов составил более 5 млн. Данные предварительно преобразованы: удалены неконтекстные пары ударений, исправлены неверно указанные позиции ударений; удалены дубликаты; символы приведены к нижнему регистру. Обучающая выборка составила 95% от общего объёма данных.

Для оценки результатов использовалась метрика WER – отношение количества слов с неверным ударением к общему количеству слов (Word Error Rate). На рисунке 3.6 приведены результаты тестирования.



Рисунок 3.6 – Зависимость loss-функции и WER от количества эпох на тестовой выборке

Для сравнительной оценки результатов обучена оригинальная модель Transformer с теми же гиперпараметрами, что и у модифицированной. Как показали исследования, предложенная модификация позволила снизить показатель WER до 0.08, повысив точность модели по сравнению со стандартной на 10%.

### 3.5 Разработка метода получения практической транскрипции для вставок на латинице

На сегодняшний день в любом тексте на русском языке (статья, книга, новостная лента и т.п.) можно встретить большое количество вставок на латинице, чаще всего – на английском языке, которые, в основном, представлены названиями компаний и организаций («Apple», «Manchester United»), масс-медиа («Forbes»), географическими названиями («New York»), произведениями («Yesterday»), компьютерными программами («Microsoft Office»), интернет-сервисами («Amazon»), именами собственными («James Bond») и т.п. Реже встречаются цитаты или фразы на английском языке («To be or not to be»).

Данные вставки осложняют сбор данных для задач, связанных с обработкой естественного языка (например, классификация текста), а также с задачами распознавания и синтеза речи (сбор данных для формирования языковой модели;

текстовая разметка аудиоданных). То есть для того, чтобы система синтеза/распознавания речи могла «работать» со словами, написанными на латинице, их необходимо трансформировать в ту же фонетическую систему, что и для русских слов. Для решения этой проблемы систему синтеза/распознавания речи, как правило, дополняют отдельным модулем для английского языка. Например, для системы распознавания речи необходим набор обучающих данных для дополнительного языка со своей транскрипцией и языковой моделью, а также модуль для классификации языка.

В связи с вышеизложенным возникла необходимость создания автоматической системы формирования практической транскрипции из слов, написанных на латинице, с использованием лингвистических знаний совместно с глубоким обучением без дополнительного языкового модуля. Универсальность разработанного метода заключается в том, что на выходе блока нормализации формируются «нормализованные» слова на кириллице, которые далее обрабатываются по тем же правилам, что и обычные слова русского языка. Благодаря этому одни и те же правила практической транскрипции можно включить в любую русскоязычную систему синтеза/распознавания речи, независимо от фонетической транскрипции, которая в ней используется.

### 3.5.1 Недостатки существующих методов получения практической транскрипции

Опишем недостатки существующих методов получения практической транскрипции.

Наиболее распространённым методом для трансформации английских слов в кириллицу является транслитерация, когда символу или набору символов из одного алфавита ставится в соответствие символ или набор символов из другого алфавита, причём соответствие осуществляется только по их графическому сходству. Недостаток данного подхода состоит в том, что восстанавливается исходное написание слова, но без учёта его произношения.

В качестве одного из методов автоматической транскрипции используют перевод, при котором некоторому часто встречающемуся имени ставится в соответствие его эквивалент, устоявшийся в языке, на который осуществляется перевод. Недостаток данного подхода состоит в том, что восстанавливается лишь семантическая информация, без учёта произношения.

Другим способом является словарный метод, когда словам из языка N приводится в соответствие слова языка M при помощи некоего словаря. Недостаток данного подхода состоит в том, что практически невозможно учесть все имеющиеся слова, а в случае формирования такого словаря данный алгоритм будет иметь высокую вычислительную сложность.

Наиболее оптимальным подходом является метод транскрипции, в котором звучание слова в языке N записывается средствами языка M. Однако язык M может быть знаком лишь узкому кругу специалистов, поэтому используют метод практической транскрипции, который генерирует запись транскрипции иноязычных слов с помощью орфоэпических норм языка N, используя только обычные знаки (буквы) этого языка без введения дополнительных знаков. Однако, в настоящее время для построения таких систем отсутствуют готовые решения, что привело к необходимости разработки метода получения практической транскрипции, учитывающего особенности произношения носителей русского языка.

При разработке метода трансформации английских слов в кириллицу возникли следующие проблемы:

- неполное соответствие фонемного состава двух языков;
- использование запрещённых в целевом языке морфем (ряд затруднений с обозначением звуков, отсутствующих в данном языке). Поэтому зачастую при транскрипции слов приходится ставить в примерное соответствие звукам одного языка звуки другого;
- частичная потеря информации как в виде строк, так и в виде фонетической информации (длительность, палатализованность, высота тона и др.);
- возможное отсутствие семантики для транслитерированного слова;

- отсутствие единого стандарта транскрипции и транслитерации. Правила для транскрипции на кириллицу либо ещё совсем не разработаны, либо разработаны, но вызывают много вопросов (т.е. даны лишь основные соответствия, а правильная передача многих буквосочетаний остается неясной);

- различные варианты произношения одного и того же слова, связанные с носителями языка или традициями. В подобных случаях при транскрипции возникают несколько потенциально возможных вариантов транскрипции, выбрать один из которых не представляется возможным;

- отсутствие взаимно однозначного соответствия при транскрипции слова с исходного языка на язык перевода и обратно. То есть, если слово одного языка транскрибировать в другой язык, а затем транскрибировать его обратно, то полученное слово в значительном количестве случаев будет отличаться от исходного;

- транскрипции с одного языка (например, английского) имён собственных, исконно принадлежащих другому языку (например, испанскому).

Проблеме трансформации английских слов в кириллицу посвящены работы, в которых упор делается на использование фонетических правил [133, 134, 153] или на классическое машинное обучение [154]. Стоит выделить тот факт, что данные работы, в основном, направлены на трансформацию имён собственных, написанных на английском языке.

### 3.5.2 Особенности произношения английских вставок носителями русского языка

Для разработки метода получения практической транскрипции необходимо учитывать орфоэпические нормы русского языка. На основе работ [155–157] выделены следующие ключевые особенности произношения английских вставок носителями русского языка.

1. Фонетическая модификация английских слов, заимствованных в русскую речь. Важным фактором является распространенность англоязычных слов в

повседневной жизни носителей русского языка. Чем выше частотность слова, тем выше вероятность того, что оно будет озвучено диктором «по-русски» (для английских звуков, отсутствующих в русском языке, подыскиваются ближайшие по звучанию звуки-замены). Например, для слов «bluetooth», «word», «amazon», «twitter», «microsoft», «facebook» и т.п. Однако даже менее распространенные слова («WhatsApp», «Slack» и др.), как правило, произносятся «по-русски». Так, например, в окончании «-ing», как правило, произносится русское [инк], т.е. происходит редукция («драйвинг» — «дра+йвинк»). То же можно сказать о фразе «I and you» («I & U» – «а+й э+нт ю+»).

2. Существенным фактором является длина англоязычной вставки. Целые фразы на английском языке и просто длинные словосочетания редко употребляются в русскоязычных текстах. Тем не менее, длинные англоязычные фрагменты иногда озвучиваются русифицировано, особенно если данные словосочетания достаточно известны («Work & Travel», «Apple Watch», «Amazon Kindle Paperwhite»).

3. Целый ряд английских слов заимствован со смещением ударения на последний (или предпоследний) слог. Например, «email» ([ˈi:meɪl]; «и+мэйл») – «имэ+йл»; «facebook» [ˈfeɪsbʊk]; «фэ+йсбук») – «фэйсбу+к».

4. Английская фонема [w], которую лингвисты предлагают передавать через «в» перед буквой «у» и через «у» во всех остальных случаях («woods» – «ву+дс», «windows» – «уи+ндос»). Однако, на практике, данная фонема произносится как «в». Например: «twitter» («туи+ттэр», «тви+ттэр»), «windows» («уи+ндос», «ви+ндос»)

5. Суффиксы «-er» и «-ed». По фонетическим правилам фонема [ə] передается транслитерацией «э». А в таких словах, как partner [ˈpɑ:tnə]/па+ртнер/ гласный [e] смягчал предшествующий согласный [n] и по правилам ассимиляции смягчался и предшествующий зубной [t] ([пá рт'н'эр]). На практике, при произнесении англоязычных вставок дикторы смягчают согласные по большей части так же, как и в русском языке: Christies – [кр'йс'т'ьс]; Acoustic – [ьк'ус'т'ьк]. Но для таких суффиксов, как -er («эр»), -ed («эд»), -ment («мэнт») смягчение обычно не происходит.

6. На практике, фонема [ð] передается звуком «з», а фонема [θ] через звуки «з» и «с». Пример Bluetooth - «блютуз».



7. Английские аббревиатуры, являющиеся словом или состоящие из более, чем одного слога, произносятся «по-русски». В противном случае – каждой букве аббревиатуры ставится в соответствие её транскрипция как отдельного звука («IBM» - «а+й би+ э+м»).

Стоит отметить, что большое значение имеет уровень владения английским языком говорящего. Но, на практике, говорящий лишь в редких случаях полностью переключается на английский язык вне зависимости от его уровня владения языком.

При озвучивании отдельных английских слов и не очень длинных словосочетаний актуально использовать практическую транскрипцию английского языка, из тех соображений, что носители русского языка привыкли озвучивать и воспринимать на слух английские слова в их русифицированной форме. Следовательно, транскрипция английских слов по правилам англо-русской практической транскрипции, может повысить точность распознавания речи.

### 3.5.3 Общая схема работы метода получения практической транскрипции

Разработанный метод преобразует английские слова, написанные в виде латиницы, в формат кириллицы, используя практическую транскрипцию. В отличие от обычной транслитерации, практическая транскрипция основывается на произношении слов, что позволяет распознавать английские звуки, используя единую лингвистическую и акустическую базу. После того как английские слова будут преобразованы в последовательность символов кириллицы, на всех этапах их дальнейшей обработки они ничем не будут отличаться от русских. Предложенный метод может быть использован для любой системы синтеза или распознавания речи.

Одним из преимуществ предлагаемого метода обработки слов на латинице является его оптимальность: вместо написания дополнительного языкового модуля используется система англо-русской практической транскрипции для перевода английских слов на кириллицу. Кроме того, предлагаемый метод универсален, то есть может быть использован в любой русскоязычной системе

синтеза/распознавания речи: после того, как вставка на латинице переведена в кириллическую графическую систему, она может обрабатываться на всех дальнейших этапах синтеза речи по тем же правилам, что и обычные (нормализованные) русские слова на кириллице.

Разрабатываемый метод является декларативно-процедурным, он использует как словарь, так и правила англо-русской практической транскрипции. Для уменьшения сложности используется нейросетевой подход и механизм конечных автоматов. В качестве основной нейросетевой архитектуры используется архитектура типа энкодер-декодер – Transformer [158, 159].

Предложенный метод получения практической транскрипции использует нейросетевую модель с архитектурой Transformer для трансформации слов на латинице в формат кириллицы.

Стоит отметить, что при сборе данных для обучения составляется словарь, использующий DAWG. В таблице 3.1 приведены примеры записей этого словаря.

Таблица 3.1 – Пример строк из словаря

Слово на латинице	«Нормализованное» слово
Airways	эйрвэйс
Bloomberg	блумберг
British	бритиш
Microsoft	майкрософт

Схема алгоритма построения практической англо-русской транскрипции изображена на рисунке 3.7.

Алгоритм состоит из следующих шагов.

1. Считывается документ (T) после проведения нормализации, т.е. в документе отсутствуют цифро-буквенные комплексы (20th), сокращения, неконтекстные токены и т.п.

2. T разделяется на предложения ( $S = \{s_i\}$ ,  $i = 0, \dots, N$ , где N – общее количество предложений).

3.  $s_i$  разделяется на токены ( $W = \{w_j\}$ ,  $j = 0, \dots, M$ , где  $M$  – общее количество токенов в  $s_i$ ), используя набор фильтров.

4. Используя модель для определения языка, определяем язык текста для  $s_i$ , если модель детектирует не русский язык (большее количество слов не из русского алфавита) считываем  $s_{i+1}$  и повторяем шаг 4 для этого предложения. То есть переходим к следующему шагу, если текущее предложение на русском языке.



Рисунок 3.7 – Общая схема метода трансформации английских вставок

5. Проверяем, содержит ли  $w_j$  символ не из русского алфавита, и добавляем токен в переменную  $w\_cur$ . В случае, когда токен  $w_j$  представляет собой русско-английскую структуру (best-вещь, вещь-best), то  $w_j$  разделяется на токены, состоящие из символов одного алфавита без использования фильтров, применяемых для

разделения предложения на токены. После разделения  $w\_sig$  получаем массив  $E$  ( $E = \{e_k\}$ ,  $k = 2, \dots, L$ , где  $L$  – общее количество токенов, получившихся при разделении). Затем аналогично проверяем первый символ для  $e_k$ , если первый символ  $e_k$  не является символом для русского алфавита, то переходим к следующему шагу.

6. Если  $e_k$  не является аббревиатурой, то применяем модель трансформации токена  $e_k$  в формат кириллицы. В итоге получаем слово в формате кириллицы и заносим его в изменённый массив  $E$ . Соответственно, повторяем шаги 6 и 7 для всего массива  $E$ .

7. Объединяем массив  $E$  в строку через разделитель «-» получаем токен  $w_j$  в формате кириллицы.

8. Получаем массив транспонированных в кириллицу токенов  $W_{ed} = \{w_j\}$ ,  $j = 0, \dots, M$ , которые объединяем и получаем транспонированное в кириллицу предложение  $s_i$ .

9. Получаем массив трансформированных в кириллицу предложений  $S = \{s_i\}$ ,  $i = 0, \dots, N$ , которые объединяем и получаем трансформированный в кириллицу документ  $T_{ed}$ .

#### 3.5.4 Разработка нейросетевой модели для генерации практической транскрипции

Для обучения нейросети необходимо преобразовать входные данные (слова) в вектор чисел. В связи с этим разработан алгоритм кодирования слов, который, в первую очередь, направлен на оптимизацию процесса обучения модели при помощи мини-пакетного типа обучения [26]. При трансформации данных для мини-пакетного типа обучения стоит помнить об изменении длины фразы в массивах данных. Чтобы разместить фразы разных размеров в одном пакете, необходимо сделать матрицу  $E$  длины  $L_{max} \cdot b_s$  ( $L_{max}$  – максимальное количество слов в фразе,  $b_s$  – размер пакетов), где фразы короче  $L_{max}$ , должны быть дополнены нулями после индекса «EOS». Однако, если просто преобразовать фразы в матрицы путем преобразования слов в их индексы и сделать нулевое заполнение, то тензор

будет иметь форму  $(batch\_size, max\_length)$ , и при индексировании первого измерения будет возвращаться полная последовательность по всем временным шагам. Однако необходимо иметь возможность индексировать пакет по времени и по всем последовательностям в пакете. Поэтому переносим нашу форму входного пакета в  $(max\_length, batch\_size)$ , чтобы индексирование по первому измерению возвращало шаг по времени для всего множества входных данных в пакете (Рисунок 3.8). Для этого выполняется транспонирование матрицы  $E$ :  $F = E^T$ .



Рисунок 3.8 – Изображение операции преобразования матрицы индексов слов для мини-пакетного обучения

Алгоритм кодирования состоит в следующем.

- 1) Используем  $W$  из 3-го этапа алгоритма токенизации.
- 2) Каждое  $w_1$  находим в заранее созданном словаре (около 7 млн. уникальных словоформ, а также метки «BOS» (начало фразы), «EOS» (конец фразы), «PAD» (токен для разделения предложений), «UNK» (токен для обозначения вне словарного слова) и символы («.», «!», «?», «,», «:», «>» и т.п.)). В случае если слово отсутствует или его частота появления в тексте меньше заданного порога ( $thresh_w$ )  $w_1 = \text{«UNK»}$ . Иначе извлекается индекс слова. В итоге получается массив индексов слов  $U = \{u_m\}$ .

- 3) Информация записывается в матрицу индексов  $M$ , длиной  $N \times R$ , где  $N$  - общее количество фраз;  $R$  - общее количество слов во фразе:

$$R = N_s \cdot L_p, \quad (3.1)$$

где  $N_s = 3$  (максимальное количество предложений);  $L_p = 30$  (максимальное количество слов в предложении).

Матрица  $M$ , затем транспонируется и получаем  $F_{inp}$ . Помимо матрицы  $M$ , формируется матрица  $L$ , размером  $N \times 1$ , содержащая информацию о количестве слов в каждой  $p_i$ .

4) Аналогичным способом формируем матрицу для ответов  $F_{out}$ , а также тензор двоичной маски ( $B$ ) и максимальную длину ответа. Бинарный тензор маски имеет ту же форму, что и выходной целевой тензор, но каждый элемент, являющийся индексом «PAD», равен 0, а все остальные равны 1. Таким образом формируем маску предложений для фраз. Данная маска маскирует последовательность, используя значение для пропуска временных шагов. Для каждого временного шага во входном тензоре, если все значения во входном тензоре на этом временном шаге равны значению маски, тогда временной шаг будет замаскирован (пропущен) во всех нижестоящих слоях (если они поддерживают маскирующий).

Для обучения нейросети подготовлен набор данных, собранный из информации, извлеченной из работ [135–142]. В общей сложности количество элементов обучающей выборки составило около 300 тыс. пар.

В качестве основной архитектуры для обучения использовалась архитектура Transformer. При обучении ANN использовались следующие гиперпараметры:

- количество скрытых слоёв: 512;
- размер входных векторов для энкодера: 26;
- размер входных векторов для декодера: 31;
- размер батча: 128;
- количество блоков в энкодере (энкодер переводит входной сигнал в более компактное представление, при этом сохраняя семантическую информацию): 5,
- количество блоков в декодере (восстанавливает исходный сигнал из компактного представления): 3;
- количество заголовков обучающегося внимания: 4;

- функция активации для скрытых слоёв: rectified linear unit;
- функция активации выходного слоя: softmax;
- функция потерь: разреженная кросс-энтропия;
- коэффициент dropout-регуляризации: 0,2;
- функция регуляризации: L2-регуляризация;
- оптимизатор для градиентного спуска: AdamBound;
- коэффициент скорости обучения: 0,0001;
- количество эпох: 100 тыс.

Дополнительно нейросеть улучшена при помощи техник teacher forcing [152], градиентного отсечения и метода beam-search.

Teacher forcing заключается в использовании истинных значений (ground truth) в качестве входных данных для каждого временного шага, а не выходных данных нейронной сети, т.е. с некоторой вероятностью, установленной отношением принуждения учителя, используется текущее целевое слово в качестве следующего ввода декодера, а не текущее предположение декодера, помогая в более эффективном обучении.

Метод beam search, добавляемый перед выходным слоем, предназначен для выбора следующего символа на основе вектора весов, порождаемого уровнем проекции. В результате запоминается несколько вариантов наиболее вероятных символов. После того, как генерация вариантов последовательности закончена, используется один из вариантов алгоритма Витерби для выбора наиболее вероятной последовательности с учетом контекста всей последовательности.

Другой особенностью данной архитектуры является совместное применение обучения с учителем и обучения с подкреплением, реализованным в RL-block. Схема обучения этой модели представлена на рисунке 3.9.

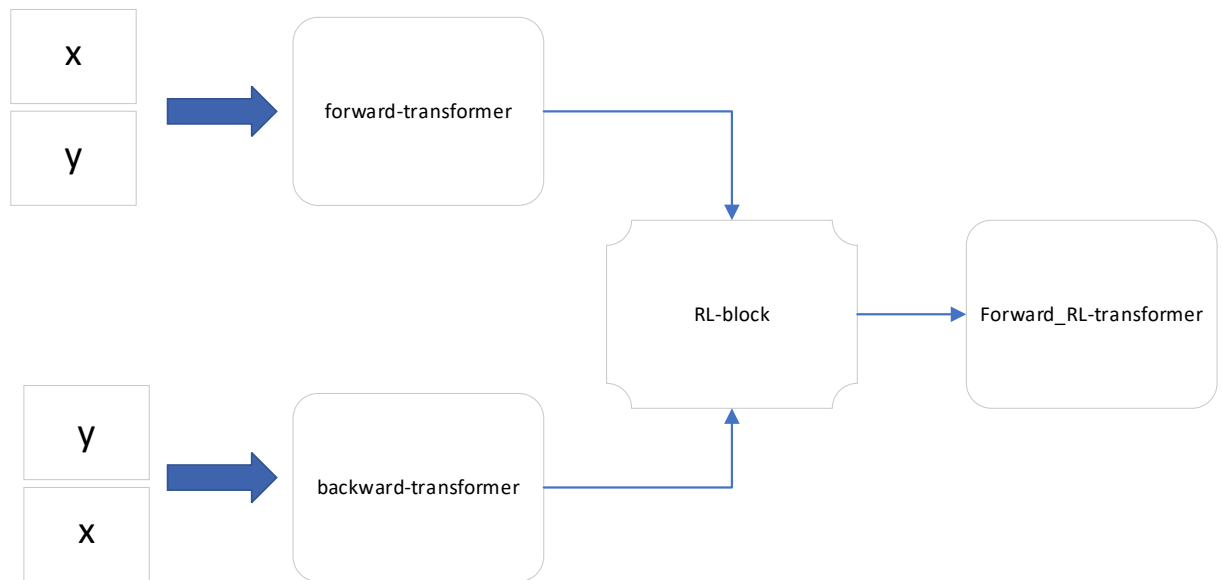


Рисунок 3.9 – Общая схема обучения модели с учителем и обучения с подкреплением

На схеме *forward-transformer* – ANN-модель для генерации транскрипции для слов, обученная на парах  $x$ – $y$  (слово-транскрипция); *backward-transformer* – ANN-модель для генерации слов для транскрипций, обученная на парах  $y$ – $x$ ; *RL-block* – механизм обучения с подкреплением; *forward\_RL-transformer* – итоговая ANN-модель генерации транскрипции.

Механизм *RL-block* используется для переопределения вероятностей, т.е. для увеличения правдоподобия «хороших» сценариев (обладающих высокой наградой, *reward*,  $R_t$ ) и понизить правдоподобие «плохих» сценариев (*policy gradient*):

$$\nabla_{\theta} J(\theta) = E_{T \sim p_{\theta}(T)} [\nabla_{\theta} \log p_{\theta}(T) R_T], \quad (3.2)$$

где  $p(T)$  – это вероятность того, что будет реализован сценарий  $T$  при условии параметров модели  $\theta$ , т. е. функция правдоподобия.

Двигаясь вверх по этому градиенту, мы повышаем логарифм функции правдоподобия для сценариев, имеющих большой положительный  $R_t$ .

Данный механизм *RL-block* заключается в следующем.

1) Собирается список возможных вариантов слов для соответствующих транскрипций (слово-транскрипция), который затем кодируется.



2) Дополнительно к `forward_dia`, обучается `backward-dia`, используя реверсный набор данных для обучения.

3) Используя закодированный набор слово-транскрипция, при помощи `forward_model` генерируется набор транскрипций, а также выводится `loss`.

4) Сравнение векторных расстояний. Вычисляется косинусное расстояние между векторами признаков, извлечённых из выходного слоя (`vect_o`) и предпоследнего скрытого слоя (`vect_h`; вектора признаков сжимаются до минимального размера вектора из двух вышеуказанных векторов):

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}, \quad (3.3)$$

где  $A$  – `vect_o`; а  $B$  – `vect_h`.

На основе этого вычисляется промежуточный `reward` ( $rs_1$ )

$$rs_1 = \begin{cases} -similarity; & \text{if } similarity < 0, \\ -\log(similarity); & \text{if } similarity > 0. \end{cases} \quad (3.4)$$

5) семантическая когерентность. На этом этапе промежуточный `reward` ( $rs_2$ ) вычисляется с использованием `backward_transformer`. Предсказывается слово для транскрипции с соответствующей величиной `loss`. А также используются данные из `forward_transformer`:

$$rs_2 = \frac{forw\_loss}{forw\_res} + \frac{back\_loss}{back\_res}, \quad (3.5)$$

где `forw_loss`, `back_loss` – величина `loss` при использовании `forward_transformer` и `backward_transformer`; `forw_res`, `back_res` – результирующий вектор для `forward_transformer` и `backward_transformer`.

6) Подсчёт финального `reward`

$$rs_{end} = \frac{(rs_1 + rs_2)}{2}. \quad (3.6)$$

7) Формирование списка  $N$  размера финальных `rewards` (`rewlist`)

$$rewlist = [rs_{end}[0] \dots rs_{end}[N]]. \quad (3.7)$$

8) Получение среднего `reward`

$$rew_{mean} = \frac{\sum rewlist}{N}. \quad (3.8)$$

9) Пересчитывается  $loss$  для `forward_transformer`, на основе которых перестраивается модель.

$$loss_{rl} = forward_{loss} \cdot rew_{mean}. \quad (3.9)$$

В итоге, после всех вышеописанных действий переобучаем модель `forward_transformer`, и получаем модель `forward_rl_transformer`.

Для сравнительной оценки результатов обучена оригинальная модель `Transformer` с теми же гиперпараметрами, что и у модифицированной. Для оценки результатов использовались метрики: WER – отношение количества неверно трансформированных слов к общему количеству слов (Word Error Rate) и PER – отношение количества неверно трансформированных символов к общему количеству символов (Phoneme Error Rate), а также значения функции потерь. Графики результатов тестирования приведены на рисунке 3.10. На рисунках синим цветом изображены графики для стандартной модели, оранжевым – для модифицированной.

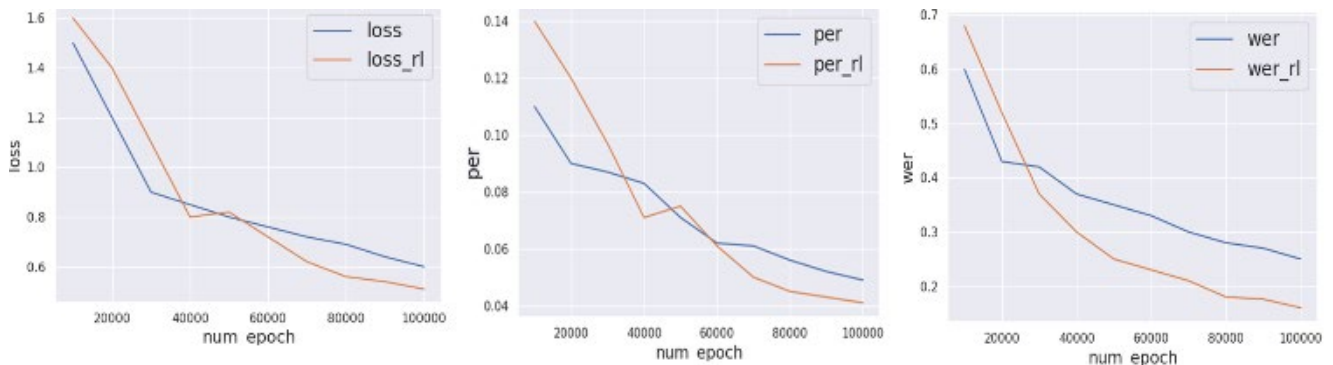


Рисунок 3.10 – Зависимость  $loss$ -функций, метрик PER и WER от количества эпох

Из графиков видно, что после 60000 эпохи по всем трем критериям качества модифицированная модель превосходит стандартную. Предложенный подход для создания автоматической системы формирования практической транскрипции слов английского языка позволяет получать англо-русскую практическую транскрипцию с точностью более 90% для слов и более 95% для символов. Такое высокое качество обеспечивает учет орфоэпических норм русского языка не только на основе фонетических правил, но и с использованием глубокого обучения.

Предложенная технология получения практической транскрипции слов английского языка может быть использована в системах синтеза/распознавания русской речи с целью адаптации англоязычных слов на этапе формирования «нормализованных» слов на кириллице для их дальнейшей обработки системой по правилам, применяемым для слов русского языка. Универсальность данного метода заключается в том, что на выходе блока нормализации формируются «нормализованные» слова на кириллице, которые далее обрабатываются по тем же правилам, что и обычные слова русского языка. Благодаря этому одни и те же правила практической транскрипции можно включить в любую русскоязычную систему синтеза/распознавания речи, независимо от фонетической транскрипции, которая в ней используется.

### 3.6 Разработка нейросетевой модели получения транскрипции слов-исключений

Для получения транскрипции слов-исключений создана и обучена нейросеть PhonExcNN [149]. Для ее обучения собран набор данных, состоящий из слов, отличающихся от фонетических норм русского языка. Данный набор расширен за счёт генерации парадигм для исходных слов при помощи морфоанализатора `rumorphy2`. Сгенерированные парадигмы просмотрены и удалены их неверные варианты. Общий объём слов составил около 10 тыс.

Модель, используемая для PhonExcNN, имеет ту же архитектуру и параметры, что и PractTransNN. Для сравнительной оценки результатов обучена оригинальная модель Transformer с теми же гиперпараметрами, что и у модифицированной. Результаты тестирования сети PhonExcNN приведены на рисунке 3.11. На рисунках синим цветом изображены графики loss-функций, метрик PER и WER для стандартной модели, оранжевым – для модифицированной.

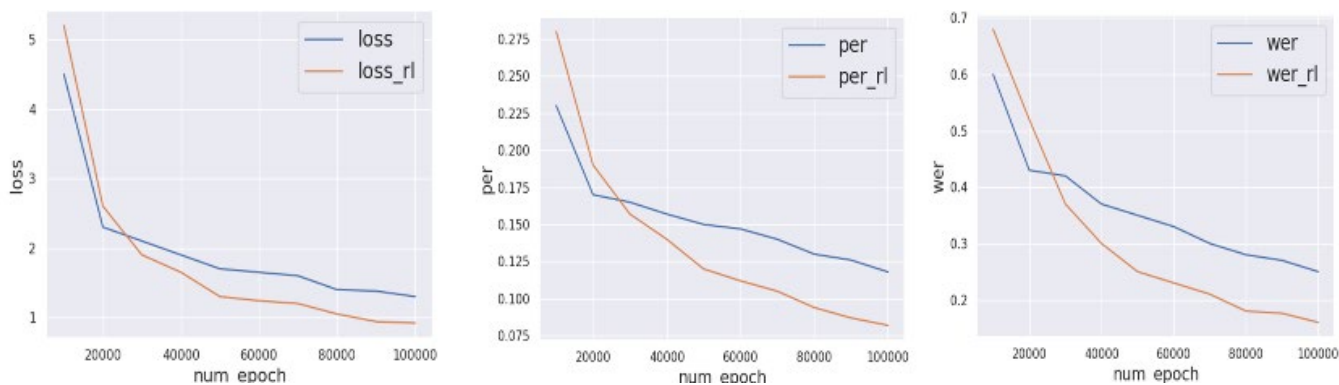


Рисунок 3.11 – Зависимость loss-функций, метрик PER и WER от количества эпох

Из графиков видно, что после 20000 эпохи по всем трем критериям качества модифицированная модель превосходит стандартную. Результаты тестирования показали, что предложенная техника модернизации моделей типа sequence to sequence на основе внесения изменений в структуру алгоритма построения позволила повысить точность обученной модели генерации транскрипций для слов-исключений по критерию PER на 9%, по критерию WER – на 3%.

### 3.7 Выводы к главе 3

1. Сформирована база правил орфоэпических норм произношения слов русского языка, на основе которой сгенерирован размеченный словарь слов русского языка объемом более 5 млн. слов с пометами ударения, редукции, смягчения согласных. Словарь использован для обучения нейросети, генерирующей транскрипцию слов русского языка в формате международного фонетического алфавита в соответствии с особенностями русского языка.

2. Сформированы словари транскрипций: общий, для слов-исключений, практической транскрипции и омографов объемом около 5 млн. пар слов.

3. Предложен гибридный метод к созданию системы автоматической генерации транскрипций, использующий словари и нейросети для определения положения ударения, генерации транскрипций для слов-исключений и практической транслитерации.

4. Предложен алгоритм стемматизации слов для разделения слов на подслова, использующий отдельные словари окончаний, приставок, суффиксов и постфиксов для каждой самостоятельной части речи. В результате работы алгоритма для любого слова русского языка происходит его разбиение на морфемы и генерация списка стеммов.

5. Для автоматического определения позиции ударения разработана нейросетевая модель с модернизированной архитектурой Transformer за счет использования методов градиентного отсечения и teacher forcing для оптимизации параметра скорости обучения, что позволило снизить показатель WER до 0.08 и повысить точность генерации транскрипции на 10% по сравнению со стандартным подходом.

6. Сформирована база правил получения англо-русской практической транскрипции с учетом орфоэпических норм русского языка, на основе которой создан словарь пар слов объемом около 300 тыс., используемый для обучения нейросети, генерирующей практическую транскрипцию.

7. Усовершенствована sequence-to-sequence модель для генерации практических транскрипций англоязычных слов и слов-исключений за счет внесения изменений в структуру алгоритма её построения и применения механизма обучения с подкреплением, что позволило повысить точность модели по критерию количества ошибочно сгенерированных символов на 0,8% и 3%, по критерию неправильно сгенерированных слов на 0,6% и 9% соответственно.

8. На основе предложенных алгоритмов и моделей разработана система автоматической генерации транскрипций Cyr2Trans, учитывающая особенности фонетики русского языка, особенности произношения слов-исключений и английских вставок носителями русского языка. Система генерирует транскрипцию в формате IPA и может быть использована разработчиками систем распознавания/синтеза русской речи.

## ГЛАВА 4

РАЗРАБОТКА РОБАСТНЫХ АКУСТИЧЕСКИХ МОДЕЛЕЙ НА ОСНОВЕ  
ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

Автоматическое распознавание речи для русского языка, несмотря на наличие готовых продуктов, до сих пор является нерешённой задачей. Если для английской и китайской речи процент ошибок для распознавания составляет порядка 6% (на корпусе WSJ, Fisher) [160], без использования языковых моделей он возрастает до 10%, то для русской речи процент ошибок превышает 20% (при тестировании на объёмных данных, записанных в обычных условиях) [161], а без применения языковой модели применение такой системы может обладать низкой вероятностью результатов распознавания. Поэтому задача совершенствования технологии построения АМ, решение которой позволит обеспечить инвариантность к голосам дикторов и различным акустическим обстановкам, является актуальной.

Данная глава посвящена описанию технологии повышения робастности акустической модели в задаче распознавания речи с применением глубокого и машинного обучения. Предлагаемая технология основана на использовании информативных акустических признаков, извлечённых из иерархических нейросетевых моделей, а также на гибридных акустических моделях, обученных на основе машинного и глубокого обучения с применением дискриминативного подхода.

#### 4.1 Факторы, искажающие речевой сигнал в системах распознавания речи

Условия, в которых проходит эксплуатация систем автоматического распознавания речи, практически никогда не совпадают с условиями, в которых проходило обучение АМ. Следствием этого является то, что построенные модели не являются оптимальными для данных условий. Выделяют следующие факторы, искажающие речевой сигнал или обуславливающие его вариативность [22] (Рисунок 4.1).

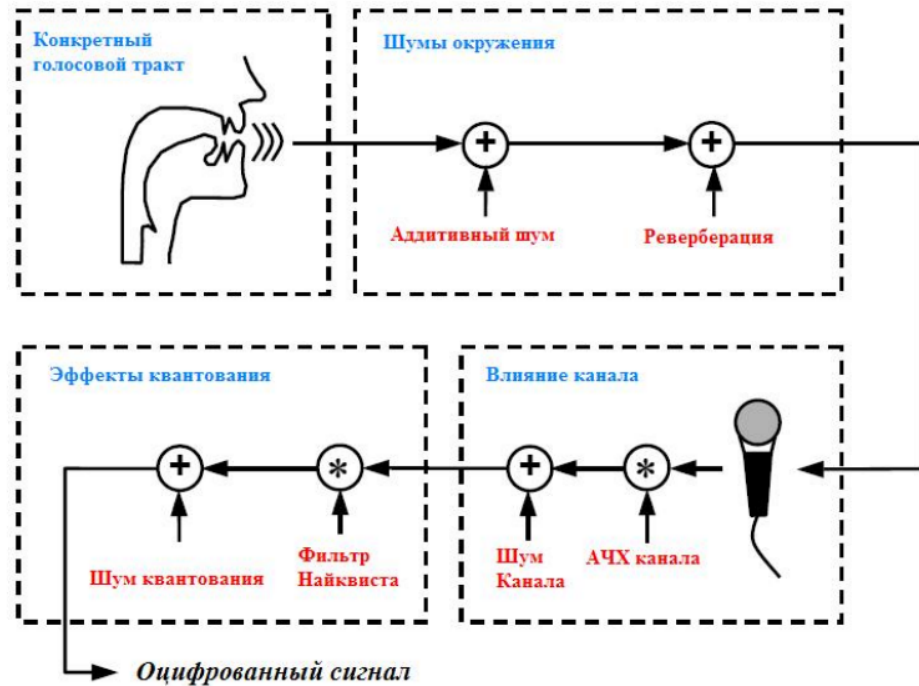


Рисунок 4.1 – Схема воздействия факторов, искажающих речевой сигнал

1. Голосовой тракт и манера произнесения. Этот фактор определяет вариативность сигнала. Как бы ни была велика обучающая выборка, всегда найдутся дикторы, отличающиеся по своим характеристикам от представленных в базе.

2. Аддитивный шум, всегда присутствующий в помещениях.

3. Реверберация (мультипликативный шум) – отражённый от стен основной сигнал.

4. Амплитудно-частотная характеристика микрофона и канала передачи.

5. Аддитивный шум канала передачи.

6. Преобразование сигнала фильтром Найквиста и шум квантования.

Стоит отметить, что сигнал, прошедший процедуру шумоподавления, не обеспечивает качество распознавания на высоком уровне. Это является следствием того, что шумоподавление предназначено для получения субъективных критериев качества и разборчивости звука, т. е. для того, чтобы трансформировать зашумлённые признаки к признакам чистого сигнала, из-за чего в признаках, извлекаемых из очищенного сигнала, может отсутствовать значимая информация для дальнейшего распознавания речи.

Для снижения влияния вышеуказанных факторов на качество АМ в данной работе использовались следующие подходы:

- 1) обучение акустических моделей при помощи зашумления обучающей выборки – аугментации, техника которой описана в пункте 2.3;
- 2) извлечение информативных акустических признаков, полученных с применением нейронных сетей.

Для повышения робастности акустических признаков в данной работе использовался подход *bottleneck*, позволяющий извлечь вектор информативных признаков из скрытых слоев глубокой сети, обладающий относительно небольшой размерностью.

#### 4.2 Технология повышения робастности акустической модели

Для повышения робастности АМ предлагается технология обучения, состоящая из таких основных стадий как:

- 1) Извлечение акустических признаков.
- 2) Обучение акустической модели GMM-HMM с использованием машинного обучения.
- 3) Обучение нейросетевой модели для извлечения информативных робастных акустических признаков.
- 4) Обучение нейросетевой модели для предсказания последовательности фонем.

В качестве акустических признаков используются MFCC, их первые и вторые производные, FBANK, а также PLP. Размерность вектора признаков для обучения модели GMM-HMM составляет 43 (40 MFCC, 3 PLP). Для обучения нейросетевых моделей MFCC-признаки и их производные заменялись 40 фильтрами FBANK.

##### 4.2.1 Разработка алгоритма обучения акустической модели с использованием машинного обучения

Поскольку, как правило, в банке речевых данных нет информации о временном нахождении каждой фонемы в речевой дорожке, то перед



использованием ANN необходимо иметь информацию о том, какая часть вектора соответствует какой фонеме, и эта процедура совершается при помощи GMM-HMM AM.

С помощью цепочки состояний HMM моделируют фонемы языка, которые, в свою очередь, объединяют в слова. Наиболее адекватной считается модель фонемы из трех состояний: начального, среднего и конечного. Также обычно выделяют отдельное состояние под тишину и неинформативные звуки, например, вдохи и выдохи; в качестве наблюдения рассматривается вектор акустических признаков, а для определения того, насколько хорошо определенное состояние HMM описывает текущий кадр речевого сигнала (вероятность эмиссии). При этом выходные вероятности моделируются с помощью GMM. В этом случае, плотность распределения вероятностей эмиссии задается в виде смеси гауссовских случайных векторов с диагональными ковариационными матрицами (Рисунок 4.2).

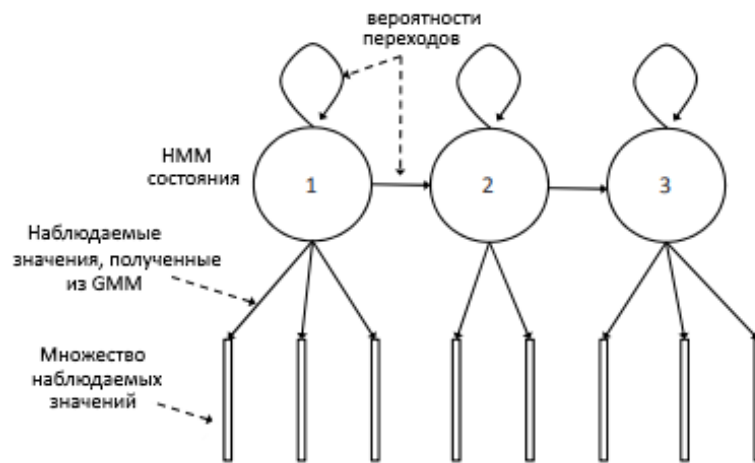


Рисунок 4.2 – Примерный вид GMM-HMM модели

Обучением GMM-HMM по критерию максимального правдоподобия (Maximum Likelihood, ML) называется подстройка параметров модели по заданной последовательности наблюдений таким образом, чтобы для модифицированной модели увеличить вероятность появления этой последовательности наблюдений. В случае неполноты данных обучение выполняется при помощи модифицированного алгоритма математического ожидания (Expectation-maximization, EM-алгоритм)

[26], посредством максимизации функции ожидания или вспомогательной функции логарифмического правдоподобия над недостающими переменными.

Обучение GMM-НММ АМ состоит из следующих этапов [162, 163].

1) Обучение монофонной модели (mono) на основе MFCC/FBANK и PLP. Это АМ, не содержащая никакой контекстной информации о предыдущей или последующей фонеме.

2) Обучение АМ для квифонов (qui1), представляющих вариант фонем в контексте четырёх других (двух слева и двух справа). Для этого обучается модель с использованием первых и вторых производных MFCC-признаков. Таким образом, вычисление производится для большего окна векторов признаков.

3) Уменьшение признакового пространства (qui2) с помощью LDA с применением линейного преобразования признаков, максимизирующего среднее правдоподобие (Maximum Likelihood Linear Transformation, MLLT). LDA, используя векторы признаков, строит состояния НММ, но с уменьшенным пространством признаков для всех данных и выводит уникальное преобразование для каждого диктора. MLLT же обеспечивает дикторонезависимость и робастность, минимизируя различия между дикторами.

4) Зачастую при оценке точности распознавания тишина (пауза) оценивается как фонема, что «ухудшает» модель, т. к. общее число фонем с тишиной может быть велико. Фиксация произношения позволяет больше учитывать фонемы с произношением, переопределяя вероятности произношения. Поэтому четвёртым этапом (qui2\_sp) является техника для определения вероятности тишины между словами (с применением обученной АМ) без учета просодической структуры, тем самым обуславливая вероятность тишины между словами для идентичности окружающих слов [164].

5) Применение адаптивного обучения диктора (Speaker Adaptive Training, SAT) совместно с применением линейной регрессии максимального правдоподобия пространства признаков fMLLR (qui3\_sp). SAT выполняет нормализацию дикторов и шумов путем адаптации к каждому конкретному диктору с определенным

преобразованием данных. fMLLR применяется для удаления идентичности дикторов из элементов матрицы путем оценки идентификатора диктора.

6) Обучение модели (sgmm\_qui3\_sp) SGMM. В SGMM параметры гауссовых смесей выводятся через подпространство низкоразмерной модели, которое фиксирует корреляции между состояниями квифонов и вариабельностью диктора, тем самым обеспечивая робастность акустической модели. В SGMM все состояния НММ используют одну и ту же структуру GMM, с таким же количеством гауссиан в каждом положении [33]. Модель определяется векторами, связанными с каждым состоянием  $N$ -размерности вместе с глобальным отображением из этого векторного пространства в пространство параметров GMM. Главным преимуществом SGMM является относительная компактность (число параметров, связанных с конкретными речевыми состояниями, довольно мало), что позволяет проводить обучение с меньшим количеством данных и позволяет использовать внеобластные и внеязыковые данные для обучения общих параметров.

7) Получение модели  $i$ -векторов. Идея метода  $i$ -векторов заключается в предположении, что существует линейная зависимость между дикторозависимыми математическими ожиданиями и дикторонезависимыми математическими ожиданиями. Таким образом,  $i$ -вектор представляет собой малоразмерный вектор, кодирующий отличие плотности распределения вероятностей акустических признаков, оцененной по фонограмме, от эталонной.  $i$ -вектор содержит канальную и дикторскую информацию. Метод адаптации DNN при помощи  $i$ -векторов, предложенный в работе [64], заключается в добавлении к вектору акустических признаков  $i$ -вектора, вычисленного по фрагменту фонограммы, соответствующему определенному диктору. Таким образом, осуществляется адаптация как к диктору, так и к акустической обстановке [65]. Для надежной оценки  $i$ -вектора необходимо наличие достаточного количества данных, приходящихся в среднем на одного диктора (десятки секунд). В этом случае обучение можно проводить стандартным образом. Если данных недостаточно, обучение можно проводить в два этапа. На первом этапе выполняется стандартное обучение дикторонезависимой нейронной сети. На втором этапе входной слой нейронной сети расширяется до размерности

вектора признаков, дополненного  $i$ -векторами. Соответствующие параметры инициализируются нулями, и выполняется дообучение нейронной сети по дикторозависимым признакам со штрафом на отклонение параметров от параметров дикторонезависимой сети и меньшей начальной скоростью обучения. В данной работе размерность признаков  $i$ -вектора составляет 100.

#### 4.2.2 Техника извлечения информативных акустических признаков

Извлечение информативных акустических признаков – извлечение дискриминативной характеристики аудио с использованием нейросетевой параметризации. Для этого разработана иерархическая мультимодульная ANN, MultiBN (Рисунок 4.3) на вход которой подаются 100-мерные вектора по 16 тыс. фреймов, данные вектора  $x$  получаются путём объединения FBANK-признаков и  $i$ -векторов при помощи линейного дискриминативного преобразования.

MultiBN – ансамбль из bottleneck нейросетей, обученных на основе дискриминативного критерия MPE. Каждое преобразование соответствует определенному участку в векторном пространстве. Каждый признак вектора преобразуется с помощью линейного преобразования, соответствующего участку, к которому принадлежит вектор.

$$MultiBN(x) = BN_3 \left( BN_2 \left( BN_1(x) \right) \right) = \sum_{i=1}^N A_i x_t, x \in \mathbb{R}^{n \times d}, \quad (4.1)$$

где  $BN_1, BN_2, BN_3$  – соответствующие BN-сети,  $A_i$  – линейное преобразование, соответствующее  $BN_i$  ANN,  $x_t$  – входные признаки, взятые с определённым временным контекстом  $t$ .

$$x = DBN(LDA(x_{FBANK}, x_{ive})), \quad (4.2)$$

где  $x_{FBANK}$  – входной вектор с FBANK-признаками;  $x_{ive}$  – входной вектор с  $i$ -векторами; LDA () – операция преобразования пространства входных векторов при помощи линейного дискриминантного анализа.

Каждая из bottleneck-нейросетей состоит из 3-х скрытых слоёв, по 2048 нейронов в каждом слое. На каждом уровне bottleneck-нейросети производится процедура получения трансформированных весовых коэффициентов (fine-tune),

которые и являются нашими информативными признаками, состоящая в следующем (Рисунок 4.3).

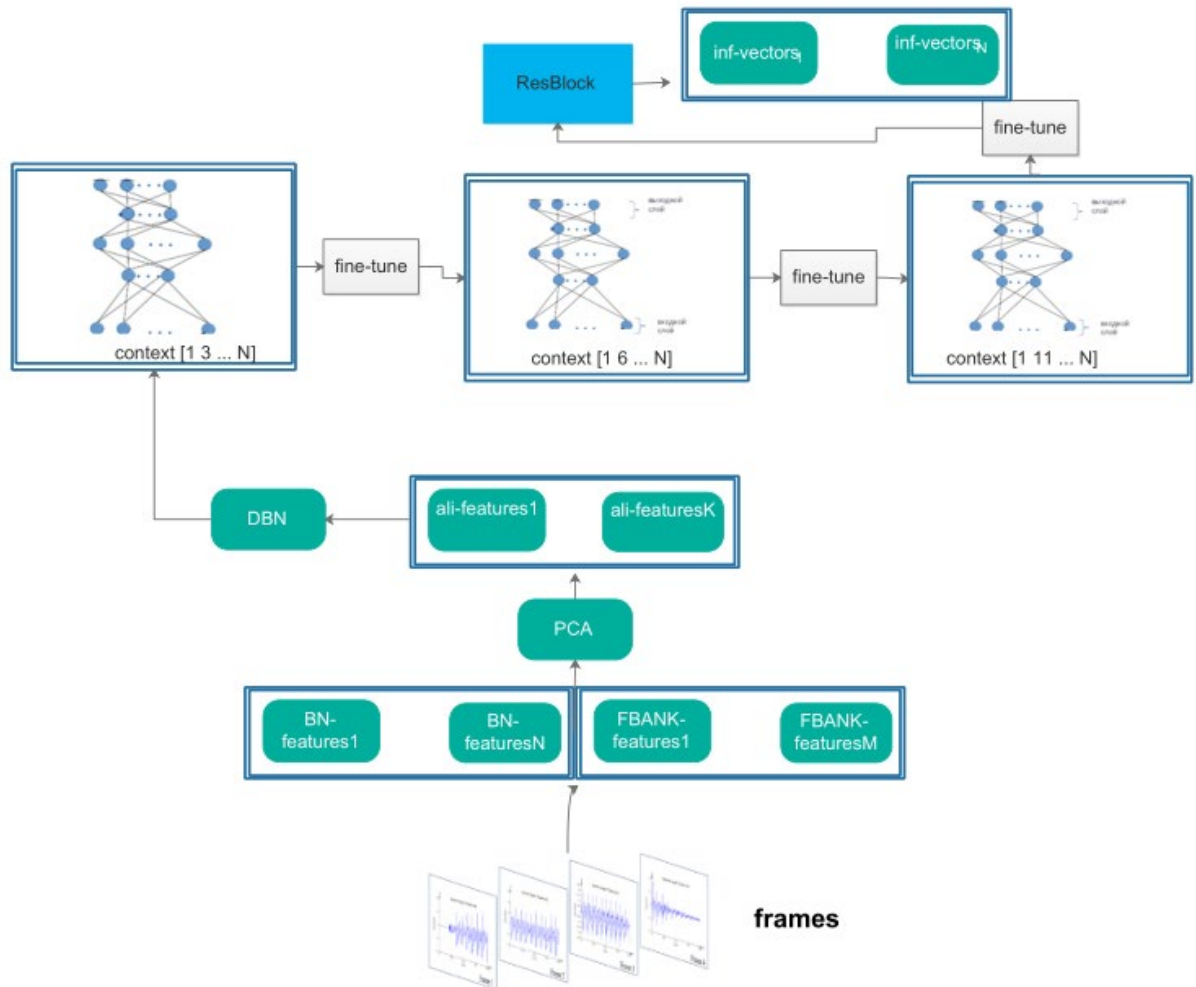


Рисунок 4.3 – Общая схема извлечения информативных признаков

1) Извлечение FBANK и PLP-признаков, а также  $i$ -векторов, относящихся к данным фреймам акустического сигнала.

2) Объединение извлечённых акустических признаков при помощи метода LDA, уменьшая размерность расширенного вектора акустических признаков до размерности 100 (4.2).

3) Обучается DBN. Для этого сначала обучается RBM, где в качестве наблюдаемых переменных выступают векторы признаков. Затем последовательно обучаются  $N-1$  RBM, в которых в качестве наблюдаемых переменных используются значения скрытых переменных предыдущей RBM. Далее каждый скрытый слой

DNN инициализируется значениями  $W$  и  $b$  соответствующей RBM. Наконец, добавляется инициализированный случайным образом выходной softmax-слой.

4) Обучение bottleneck происходит в 2 этапа [65, 165]:

- обучается ANN с матрицами весов  $\bar{W}_N$ , используя в качестве входных признаков  $x$ , с добавлением линейного слоя ANN (инициализируя нулями соответствующие весовые коэффициенты);
- ANN с матрицами весов  $W^N$  дополнительно обучается на расширенном векторе акустических признаков, уменьшая скорость обучения, а к целевой функции добавляется значение  $R(W)$  – штраф отклонения  $W^N$  от  $\bar{W}_N$  с величиной штрафа  $\lambda$ , определяемое следующим образом:

$$R(W) = \lambda \sum_{N=1}^{K+1} \sum_{i=1}^{M_N} \sum_{j=1}^{M_{N-1}} \left( W_{ij}^N - \bar{W}_{ij}^N \right)^2. \quad (4.3)$$

5) Инициализация bottleneck-слоя при помощи разбиения последнего скрытого слоя ANN на два слоя (первый слой – bottleneck-слой с матрицей весов  $W_{bn}^N$ ; второй слой – нелинейный слой с матрицей весов  $W_{out}^N$ ), используя сингулярное разложение матрицы весов [166]:

$$v^N = f(W^N v^{N-1} + b^N) \approx f(W_{out}^N (W_{bn}^N v^{N-1}) + b^N), \quad (4.4)$$

где

$$W^N \approx W_{out}^N W_{bn}^N. \quad (4.5)$$

6) Дополнительно проводится обучение полученной bottleneck с меньшей скоростью обучения и  $R(W)$ , а также отбрасывание слоев данной ANN (fine-tune), следующих за bottleneck-слоем.

Стоит отметить, что для first\_step bottleneck-features (формирование информативных акустических признаков первого уровня) используется контекст в 2 кадра; для second\_step bottleneck-features (формирование информативных акустических признаков второго уровня) используется контекст векторов в 5 кадров, а для third\_step bottleneck-features (формирование акустических признаков

третьего уровня) используется контекст в 10 кадров. После получения признаков из `third_step bottleneck-features` используется ResBlock (Рисунок 4.4).

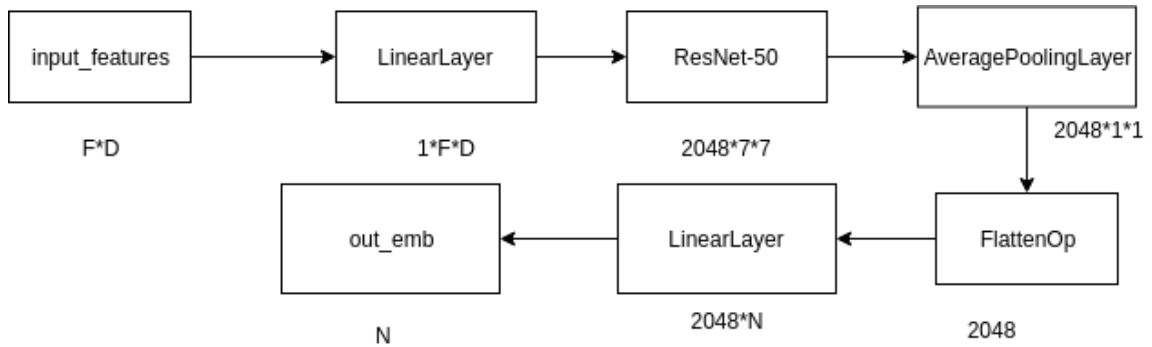


Рисунок 4.4 – Общая схема архитектуры ResBlock

На рисунке 4.4 следующие обозначения:  $F$  – размер входных признаков;  $D$  – количество фреймов,  $N$  – размер новых признаков; `input_features` – входные признаки; `LinearLayer` – линейные нейронные слои; `AveragePoolingLayer` – слой, трансформирующий данные при помощи свёртки  $1*1$ ; `FlattenOp` – операция трансформации в одномерное пространство; `out_emb` – выходные информативные признаки.

ResBlock представляет собой ANN, основанную на архитектуре ResNet-50 [167, 168] с двумя дополнительными линейными слоями. Архитектура ResNet-50 представляет собой CNN и состоит из 5 блоков (stage, Рисунок 4.5).

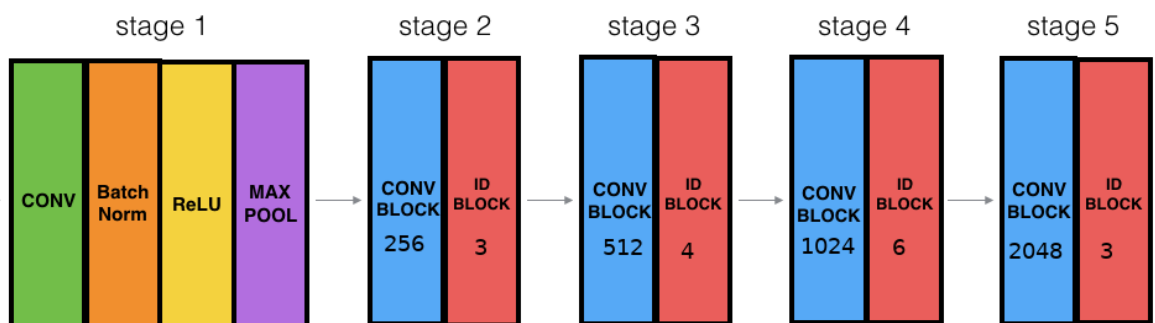


Рисунок 4.5 – Архитектура ResNet-50

CNN состоит из одной или нескольких пар сверток и слоев объединения. Слой свертки применяет набор фильтров на небольших локальных частях всего

входного пространства. Максимальный объединяющий слой берет максимальную активацию фильтра с разных позиций указанного окна. Результатом является версия уровня свертки с низким разрешением. Более высокие уровни работают на входах с низким разрешением и обрабатывают уже извлеченное высокоуровневое представление ввода [70]. На выходе слоя формируется карта признаков. Объединяющий слой выполняет понижение размерности входной карты признаков путем выбора максимального элемента и позволяет уменьшить влияние дикторской вариативности на параметры модели. Последние слои являются полностью связанными слоями, которые объединяют входы со всех позиций, чтобы классифицировать общие входы.

Поскольку CNN-архитектура предназначена для работы с трёхмерными последовательностями, как правило, изображениями, проведено предварительное преобразование входных данных. Входные акустические признаки представлены в виде двумерной матрицы  $U$  с размерностью  $N \times d$ , где  $N$  – общее количество фреймов,  $d$  – кол-во извлечённых фильтров. Преобразованные признаки представлены в виде трёхмерной матрицы размерности  $1 \times F \times D$  (количество каналов равно 1).

На этапе 1 после входного слоя используется двумерный свёрточный слой (CONV) с преобразованием матрицы пространства признаков из размера  $1 \times 1$  в  $3 \times 3$  (с применением техники дополнения, padding), использующий фильтры размером  $7 \times 7$ , и полносвязный слой (MAX POOL), использующий фильтры размером  $3 \times 3$ , величина шага свертки (stride) равна 2.

$$U_{pad} = (n_h - k_h + p_h + 1) \times (n_w - k_w + p_w + 1), \quad (4.6)$$

где  $n_h, n_w$  – размерность входных признаков,  $k_h, k_w$  – размерность фильтра,  $p_h, p_w$  – размерность дополнения.

После стадии 1 используется похожая топология стадий: на каждой стадии используются ID-блоки (или Residual-блоки, Рисунок 4.6), состоящие из 3 свёрточных слоёв (CONV BLOCK, первый свёрточный слой использует фильтры размером  $1 \times 1$ , второй – размером  $3 \times 3$ , а последний –  $1 \times 1$ ) между собой при помощи skip-связи. Размер ядер каждой из трёх свёрточных сетей на первой стадии



составляет 64, 64 и 128, на каждой стадии размер ядер удваивается относительно предыдущей стадии. Количество ID-блоков на второй стадии – 3, на третьей – 4, на четвёртой – 6, на пятой – 3:

$$\prod_{i,j} \begin{bmatrix} 1 \times 1, & 64 \cdot i \\ 3 \times 3, & 64 \cdot i \\ 1 \times 1, & 256 \cdot i \end{bmatrix} \times j; \quad i = 1,2,3,4; \quad j = 3,4,6,3. \quad (4.7)$$

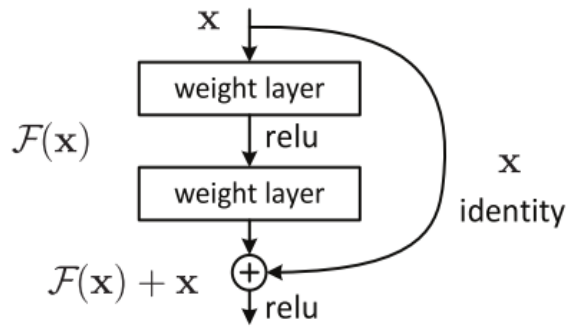


Рисунок 4.6 – Архитектура ID-block

Как видно из рисунка 4.6, выходной вектор ID-block описывается следующим образом:

$$y = F(x, \{W_i\}) + W_s x, \quad (4.8)$$

где  $x$  – входные признаки;  $W_s$  – набор весов относительно  $x$ ;  $y$  – целевая функция ID-блока относительно  $x$ ;  $F(x, \{W_i\})$  – ID-блок.

$$F(x, \{W_i\}) = W_2 \sigma(W_1 x), \quad (4.9)$$

где  $W_i$  – набор весов ID-блока для  $1 \leq i \leq n$ ,  $n$  – количество слоёв в ID-блоке;  $W_1$  и  $W_2$  – набор весов между соединёнными слоями;  $\sigma$  – функция активации слоя.

Таким образом, skip-связь позволяет модели изучать функцию идентичности, которая гарантирует, что верхний уровень слоя будет функционировать так же хорошо, как и нижний. При этом skip-связь не вводит дополнительных параметров, т. е. не усложняет вычислительный процесс, и используется до применения функции активации. Данная архитектура позволяет частично решить проблему исчезающих градиентов, быстрее сходится, обладает большей точностью.

Для обучения нейросетевой модели извлечения информативных признаков использованы следующие параметры:

- размер входных признаков – 100;

- размер выходных информативных признаков: 100;
- метод оптимизации – AdamBound;
- функция активации – ReLU.
- коэффициент регуляризации – 0,01.
- коэффициент обучения – 0,001.

При обучении обновление параметров нейросети происходило следующим образом:

$$\Delta W_{CE} = W_{CE} - \varepsilon \left( \frac{\partial loss_{CE}(x)}{\partial W_{CE}} \right), \quad (4.10)$$

$$\Delta W_{MSE} = W_{MSE} - \varepsilon \left( \frac{\partial loss_{MSE}(x)}{\partial W_{MSE}} \right), \quad (4.11)$$

$$\Delta W = W - \varepsilon \left( \frac{\partial loss_{CE}(x)}{\partial W_{CE}} - \lambda \frac{\partial loss_{MSE}(x)}{\partial W_{MSE}} \right), \quad (4.12)$$

где  $x$  – вектор входных данных,  $\varepsilon$  – коэффициент обучения;  $\lambda$  – коэффициент регуляризации,  $loss_{CE}$  – функция потерь перекрёстной энтропии;  $loss_{MSE}$  – функция потерь среднеквадратичного отклонения.

$$loss_{CE}(x) = - \sum_{c=1}^M y \log(P(x|c)), \quad (4.13)$$

где  $M$  – число уникальных классов.

$$loss_{MCE}(x) = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i), \quad (4.14)$$

где  $n$  – общее количество данных;  $\bar{y}_i$  – предсказанный класс.

На рисунке 4.7 показаны результаты тестирования разработанной нейросетевой модели.

Для обучения модели использовалась выборка, составляющая 5% от речевого корпуса, описанного в пункте 2.1, количество классов  $M$  составило 65.

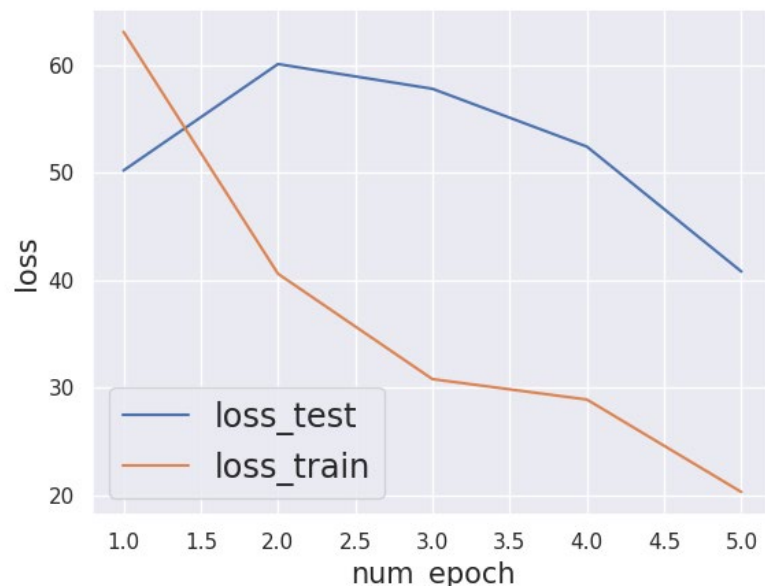


Рисунок 4.7 – Зависимость функции  $loss_{CE}$  на тестовой и обучающей выборках от количества эпох обучения

Объединенные при помощи LDA признаки – признаки, извлечённые из ResBlock, и FBANK, поступают на вход сети, предсказывающей последовательность фонем.

#### 4.3 Разработка нейросетевой модели для предсказания последовательности фонем

Применение глубокого обучения для распознавания речи способствует оптимальной адаптации акустических признаков как под дикторов, так и под окружение. Модель, применяющаяся в данной работе, использует две архитектуры – нейронную сеть с временной задержкой TDNN [169] и двунаправленную долгую кратковременную нейронную сеть (Bidirectional Long Short Memory, BLSTM) [170] со слоем внимания (attention).

TDNN представляет собой многоуровневую архитектуру искусственной нейронной сети, целью которой является классификация шаблонов с неизменностью сдвига и получение контекста на каждом уровне сети (Рисунок 4.8).

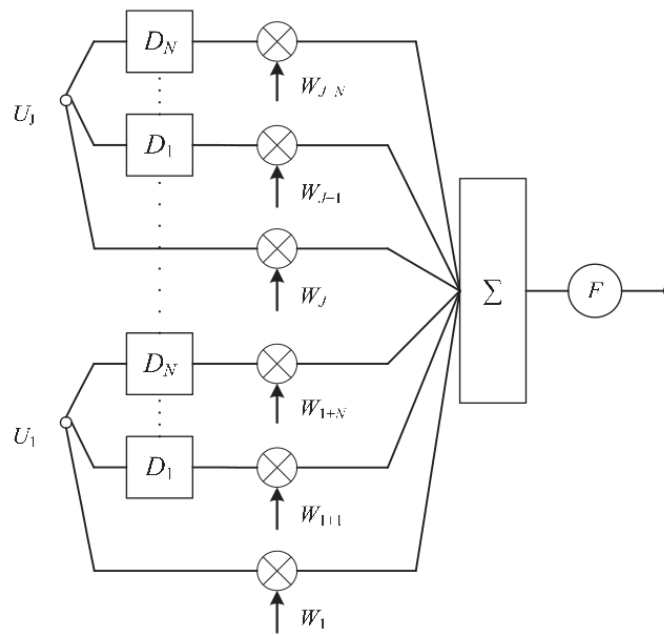


Рисунок 4.8 – Структура TDNN

На приведенном рисунке 4.8  $U_1, \dots, U_J$  – входы узла; каждый из  $J$  входов умножается на соответствующий весовой коэффициент  $W$ ;  $D_1, \dots, D_n$  – временные задержки,  $F$  – активационная функция. Таким образом, в TDNN встраивается кратковременная память, а временная задержка позволяет сделать TDNN инвариантной ко временным сдвигам.

В данной работе на вход сети TDNN подается последовательность векторов признаков, полученная на фреймах речевого сигнала длительностью не более 16 секунд (Рисунок 4.9). В качестве входных признаков используются объединённые признаки, полученные на основе FBANK-признаков и высокоуровневых признаков, а также их временные границы.

TDNN не требует явной сегментации перед классификацией. Таким образом, для классификации временного шаблона (такого как речь) данная архитектура имеет незначительную зависимость от границ фонем перед их классификацией. Для контекстного моделирования в TDNN каждый элемент в первом скрытом слое получает входные данные из коэффициентов 4-х фреймового окна, на втором скрытом слое каждый из 3-х элементов TDNN просматривает 6-ти фреймовое окно уровней активности в 1-ом скрытом слое, на третьем скрытом слое просматривается 10-ти фреймовое окно. Таким образом, элемент TDNN имеет

возможность кодировать временные отношения внутри диапазона в N-задержек. Более высокие уровни могут работать в больших временных промежутках.

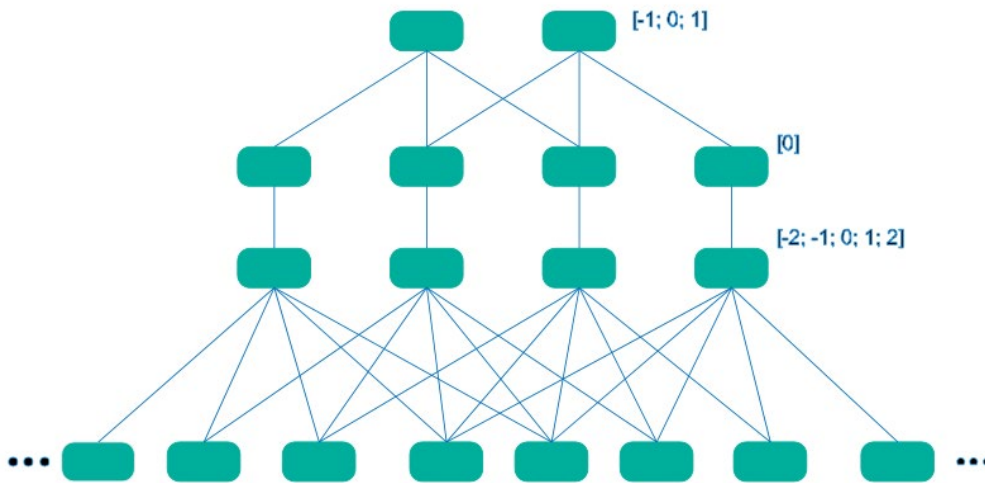


Рисунок 4.9 – Схема архитектуры TDNN, используемой в данной работе

LSTM – рекуррентная сеть, хранящая информацию о своих предыдущих состояниях и учитывающая ее при прогнозировании. В этой архитектуре частично решена проблема затухания градиента, возникающая вследствие использования алгоритма обратного распространения ошибки, когда величина градиента постепенно уменьшается в рекуррентных слоях. Архитектура рекуррентной сети LSTM, описанная в пункте 1.3.3, позволяет определить гибкие долгосрочные зависимости от данных, что особенно важно в контексте человеческой речи. Однако однонаправленные LSTM имеют ограничения: слои этих сетей имеют доступ к прошедшему контексту, и не имеют доступа к следующему контексту. Для этого и используется двунаправленная долгая кратковременная нейронная сеть (Bidirectional Long Short Memory, BLSTM). В подобной архитектуре два разных внутренних слоя оперируют с данными в двух направлениях (вперёд и назад). Оба этих слоя соединены с одним выходным слоем, что позволяет использовать контекст из двух направлений. Недостатком подобной архитектуры по сравнению с однонаправленной является большее время обучения.

Каждая из используемых для распознавания фонем сетей (TDNN и BLSTM) имеет 5 слоёв с 2048 нейронами (1 входной, 1 выходной и 3 скрытых слоя,

Рисунок 4.10). Данное количество нейронов выявлено экспериментально в работе [41].

$$F(x) = TDNN(BLSTM(\text{attn}(\text{softmax}(x)))) = \bar{y}, \quad (4.15)$$

где  $TDNN(x)$  – преобразование признаков  $x$  при помощи TDNN;  $BLSTM(x)$  – преобразование при помощи BLSTM;  $\text{attn}(x)$  – преобразование признаков при помощи техники внимания;  $\text{softmax}(x)$  – классификация признаков;  $\bar{y}$  – набор фонем,  $x = LDA(x_{FBANK}, x_{ResBlock})$ .

Механизм внимания в рекуррентной нейронной сети – это способ увеличить важность одних данных по сравнению с другими. Существует две модели внимания: «мягкая» (soft) и «жесткая» (hard). В первом случае сеть все равно обратится ко всем данным, к которым имеет доступ, но веса этих данных будут разными. Во втором случае из всех существующих данных сеть обратится лишь к некоторым, а у остальных будут нулевые веса. В данной работе использовалась «мягкая» модель внимания.

Механизм внимания состоит в следующем (Рисунок 4.11). Выделяется множество векторов  $\{v_i\}_{i=1}^K$ , над которыми будет осуществляться внимание. При этом генерируется ключ  $k$ , определяющий какие вектора из множества  $\{v_i\}_{i=1}^K$  будут использованы. В данное множество также вносится информация о временной позиции. Используя  $k$ , каждому вектору  $v_i$  сопоставляется вес  $w_i$ , который может быть интерпретирован как вероятность того, что внимание должно быть сконцентрировано именно на объекте  $v_i$ :

$$w_i = \frac{\exp(\text{dist}(k, h_j))}{\sum_{i=1}^K \exp(\text{dist}(k, h_i))} = \frac{\exp(k \cdot q_i)}{Z}, j = 0 \dots N,$$

где  $N$  – количество примеров в выборке;  $\text{dist}(k, h)$  – мера близости, обычно используют косинусное сходство:

$$\text{dist}(k, h) = \frac{k^T h}{\|k\| \cdot \|h\|}.$$

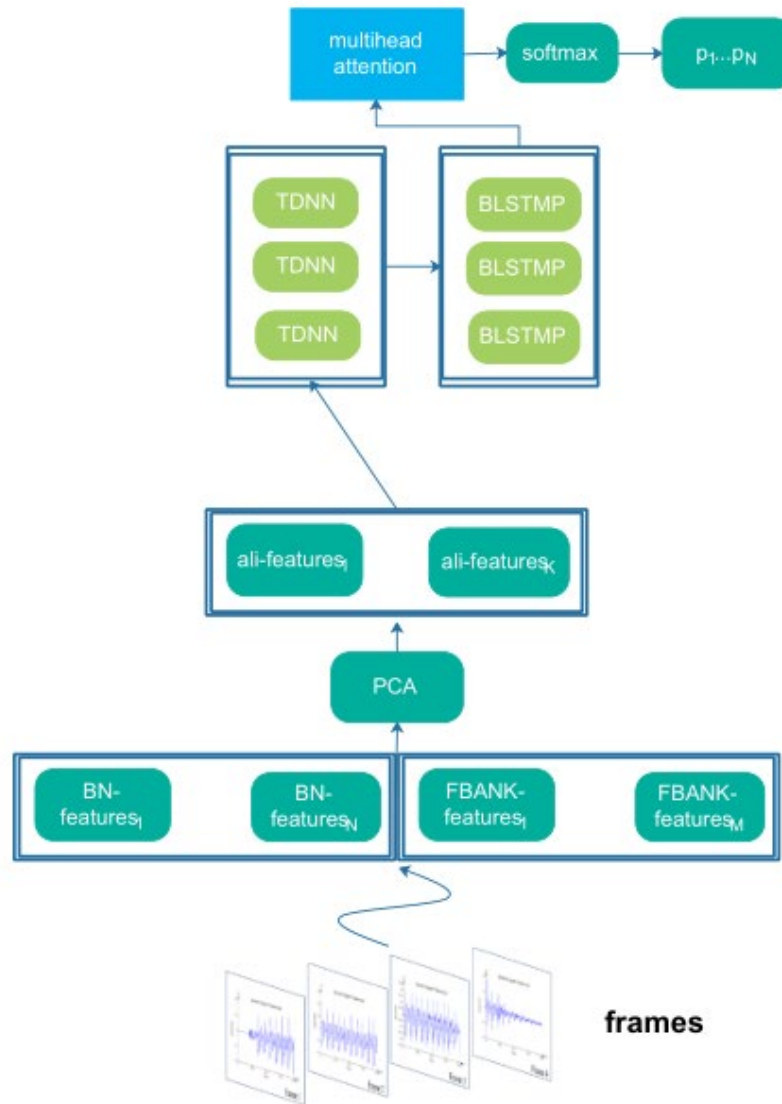


Рисунок 4.10 – Архитектура TDNN-BLSTM-attention

Т.е. внимание фокусируется в основном на объектах схожих с ключами. На следующем шаге вычисляется контекстный вектор  $c$ :

$$c = \sum_{i=1}^K w_i v_i.$$

Основываясь на значении вектора  $c$ , может быть предсказан класс фонемы, используя softmax-функцию:

$$y_i = \text{softmax}(c), i=0 \dots N.$$

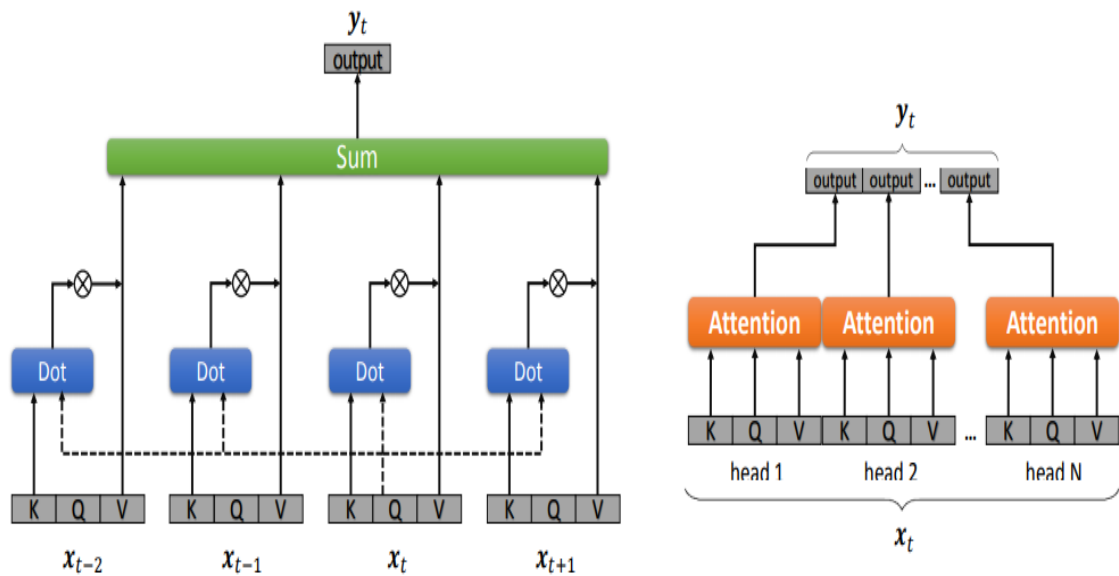


Рисунок 4.11 – Схема работы слоя внимания

Обучение вышеописанной модели проведено с использованием критерия минимизации взаимной энтропии; в качестве метода оптимизации процедуры градиентного спуска использован AdamBound; в качестве функции активации слоёв – ReLU, функция loss и обновление параметров проводилось согласно с (4.13).

#### 4.4 Численные исследования эффективности использования предложенной мультимодульной архитектуры акустической модели

Ниже описаны эксперименты, проведённые для оценки эффективности работы предложенного метода извлечения информативных робастных акустических признаков и предсказания на их основе последовательности слов. В качестве обучающего материала для формирования АМ использованы два речевых корпуса – VoxForge [105] и SpokenCorpora; общей продолжительностью около 20 часов, из которых предварительно удалены неконтекстные речевые данные.

Языковая модель обучена на основе триграмм. Словарь сформирован из 500 тыс. наиболее встречаемых слов, извлечённых из языковой модели. Для формирования транскрипции использована система, описанная в главе 3. Обучающая и тестовая выборки с аудиоданными разделены в отношении 95/5.



Результаты распознавания с использованием различных АМ приведены в таблице 4.1, где кол-во НММ – количество состояний для скрытой марковской модели, кол-во GMM – количество гауссиан, WER – метрика, используемая для измерения качества распознавания; mono – монофонная АМ; qui1 – квифонная АМ, обученная с использованием дельта и дельта-дельта акустических признаков; qui2 – квифонная АМ, с применением LDA и MLLT; qui2\_sp – к модели qui2 применена техника переопределения вероятности тишины; qui3\_sp – квифонная АМ, с применением SAT и fMLLR; sgmm\_qui3\_sp – АМ, с применением SGMM; nn\_stbn\_sgmm – АМ, с применением TDNN-BLSTM-attention и стандартных bottleneck-признаков, объединённых с FBANK на основе тандемного метода [9]; nn\_resblock\_sgmm – АМ, с применением TDNN-BLSTM-attention и объединения акустических признаков FBANK и признаков, извлечённых из ResBlock.

Система Google Cloud Speech API протестирована на тех же данных, ошибка распознавания составила 9,33%.

Таблица 4.1 – Результаты распознавания с использованием различных акустических моделей

№ модели	Модель	WER, %	Кол-во GMM	Кол-во НММ
1	Mono	64,01	4000	1500
2	qui1	35,76	20000	2500
3	qui2	21,55	50000	4000
4	qui2_sp	19,98	50000	4000
5	qui3_sp	14,07	100000	5000
6	sgmm_qui3_sp	10,81	120000	8500
7	nn_stbn_sgmm	7,93	-	-
<b>8</b>	<b>nn_resblock_sgmm</b>	<b>5,2</b>	-	-

Из таблицы 4.1 видно, что для обучения АМ на относительно небольших речевых данных (продолжительностью около 20 часов) применение комбинации акустических признаков FBANK и PLP показывает результат лучше (модели 6–8), чем MFCC (модели 1–5). Использование комбинации признаков FBANK и PLP, подаваемых на вход нейронной сети, в ряде случаев уменьшает WER, т.к. MFCC направлены на применение для алгоритмов машинного обучения, в то время как FBANK и PLP отображают более естественные признаки речи.

Применяемая процедура обучения АМ позволяет добиться высокой робастности и дикторонезависимости, что видно из таблицы 4.1. Использование предложенной модели нейросетевой параметризации речевого сигнала позволяет повысить точность распознавания на 2.7% по сравнению с моделью, извлекающей стандартные bottleneck-признаки (модель 7), а разработанные АМ и архитектура нейросети для распознавания фонем показывают результат распознавания на 4.1% лучше, чем ASR от Google, которая обучалась на речевых базах объемом в десятки тысяч часов.

#### 4.5 Выводы к главе 4

1. Предложена технология повышения робастности GMM-HMM АМ, использующая LDA с применением MLLT для уменьшения признакового пространства, метода SAT и fMLLR для адаптации под диктора, модели SGMM для улучшения качества распознавания, добавления  $i$ -векторов для адаптации как к диктору, так и к акустической обстановке.

2. Впервые предложена модель нейросетевой параметризации речевого сигнала, основанная на объединении иерархической мультимодульной архитектуры MultiBN, состоящей из трёх связанных нейронных сетей с узким горлом (с контекстами в 2, 5 и 10 кадров), и блока ResBlock на основе архитектуры ResNet-50, что позволяет повысить точность распознавания на 2,7% по сравнению с моделью, извлекающей стандартные bottleneck-признаки.

3. Получили дальнейшее развитие методы нейросетевой классификации фонем за счет использования механизма внимания в последнем скрытом слое сети, включающей в себя ANN с временными задержками и двунаправленную нейросеть с долгой кратковременной памятью. Как показали численные исследования, предложенная архитектура сети для классификации фонем сохраняет высокую точность на относительно небольшом обучающем наборе аудиоданных, в отличие от end-to-end систем, для обучения которых требуется речевая база длительностью в десятки тысяч часов.

## ГЛАВА 5

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ  
РУССКОЙ СЛИТНОЙ РЕЧИ

В данной главе представлена структура разработанной системы автоматического распознавания слитной русской речи. Рассмотрены основные её структурные блоки. Описан процесс получения языковой и акустической модели, процесс декодирования. Охарактеризованы обучающие и тестовые корпуса текстовых и речевых данных. Проведено сравнение эффективности работы разработанной системы с существующими решениями.

## 5.1 Структура системы распознавания речи

Разработанная автоматическая система распознавания речи работает в двух режимах (режиме обучения и режиме распознавания) и состоит из трёх основных блоков:

- блока обучения ЯМ и транскрипционной модели, в котором реализован сбор и предварительная обработка данных, обучение ЯМ и генерация словаря транскрипций;
- блока обучения АМ, результатом работы которого является обученная АМ на основе информативных робастных акустических признаков, а также  $i$ -векторов, технология извлечения данных признаков описана в главе 4. В данном блоке реализованы следующие этапы:
  - выделение акустических признаков (MFCC, FBANK и PLP).
  - обучение АМ на основе GMM-HMM подхода с применением дискриминативных методов обучения.
  - модификация ЯМ с использованием полученной АМ и ЯМ, а также алгоритмом *rescoring lattices* [171];
- блока распознавания, в котором происходит запись и предобработка сигнала, а также его дальнейшее преобразование в текстовый вид.

Программные средства, входящие в состав блока обучения, реализованы автором с использованием языков программирования C++, Perl, Python, Cython, Bash, а также библиотек OpenFST (библиотека для конструирования, комбинирования и поиска взвешенных конечных преобразователей для представления вероятностных моделей, в частности, n-грамм) [172]; Kaldi (для извлечения признаков и распознавания речи) [173]; Tensorflow (для построения ANN-моделей) [174]. Эти современные средства разработки позволили реализовать нейросетевые модели, методы и алгоритмы, представленные в диссертации.

На основе обученных ЯМ и АМ в режиме распознавания производится процесс декодирования (поиск наиболее правдоподобной последовательности слов, соответствующей последовательности векторов высокоуровневых признаков для данной фонограммы). При декодировании кроме АМ и ЯМ используется словарь транскрипций и транскрипционная модель. Выдаваемая в процессе декодирования последовательность слов (результат распознавания) записывается в выходной текстовый файл.

Опишем функционирование системы в двух режимах более подробно.

#### 5.1.1 Функционирование системы автоматического распознавания русской речи в режиме обучения

В режиме обучения (Рисунок 5.1) осуществляется: сбор и обработка речевых и текстовых данных; формирование словаря транскрипций и обучение моделей генерации транскрипций; обучение АМ и ЯМ.

Первыми этапами в приведенной схеме обучения является сбор данных из Сети. После осуществляется сбор речевых, текстовых и транскрипционных данных. Затем проводится их нормализация.

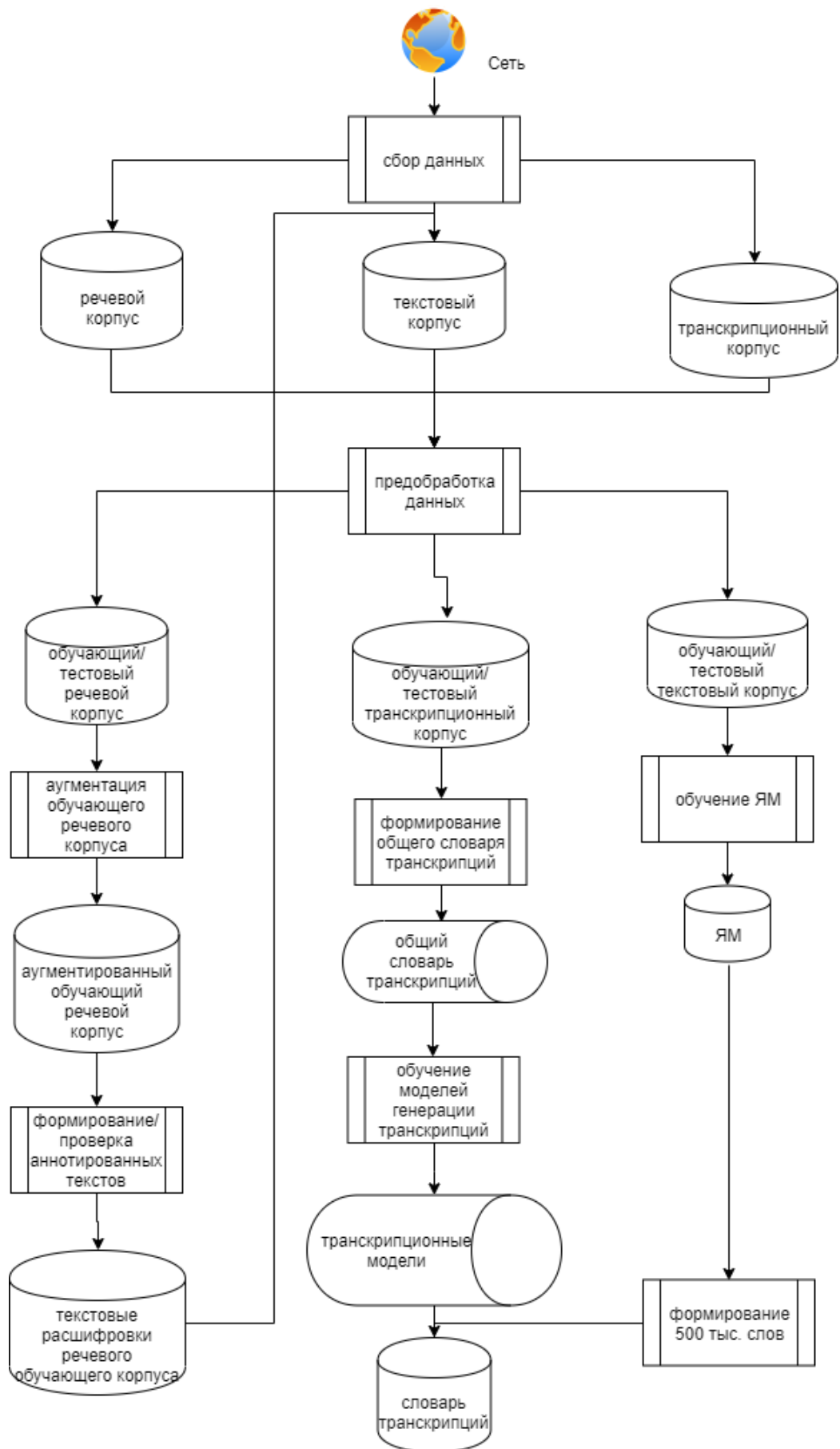


Рисунок 5.1 – Схема функционирования системы распознавания в режиме обучения

На рисунке 5.2 представлена схема процесса предобработки речевых данных.

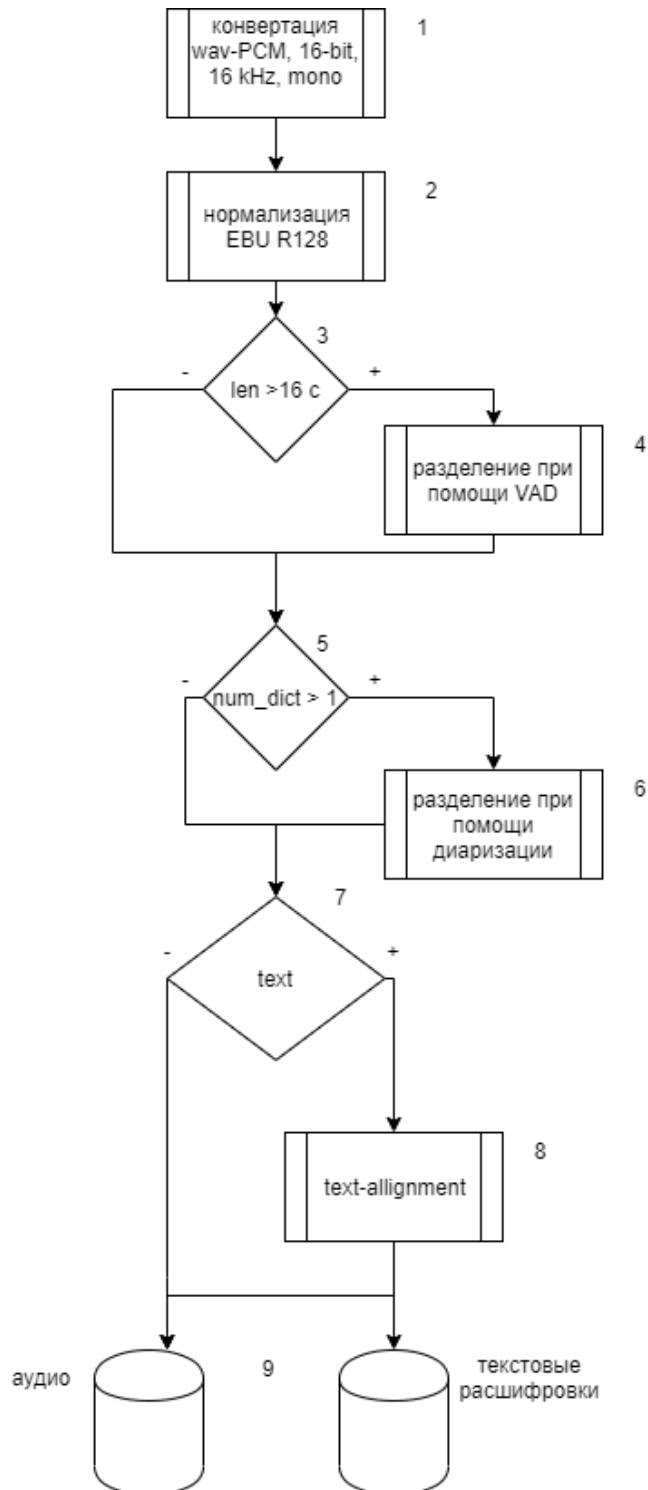


Рисунок 5.2 – Схема предобработки речевых данных

Предобработка речевых данных состоит из следующих этапов.

1. Конвертация файлов в формат wav-PCM, 16-bit, 16 kHz, mono.

2. Нормализация аудиофайлов согласно стандарту EBU R128 [175].
3. Проверка речевых файлов на длительность: файлы длиной более 16 сек разделяются на отдельные аудиофайлы длиной не более 16 сек.
4. В случае если продолжительность аудио более 16 сек используется подмодуль VAD для разделения аудио на массив данных, используя значение времени тишины  $t_{sil} \leq 16$  сек.
5. Проводится проверка на количество дикторов (`num_dict`), используя модель диаризации [4], в текущем аудио, если более одного диктора, то переходим к следующему шагу, иначе – переходим к шагу 7.
6. Используя информацию о времени смены диктора и о метке диктора, аудио разделяется на несколько речевых отрезков.
7. Проводится проверка на наличие текстовых расшифровок к данному речевому отрезку аудио, если таковых нет, то переходим к следующему шагу, иначе – к шагу 9.
8. Используя автоматическую систему распознавания речи, генерируется текст к аудио. В случае, если для изначального аудио имелась текстовая расшифровка, то используется модифицированный алгоритм Смита-Уотермана, описанный в пункте 2.2.

После вышеописанных действий проводится аугментация речевых данных, описанная в пункте 2.3, с целью повышения робастности АМ. В результате формируется речевой обучающий корпус. Процесс получения транскрипционных моделей описан в главе 3.

Схема обучения ЯМ приведена на рисунке 5.3. В данной работе использовалась модель, содержащая LSTM-блоки.

В простых DNN входной слой сети представляет собой множество, состоящее из  $n-1$  слов, предшествующих данному слову. Каждое слово из словаря ассоциировано с вектором длиной  $V$  (размер словаря), где только одно значение, соответствующее индексу данного слова в словаре, равно 1, а все остальные значения равны 0. Слой, сформированный путем объединения векторов слов, называется проекционным слоем. Основным недостатком таких сетей является то,

что для предсказания слова они используют предшествующий контекст определенной длины.

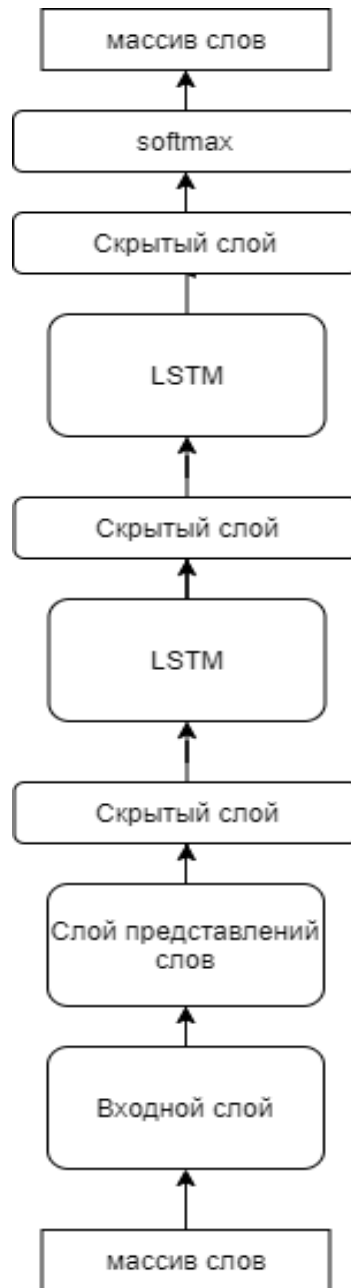


Рисунок 5.3 – Архитектура ANN для языковой модели

В рекуррентных сетях [88] скрытый слой хранит всю предыдущую историю, поэтому размер контекста неограничен. Использование LSTM для построения ЯМ базируется на следующих особенностях [89]: входной вектор кодируется в виде 1-of-N слов; функция softmax, используемая в выходном слое для получения нормированных вероятностей; кросс-энтропия используется в качестве критерия



обучения; нормализация входного вектора, которая обычно рекомендуется для нейронных сетей, не требуется из-за входного кодирования 1-of-N слов.

После сбора текстовых данных проводится их нормализация, подробно описанная в главе 2. К нормализованному текстовому корпусу добавляются нормализованные текстовые расшифровки аудиофайлов. На основе этих данных обучаем ЯМ (архитектура нейросети для ЯМ изображена на рисунке 5.4) со следующими параметрами: количество скрытых слоёв – 5; количество нейронов в скрытых слоях – 256; тип активации – sigmoid; коэффициент обучения – 0.0001; оптимизация градиентного спуска – RMSprop; количество эпох – 5. Дополнительно обучается 5-gram ЯМ для отбора текстов для обучения по следующему критерию: если в тексте содержался 1% 5-gram от общего значения, то данный текст не применялся для обучения. Для упрощения n-gram ЯМ использовалась техника n-gram pruning – из модели удалены n-gram, встречающиеся менее 4 раз. Используя данную ЯМ, извлечены 500 тыс. наиболее встречаемых слов, формируя список слов для словаря. Для слов из данного списка сгенерированы парадигмы при помощи `rumorphy2` и удалены повторы слов. Для формирования словаря транскрипций к полученному списку слов применена ANN-модель для автоматической генерации транскрипций.

Для оценки качества ЯМ используют вероятность, которую модель назначает тестовым данным; SE и коэффициент неопределенности (перплексия, *perplexity*).

Для тестовых данных  $T$ , состоящих из предложений  $(t_1, t_2, \dots, t_{l_T})$ , содержащих суммарно  $W_T$  слов, вероятность определяется как произведение вероятностей для каждого из предложений:

$$P(t) = \prod_{k=1}^{l_t} P(t_k).$$

SE определяется следующим образом:

$$H(T) = -\frac{1}{W_T} \log_2 P(T)$$

и может интерпретироваться как среднее количество бит информации, необходимое для кодирования каждого слова в тестовых данных при помощи алгоритма сжатия, связанного с моделью [86].

Перплексия (*PPL*) определяется следующим образом:

$$PPL(T) = 2^{H(T)} = P(T)^{\frac{1}{WT}}$$

Чем меньше эта величина, тем лучше модель предсказывает появление слов в документах текстового корпуса.

Между перплексией и количеством неправильно распознанных слов существует сильная корреляция [87]: чем меньше взаимная энтропия и перплексия, тем лучше модель соответствует тестовым данным.

Для качественной модели величина перплексии имеет значение не менее 700 (зависит от сложности тематики обучаемой выборки) [176]. Для ЯМ, полученной в ходе выполнения данного диссертационного исследования, на текстовом корпусе, сформированном после сбора текстовых данных и их нормализации, значение величины перплексии составило 368.73. Общее описание текстовых данных, на которых обучалась ЯМ, приведено в таблице 2.2 второй главы.

Для обучения АМ извлекаются информативные робастные и дикторонезависимые акустические признаки, а также формируются модели *i*-векторов. Схема обучения АМ представлена на рисунке 5.4.

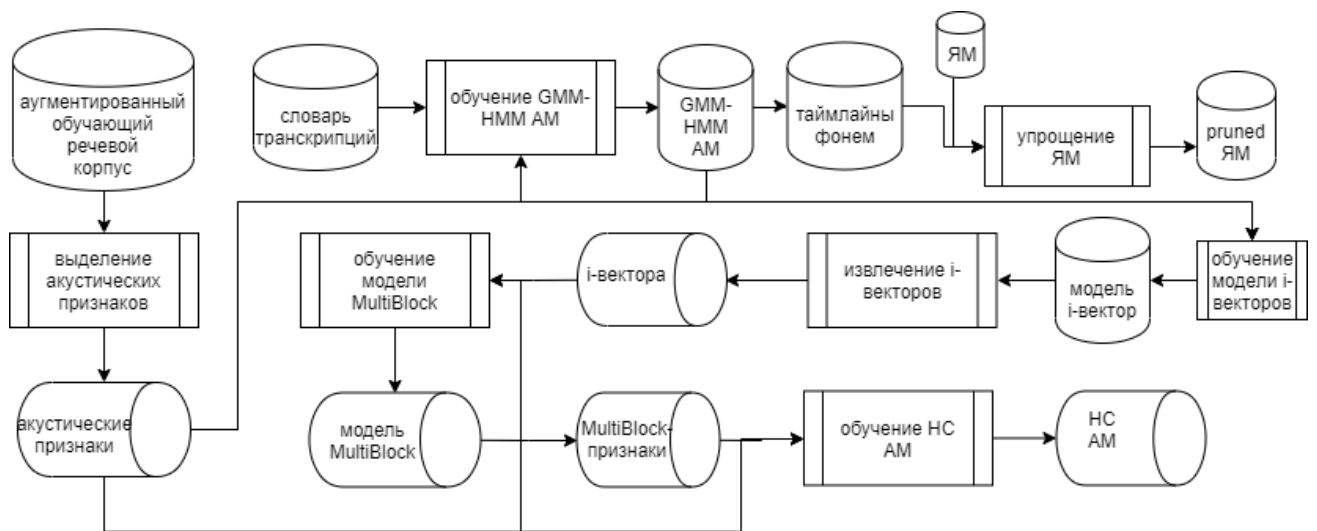


Рисунок 5.4 – Схема обучения акустической модели

Процесс обучения АМ состоит из следующих этапов.

1. Аугментация речевых сигналов с целью повышения робастности системы распознавания речи.
2. Выделение акустических признаков (MFCC, FBANK и PLP).
3. Обучение АМ на основе GMM-HMM подхода с применением дискриминативных методов обучения. По выделенным речевым сегментам происходит вычисление признаков PLP, FBANK,  $i$ -векторов. Построенный на каждом кадре вектор признаков и  $i$ -вектор, соответствующий участку фонограммы, которому принадлежит рассматриваемый кадр, объединяются в единый вектор признаков. По объединенным векторам признаков вычисляются вектора робастных признаков при помощи модулей MultiBlock и ResBlock, описанных в главе 4.
4. Полученные информативные признаки поступают на вход ANN, архитектура которой основана на TDNN и LSTM, а также использует слой с вниманием.

Языковые модели хранятся в формате arpa [177], а также в формате Tensorflow weights, models [178]. Акустическая модель хранится в формате arc [179]. Этот формат используется библиотекой OpenFST и предназначен для хранения информации о фонемном представлении, а также о весовых коэффициентах модели. Помимо этого, модели хранятся в виде fst-графов (L.fst, G.fst, H.fst, C.fst), а также объединения графов (CLG.fst). Общая топология модели, построенной для распознавания речи на основе отдельных fst-графов, хранится в mdl-формате для дальнейшего декодирования [173]. Входные аудиоданные хранятся в формате wav.scp.

Файлы с текстовыми расшифровками, а также словари имеют текстовый формат. Кроме того, все входные данные хранятся в ark-формате [173].

Часть нейросетевых моделей хранится в формате Tensorflow weights models.

После того, как получены результаты в режиме обучения, система готова к распознаванию речи.

## 5.1.2 Функционирование системы автоматического распознавания русской речи в режиме распознавания

Схема работы системы распознавания речи в режиме распознавания представлена на рисунке 5.5.

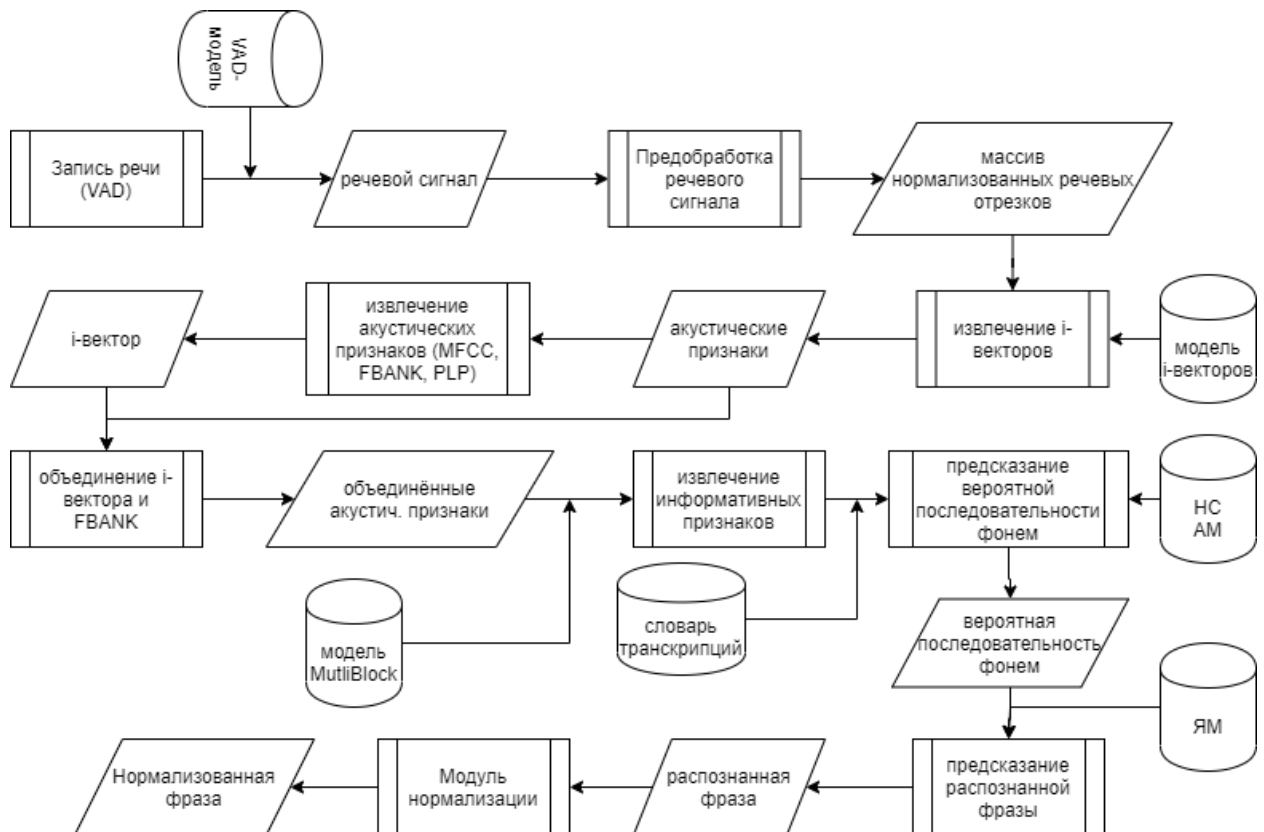


Рисунок 5.5 – Схема автоматической системы распознавания речи в режиме распознавания

Этапы работы системы в режиме распознавания следующие:

1. Аудиозапись речевого сигнала с частотой дискретизации 16000 Гц, 16 бит на отсчет, формат wav-PCM.
2. Выделение акустических признаков FBANK.
3. Выделение речевых сегментов при помощи детектора активности диктора. Для каждого сигнала затем происходит нормализация согласно формату EBU R128.

4. По выделенным речевым сегментам происходит вычисление признаков PLP, FBANK,  $i$ -векторов. Построенный на каждом кадре вектор признаков и  $i$ -вектор, соответствующий участку фонограммы, которому принадлежит рассматриваемый кадр, объединяются в единый вектор признаков.

5. По объединенным векторам признаков вычисляются вектора робастных признаков при помощи модулей MultiBlock и ResBlock, описанных в главе 4.

После вышеописанных действий производится процесс декодирования – поиск наиболее правдоподобной последовательности слов, соответствующей последовательности векторов высокоуровневых признаков для данной фонограммы. При декодировании используются акустическая модель, языковая модель и словарь транскрипций, полученные в результате работы системы в режиме обучения. Выдаваемая в процессе декодирования последовательность слов (результат распознавания) записывается в выходной текстовый файл.

Построенные АМ, ЯМ и сгенерированный словарь транскрипций достаточно объемны, поскольку они содержат лексикон, фонетическое дерево решений, топологию фонем НММ, поэтому в данной работе использовался подход к статическому декодированию, основанный на конечных автоматах (Finite State Automata, FSA).

При использовании FSA разрешённые слова с их вероятностями проставлены на дугах, составляющих пути. Каждое слово представляется также автоматом. Полная вероятность произнесения подсчитывается как произведение вероятностей, полученных на всех вложенных автоматах. Эти автоматы состоят из набора промежуточных состояний, начального состояния и набора конечных состояний, соединённых переходами. Каждый переход имеет начальное и конечное состояния, метку и вес.

Задачей декодера является оптимизация автомата. Декодер находит в лексиконе варианты произнесений слов и подставляет их в грамматику. Представление в виде фонетического дерева на данном этапе может быть использовано для уменьшения количества путей. Далее декодер определяет контекстно-зависимые модели для каждой фонемы в контексте и подставляет их в

граф. Для чего используют преобразователи с конечным числом состояний (Finite State Tranducers, FST) и взвешенные преобразователи с конечным числом состояний (Weighted Finite State Tranducers, WFST) [100].

Пример FST приведен на рисунке 5.6.

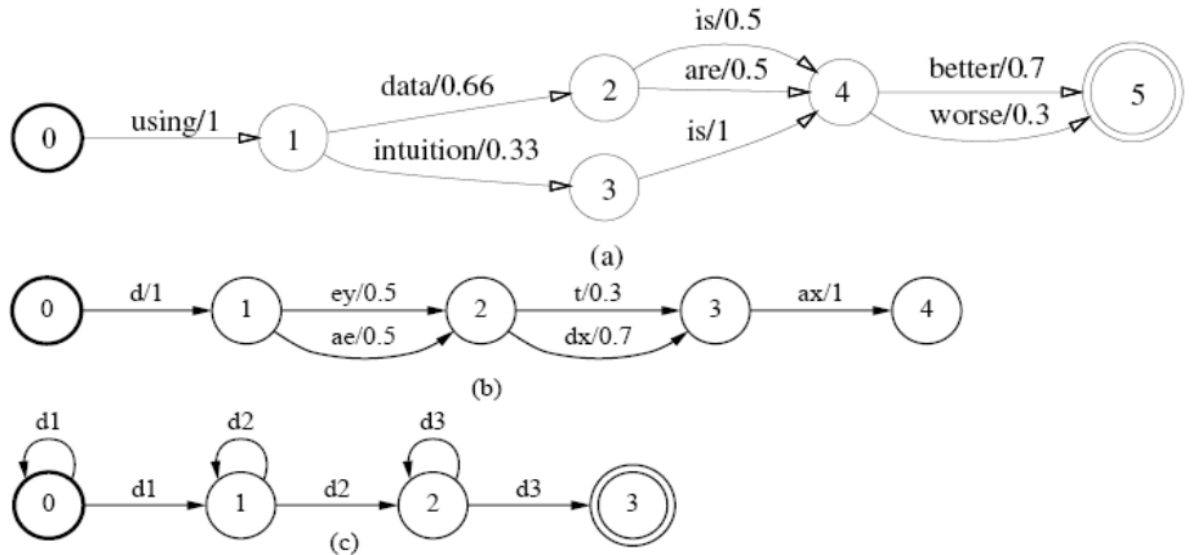


Рисунок 5.6 – Пример FST: (a) – граф последовательности слов; (b) – граф слова «data» с двумя вариантами произнесения; (c) – автомат, описывающий стандартную марковскую цепь для фонемы ‘d’.

Пример WFST отображён на рисунке 5.7.

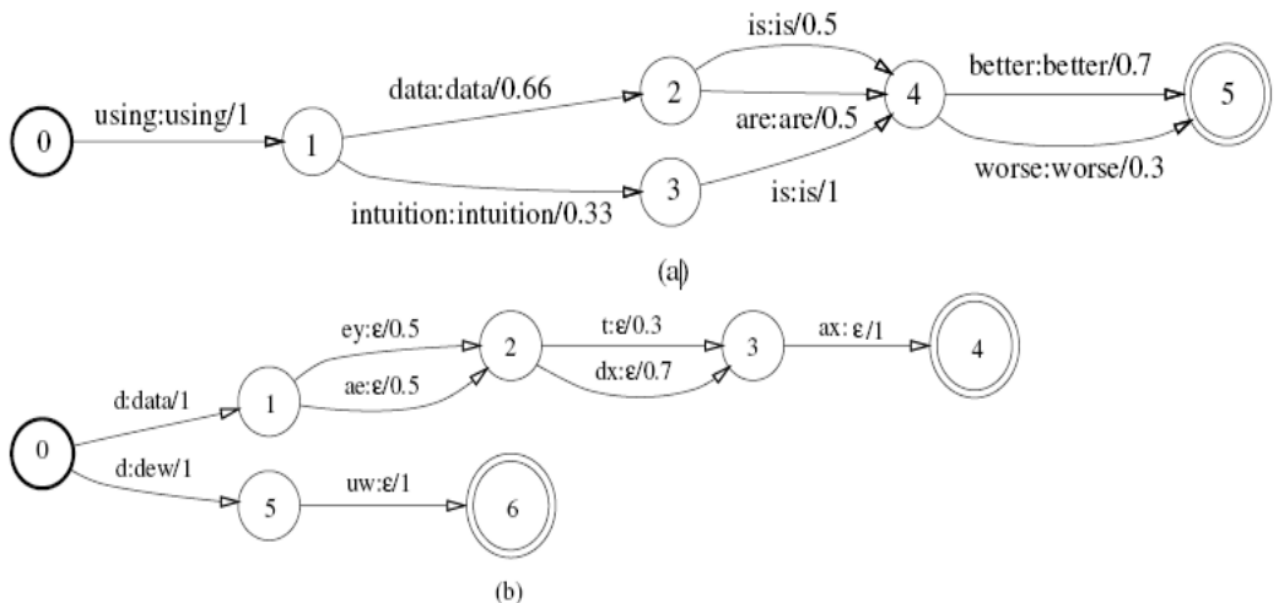


Рисунок 5.7 – Пример WFST: (a) – граф последовательности слов; (b) – объединенный граф слов «data» и «dew»

Каждый переход на рисунке 5.7 имеет идентичные метки входа и выхода. Поскольку слова кодируются выходной меткой, стало возможно объединять преобразователи для нескольких слов (слова «data» и «dew»). Аналогично можно объединять НММ фонем.

В WFST объединяются различные уровни, например, уровень фонем и уровень слов. Благодаря этому подходу, удаётся объединить в одну сеть WFST различные источники знаний – марковские модели, лексиконы, N-граммные статистические модели языка. При поиске оптимального пути на графе декодеру не придётся обращаться к представлению фонем, лексикону, модели языка – вся информация уже заключена в структуре графа. Благодаря этому, декодер упрощается и ускоряется, не изменяя ЯМ и АМ, поскольку ему остаётся только подставлять вероятности эмиссии в соответствии с рассматриваемыми гипотезами. При этом использование WFST-графа для декодирования позволяет значительно снизить WER [101]. Очевидно, что WFST обладает явным преимуществом перед FST – преобразователем с конечным числом состояний.

В работе декодер представлен в виде следующего графа:

$$HCLG = H \circ C \circ L \circ G = asl(\min(rds(\det(H \circ \min(\det(L \circ G))))),$$

где « $\circ$ » – операция конкатенации, *asl* – операция добавления собственных петель в графе, *rds* – операция удаления неоднозначных символов [173], реализованные в OpenFST; *H* – граф, описывающий акустическую модель (входные состояния – контекстный вектор); *G* – граф, описывающий языковую модель; *L* – граф, построенный средствами Kaldi из текстового файла, содержащего словарь транскрипций; *C* – граф, построенный средствами Kaldi из АМ, который описывает контекстные состояния фонем.

Отличие декодера для НММ-моделей и ANN-моделей состоит в том, что в качестве вершин рассматриваются фонемы, а не их состояния, но с использованием алгоритма forward-backward для получения весовых значений, в то время как для НММ используется алгоритм Витерби. Набор фонем ограничен в соответствии с транскрипциями.

Текст, полученный после декодирования, подвергается денормализации, в ходе которой расставляются знаки препинания.

### 5.1.3 Реализация модуля денормализации текста

Для решения задачи денормализации результатов распознавания речи обучена система ВРЕ-токенизации на основе решения YouTokenToMe [180] и текстового корпуса, на котором обучалась ЯМ. Размер ВРЕ-словаря составил 32 тыс. токенов. Дополнительно словарь расширен путем добавления в него знаков препинания и цифр, а также позиций начала («<BOS>») и конца («<EOS>») предложения, а также неизвестного токена («<UNK>») и токена иностранного языка («<ENG>»). На основе DAWG созданы 2 словаря: dict\_num2word для кодирования слов (слово → индекс); dict\_word2num для декодирования слов (индекс → слово).

Входные и выходные данные (нормализованный текст) сформированы из данных, предоставленных для соревнования Kaggle Russian text Normalization [181], объёмом около 8 Гб. Эти данные представляют собой таблицы, в поля которых помещены:

- 1) имя типа/метки текста, подвергающегося нормализации;
- 2) оригинальное написание текста в документе;
- 3) нормализованный текст.

Дополнительно этот набор данных преобразован следующим образом:

- слова, обозначающие римские цифры, трансформированы в арабские;
- слова на латинице маскировались токеном «<ENG>».
- удалены лишние пробелы;
- строки подобные «-1925» разделены на массив слов [«-», «1925»];
- для строк типа «768 - 845», а также «3 521» удалены лишние пробелы;
- трансформированы нормализованные url-адреса;
- удалены предложения с украинскими символами;



- все символы «;» заменены на «,»; символы многоточия заменены на точку; символы «?!» на «?».

- игнорируются символы пунктуации вне заданного словаря слов;
- проверяется предложение на принадлежность русскому языку с помощью разработанной модели определения языка текста, описанной в пункте 2.4.2.

В результате предварительной обработки получаем исходный массив предложений (`arr_source_sent`) и нормализованный массив предложений (`arr_tgt_sent`). Данные в `arr_source_sent` не содержат знаков препинания. Данные в `arr_tgt_sent` содержат знаки препинания.

Дополнительно набор данных отсортирован по максимальной длине предложений равной 120 ВРЕ-токенов. Разделение на обучающую и тестовую выборки совершено в соотношении 80/20.

Процесс кодирования слов заключался в следующем:

- 1) извлекалось входное предложение;
- 2) предложение делилось на ВРЕ-токены с помощью ВРЕ-модели;
- 3) осуществлялась кодировка слов с помощью `num2word`, при этом цифро-буквенные комплексы, аббревиатуры, а также числа считывались посимвольно.
- 4) если длина полученной кодировки больше значения максимального количества токенов, кодировка игнорируется, если меньше, то дополняется индексами токена EOS.

После этого данные трансформируются в тип данных `tf.data.Dataset` [182] в виде словаря с значениями `inputs` (исходный текст); `targets` (получаемый текст). Извлекая значения `inputs/targets` происходит деление набора данных на пакеты (`batch`), которые подаются на нейросеть.

Модель, используемая для задачи нормализации текста, основана на архитектуре `Transformer-Encoder` и имеет следующие гиперпараметры: количество скрытых слоёв – 6; размер скрытых слоёв – 128; размер вектора представлений слов – 128; количество блоков энкодера – 4; коэффициент `dropout` – 0.2; коэффициент обучения – 0.001; размер ВРЕ-словаря – 32050, общее количество пунктуационных символов – 6.

В данной реализации внесено несколько изменений в архитектуру нейронной сети:

1) Изменена процедура позиционного кодирования слов:

$$PE_{t-j} = \left[ \dots \sin\left(\frac{t-j}{10000^{2i/d_k}}\right) \cos\left(\frac{t-j}{10000^{2i/d_k}}\right) \dots \right]^T, \quad (5.3)$$

где  $t$  – индекс выходного токена;  $j$  – индекс входного токена;  $i = [0 \dots d_k/2]$ ;  $d$  – размерность позиции векторного представления.

$$PE_t = \begin{bmatrix} \sin(c_0 t) \\ \cos(c_0 t) \\ \dots \\ \sin\left(\frac{c_{\frac{d}{2}-1} t}{2}\right) \\ \cos\left(\frac{c_{\frac{d}{2}-1} t}{2}\right) \end{bmatrix} \quad (5.4)$$

2) Изменена процедура трансформации тензоров через механизм self-attention:

$$Q, K, V = HW_q, H, HW_v, \quad (5.5)$$

$$A_{t,j}^{PE} = Q_t^T K_j + Q_t^T PE_{t-j} + u^T K_j + v^T PE_{t-j}, \quad (5.6)$$

$$Att(Q, K, V) = softmax(A^{PE})V, \quad (5.7)$$

где  $Q$  – вектор запроса (query) для текущего токена во фразе;  $V$  – вектор значения;  $K$  – вектор ключа (key);  $W$  – весовые матрицы,  $u, v$  – обучаемые тензоры;  $Att$  – тензор многослойного внимания;  $A$  – тензор self-attention.

3) На выходе модели дополнительно используется модель условных случайных полей (Conditional Random Fields, CRF) [183] для предсказания следующей метки. Имея множество  $s = [s_1, s_2, \dots, s_T]$ , с соответствующими метками классов  $y = [y_1, y_2, \dots, y_T]$ , и  $Y(s)$  представляет все допустимые последовательности меток. Вероятность для  $y$  вычисляется следующим образом:

$$P(y|s) = \frac{\sum_{t=1}^T e^{f(y_{t-1}, y_t, s)}}{\sum_{y'} \sum_{t=1}^T e^{f(y'_{t-1}, y'_t, s)}}, \quad (5.8)$$

где  $f(y_{t-1}, y_t, s)$  – функция, вычисляющая оценку перехода от  $y_{t-1}$  к  $y_t$ , а также оценку для  $y_t$ . Главная цель CRF – максимизировать  $P(y|s)$ . При декодировании используется алгоритм Витерби.

В качестве функции активации используется Mish [184]. В качестве функции оптимизации градиентного спуска используется Ranger [48]. Также используются техники clipping gradient. Для оценки эффективности разработанной модели нормализации текста использовались метрики точности (accuracy). Общая точность модели составила 0.87. Более подробные результаты тестирования приведены в таблице 5.1.

Таблица 5.1 – Результаты тестирования модели на тестовом наборе данных

Пунктуация	«.»	«.»	«-»	«?»	«!»	«:»
Accuracy	0.952	0.912	0.819	0.823	0.852	0.867

## 5.2 Оценка эффективности разработанной системы

Для сравнения результатов распознавания с ASR, полученной в рамках работы, выбрана система компании ЦРТ (CST ASR), а также Google Cloud ASR.

1) CST ASR [185] – технология распознавания речи от лидера российского рынка компании ЦРТ, доступная в виде простого в использовании облачного API. Технология распознавания речи поддерживает большие объемы записанной речи, в том числе потоковую передачу аудио в режиме реального времени. Поддерживаемые языки: испанский, русский, английский (США), казахский. Режим микрофона: 16 kHz, 8 kHz. Стоимость: 60 коп/мин. Существует несколько режимов работы ASR API.

- Офлайн-распознавание или слитное распознавание – режим распознавания, предполагающий обработку фонограммы после того, как ее запись завершена. Результат будет доступен пользователю целиком после обработки аудиофайла.

- Онлайн-распознавание – режим распознавания, предполагающий обработку фонограммы во время ее записи. Результат отображается по мере диктовки фразы в режиме реального времени.

- Режим взаимодействия без установки сессии. API состоит из одного метода, который включает в себя создание сессии и единичное использование

распознавания речи. Automated Speech Recognition API имеет описание в стандарте OpenAPI Specification, который содержит сведения о доступных операциях в API и о том, как необходимо структурировать данные запросов и ответов для API.

Доступные модели CST ASR:

- RU – Средний микрофон (FarField). Модель Средний микрофон (FarField) идеально подойдет для распознавания записей полилогов из переговорных комнат, ритейл локаций, а также для разработки голосовых помощников. Доступна в пакетном и потоковом режимах.

- RU – Телефонный канал (TelecomRus) Модель Телефонный канал (TelecomRus) предназначена для распознавания записей телефонных разговоров на русском языке. Доступна в пакетном режиме, то есть принимает на вход законченную запись.

- RU – Телефонный канал (IVR) Модель Телефонный канал (IVR) предназначена для создания IVR систем, где требуется минимальная задержка распознавания. Доступна в режиме бета-тестирования с помощью API потокового распознавания по протоколу gRPC.

2) Google Cloud ASR [186]. Система распознавания речи Google Cloud ASR основана на передовых алгоритмах глубокого обучения, разработанных компанией Google для автоматического распознавания речи (ASR). Основные характеристики:

- поддержка распознавания речи для более чем 125 языков;
- поддержка распознавания речи в режиме реального времени, когда API обрабатывает аудиовход, транслируемый с микрофона или отправляемый из предварительно записанного аудиофайла (встроенного или через облачное хранилище);
- функция преобразования речи в текст может распознавать отдельные каналы в многоканальных ситуациях (например, при видеоконференции) и комментировать стенограммы, чтобы сохранить порядок;
- обработка шумного звука из различных сред, не требуя дополнительного шумоподавления;

- автоматическая расстановка знаков пунктуации в транскрипции (например, запяты, вопросительные знаки и точки);
- распределение итогов транскрипции по разным дикторам.

Для проверки CST ASR используется API, предоставленное разработчиками компании. Для проверки Google Cloud ASR используется платное API (бесплатный доступ).

В качестве метрик для оценивания систем используются WER и SER. Дополнительно использовалась метрика оценки скорости получения результата распознавания (speed rate, SR).

$$SR = \frac{T_{rec}}{T}, \quad (5.9)$$

где  $T_{rec}$  – время, затраченное на распознавание аудиозаписей общей продолжительностью  $T$ . Если  $SR < 1$ , то выполняется требование к распознаванию в режиме реального времени.

Оценка системы ASR, описанной в данной работе (ASR\_work) производилась на следующей конфигурации оборудования:

- 6-ядерный процессор AMD Ryzen 2600 с тактовой частотой 3.4 ГГц,
- 32 ГБ ОЗУ;
- операционная система Linux Mint 19.2 Tina;
- графический ускоритель Nvidia GTX 1060 6 Гб.

В качестве тестовой выборки аудиоданных использовался тестовый, который представлен в виде диалогов между двумя дикторами в виде телефонного разговора. Предварительно используя модель идентификации дикторов набор данных разделен на N-дикторов, речевые отрезки длиной более 16 сек, используя механизм VAD делились на несколько сегментов. Сеть основана на модели ResBlock, в качестве обучающей и тестовой выборки для модели идентификации дикторов использован корпус VoxForge, который предварительно обработан следующим образом – удалены речевые данные длительностью менее 3 сек; речевые данные длительностью более 15 сек разделены на несколько файлов. Алгоритм определения дикторов заключается в следующем.

1. Определяется доверительный порог ( $id_{th}$ ).

2. Вычисляются векторные представления ( $F_{id}$ ) для всех аудиофайлов, используя ResBlock
3. Определяем  $F_i$ , при  $i=0 \dots N-1$ , где  $N$  – общее число дикторов, относим к диктору  $i$ , перемещаем аудио в директорию, соответствующую данному диктору.
4. Определяется евклидово расстояние ( $e_i$ ) между  $F_{id}$  и  $F_i$ . Считается, что аудиофайл относится к текущему диктору, если  $e_i \leq id_{th}$  (чем меньше расстояние – тем больше сходство), в данном случае файл перемещается в директорию, соответствующую текущему диктору.
5. Если  $e_i > id_{th}$ , то  $F_i$  относим к диктору  $i+1$ , перемещаем аудио в директорию, соответствующую данному диктору, при этом повторяем пункт 4, данная процедура повторяется до тех пор, пока не будут определены дикторы для всех оставшихся  $F_{id}$ .

В результате вышеуказанного алгоритма формируется речевой корпус, размеченный по дикторам. Формат аудио: PCM-WAV, 16 кГц, 16 бит. Общее количество дикторов составило 1577. Общая продолжительность аудио – 7,2 ч.

Результаты оценивания систем приведены в таблице 5.2.

Таблица 5.2 – Результаты оценивания систем распознавания речи

ASR	WER	SER	SR
CST	0.3805	0.96	1,17
Google Cloud	0.2119	0.82	1,05
ASR_work	0.2763	0.91	0,15

По результатам, представленным в таблице 5.2, ASR\_work оказалась лучше системы CST по качеству распознавания на 10,42%. В то же время система Google Cloud оказалась лучше ASR\_work на 6,44%.

Из проведенных численных исследований следует, что ASR\_work обладает достаточной точностью в задаче распознавания слитной русской речи, используя для своего обучения значительно меньше данных, а также превосходит рассмотренные системы по показателю SR в среднем на 0,96.

### 5.3 Выводы к главе 5

1. Нейросетевые модели, методы и алгоритмы, представленные в диссертации, реализованы в едином программном комплексе, представляющим ASR-систему, ориентированную на русский язык. Данная ASR-система работает в двух режимах: обучения и распознавания. В режиме обучения происходит предварительная обработка текстовых и речевых данных, генерация акустической модели и языковой модели, словаря транскрипций, содержащего 500 тыс. слов.

2. Разработана архитектура языковой модели, содержащая LSTM-блоки, проведена оценка качества с помощью метрики перплексии, ее значение составило 368,73, что говорит о допустимом качестве.

3. Разработана нейросетевая модель денормализации распознанного текста на базе архитектуры Transformer, позволяющая восстанавливать знаки препинания, строчные числа заменять цифрами, восстанавливать прописные буквы в именах собственных и аббревиатурах, которая обеспечивает точность 0,87.

4. Проведена оценка эффективности разработанной ASR-системы, система по точности превосходит решение компании ЦРТ на 10,42%, уступая решению от компании Google на 6,44% по показателю WER, по показателю SR – превосходит рассмотренные системы в среднем на 0,96. Таким образом, ASR\_work обладает достаточной точностью и быстродействием в задаче распознавания слитной русской речи, используя для своего обучения значительно меньше данных.

## ЗАКЛЮЧЕНИЕ

Диссертация является законченной научно-исследовательской работой, в которой получено решение актуальной научно-технической задачи повышения эффективности дикторонезависимой системы автоматического распознавания слитной русской речи, учитывающей её особенности и адаптирующейся под любую предметную область за счет модернизации алгоритмов системного анализа, обработки и распознавания речевой информации, а также автоматической обработки текстовых данных. Основные научные результаты и выводы состоят в следующем.

1. Анализ состояния исследований в области распознавания речи показал, что для построения АМ наиболее перспективным представляется нейросетевая параметризация речевого сигнала и модификация акустических признаков для получения адаптивных и дискриминативных характеристик; для построения ЯМ – нейросети с архитектурой LSTM и Transformer; для построения транскриптора – ИНС с архитектурой seq2seq; для фонемного распознавания – глубокие нейросети; для декодирования – подходы, основанные на WFST-графе.

2. Для формирования собственного аннотированного речевого корпуса:

– модифицирован классический алгоритм парного выравнивания Смита-Уотермана для проверки соответствия текстовых расшифровок и аудио за счет запоминания начала и конца совпадения в исходных данных, что повысило его точность в среднем на 10,5 % по сравнению с исходным алгоритмом;

– предложена техника аугментации речевых данных, позволяющая повысить робастность АМ и уменьшить WER на 1,14% для трифонной АМ и на 1,7% для монофонной;

– разработаны алгоритмы нормализации текстов на основе сверточных сетей, позволяющие: определить язык отдельного предложения с точностью 82,5%; провести согласование чисел для корректной расшифровки цифро-буквенных комплексов с точностью по критерию UAS – 94,3%, по критерию LAS – 90,2%.



В результате использования предложенных алгоритмов для обучения АМ подготовлен речевой корпус длительностью более 29 часов, для обучения ЯМ сформирована база нормализованных текстов объемом 15,2 Гб.

3. Разработаны архитектуры ИНС для автоматического формирования словаря транскрипций. Предложенные нейросетевые модели позволяют:

- автоматически определять позицию ударения в слове за счет модернизации архитектуры ИНС Transformer, которая заключается в увеличении количества слоёв, использовании методов градиентного отсечения для оптимизации параметра скорости обучения, что увеличило точность определения позиции ударения на 10%;

- генерировать практические транскрипции англоязычных слов и слов-исключений путем усовершенствования seq2seq модели за счет применения механизма обучения с подкреплением и метода beam-search для выбора наиболее вероятной последовательности символов, что увеличило точность модели по критерию количества ошибочно сгенерированных символов на 0,8% и 3%, по критерию неправильно сгенерированных слов на 0,6% и 9% соответственно.

4. Для получения робастных акустических признаков и обучения АМ предложена нейросетевая параметризация, основанная на объединении ансамбля нейронных сетей с узким горлом и архитектуры ResNet-50, что позволяет повысить точность распознавания на 2,7% по сравнению с моделью, извлекающей стандартные bottleneck-признаки.

5. Для разработки классификатора фонем усовершенствованы методы нейросетевой классификации за счет использования механизма внимания в последнем скрытом слое сети, включающей в себя архитектуры TDNN и BLSTM.

6. Проведена оценка качества распознавания с использованием разработанных АМ и классификатора фонем, обученных на небольшом объеме данных (около 20 часов). Использование предложенной модели нейросетевой параметризации речевого сигнала позволяет повысить точность распознавания на 2,7% по сравнению с моделью, извлекающей стандартные bottleneck-признаки, а разработанные АМ и архитектура нейросети для распознавания фонем показывают

результат распознавания на 4,1% лучше, чем ASR от Google, которая обучалась на речевых базах объемом в десятки тысяч часов.

7. На основе предложенных методов и моделей разработана ASR-система, которая обучалась на речевом корпусе объемом около 7,2 Гб. Авторская система по качеству превосходит решение компании ЦРТ на 10,42%, уступая Google на 6,44%. Разработанная ASR обладает достаточной точностью и превосходит ASR Google и ЦРТ по скорости распознавания более, чем в 7 раз.

8. Модернизация алгоритмов системного анализа, обработки и распознавания речевой информации, а также автоматической обработки текстовых данных позволила повысить эффективность дикторонезависимой системы автоматического распознавания слитной русской речи, работающей с быстродействием и точностью, достаточными для практических задач, и требующей для своего обучения объем данных более, чем в 500 раз меньший, чем существующие аналоги.

## СПИСОК СОКРАЩЕНИЙ

- ANN – искусственная нейронная сеть (Artificial Neural Network)
- ASR – Автоматическое распознавание речи (Automatic Speech Recognition)
- BLSTM – двунаправленная долгая кратковременная нейронная сеть (Bidirectional Long Short Term Memory)
- BMMI – Boosted Maximum Mutual Information
- BP – обратное распространение (Backpropagation)
- BPE – Byte Pair Encoding
- CD-DNN – контекстно зависимые нейронные сети (Context-Dependent Deep Neural Network)
- CE – кросс-энтропия (Cross-Entropy)
- CMN – нормализация среднего кепстра (Cepstral Mean Normalization)
- CMVN – нормализация среднего кепстра и дисперсии (Cepstral Mean and Variance Normalization)
- CNN – сверточная нейронная сеть (Convolutional Neural Network)
- DAWG – направленный ациклический граф слов (Directed Acyclic Word Graphs)
- DBN – глубокая сеть доверия (Deep Belief Network)
- DNN – глубокая нейронная сеть (Deep Neural Network)
- DPT – дискриминативное предобучение (Discriminative Pretraining)
- EM – максимизация математического ожидания (Expectation-Maximization)
- FBANK – логарифмы энергии спектра набора треугольных Mel-фильтров (Mel-frequency filterbank log energies)
- fDLR – дискриминативная линейная регрессия в пространстве признаков (feature Discriminant Linear Regression)
- fMLLR – пространство характеристик максимального правдоподобия линейной регрессии (feature-domain Maximum Likelihood Linear Regression)
- GMM – гауссовы смешанные модели (Gaussian Mixture Model)
- HMM – скрытые марковские модели (Hidden Markov Model)
- LAS – оценка маркированной принадлежности (Labelled Attachment Score)

- LDA – линейный дискриминантный анализ (Linear Discriminant Analysis)
- LSTM – нейросеть с долгой краткосрочной памятью (Long Short Term Memory)
- MFCC – мел-частотные кепстральные коэффициенты (Mel-Frequency Cepstral Coefficients)
- ML – максимальное правдоподобие (Maximum Likelihood)
- MLLR – линейная регрессия максимального правдоподобия (Maximum Likelihood Linear Regression)
- MLLT – линейное преобразования, максимизирующее среднее правдоподобие (Maximum Likelihood Linear Transformation)
- MMI – максимизация взаимной информации (Maximum Mutual Information)
- MPE – минимизация фоновых ошибок (Minimum Phone Error)
- MSE – среднеквадратичное отклонение (Mean Square Error)
- PCA – анализ главных компонент (Principal Component Analysis)
- PER – процент неверно трансформированных символов (Phoneme Error Rate)
- PLP – перцептивное линейное предсказание (Perceptual Linear Prediction)
- PPL – перплексия (Perplexity)
- QRNN – квазирекуррентная нейросеть (Quasi Recurrent Neural Network)
- RBM – ограниченная машина Больцмана (Restricted Boltzmann Machine)
- ReLU – Rectified Linear Unit
- RL – обучение с подкреплением (Reinforcement Learning)
- SER – процент неверно распознанных предложений (Sentence Error Rate)
- SGD – стохастический градиентный спуск Stochastic Gradient Descent
- ST – обучение с использованием критериев разделения последовательностей (Sequence-discriminative Training)
- TDNN – нейронная сеть с временной задержкой (Time Delay Neural Network)
- UAS – оценка немаркированной принадлежности (Unlabelled Attachment Score)
- UDT – банк деревьев универсальных зависимостей (Universal Dependencies Treebank)

ULMfit – универсальная языковая модель тонкой настройки (Universal Language Model fine-tuning)

VAD – детектор активности речи (Voice Activity Detector)

VTLN – нормализация по длине вокального тракта (Vocal Tract Length Normalization)

WER – процент неверно распознанных слов (Word Error Rate)

WFST – взвешенный преобразователь с конечным числом состояний (Weighted Finite State Transducers)

АМ – Акустическая модель

ЯМ – Языковая модель

## СПИСОК ЛИТЕРАТУРЫ

1. The Holy Grail: Automatic speech recognition for low-resource languages [Электронный ресурс]. – URL: <https://beckman.illinois.edu/news/2016/03/asr-jyothi> (дата обращения: 20.05.2017).
2. Allison, B., Guthrie, D., Guthrie, L. Another Look at the Data Sparsity Problem / Allison B., D. Guthrie, L. Guthrie // Text, Speech and Dialogue, Lecture Notes in Computer Science. – 2006. – Т. 4188. – С. 327–334.
3. Davis, S.B. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. Т. 28 / S.B. Davis, P. Mermelstein. – 1980.
4. Huang, X. Spoken Language Processing: A Guide to Theory, Algorithm & System Development / X. Huang, A. Acero, H.-W. Hon. – 2001.
5. Рабинер, Л. Цифровая обработка речевых сигналов / Л. Рабинер, Р. Шафнер. – М.: Радио и связь, 1981. – 496 с.
6. Шарий, Т.В. О Проблеме Параметризации Речевого Сигнала В Современных Системах Распознавания Речи / Т.В. Шарий // Вестник Донецкого Национального Университета. – 2008. – Т. 2. – С. 536-541.
7. Hermansky, H. Perceptual linear predictive (PLP) analysis of speech / H. Hermansky // Journal of the Acoustical Society of America. – 1990.
8. Liu, F.-H. Efficient cepstral normalization for robust speech recognition / F.-H. Liu, R.M. Stern, X. Huang, A. Acero. – 1993.
9. Acero, A. Augmented Cepstral Normalization for Robust Speech Recognition / A. Acero, X. Huang // IEEE Workshop on Automatic Speech Recognition. – 1995. – С. 147-148.
10. Haeb-Umbach, R. Linear Discriminant Analysis for improved large vocabulary continuous speech recognition / R. Haeb-Umbach, H. Ney // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings. – 1992. – Т. 1.
11. Saon, G. Maximum likelihood discriminant feature spaces / G. Saon, M. Padmanabhan, R. Gopinath, S. Chen // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2000. – Т. 2.
12. Deng, L. High-performance robust speech recognition using stereo training

data / L. Deng, A. Acero, L. Jiang, J. Droppo, X. Huang // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2001. – Т. 1.

13. Hilger, F. Quantile based histogram equalization for noise robust large vocabulary speech recognition / F. Hilger, H. Ney // IEEE Transactions on Audio, Speech and Language Processing. – 2006. – Т. 14. – № 3.

14. Eide, E. Parametric approach to vocal tract length normalization / E. Eide, H. Gish // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 1996. – Т. 1.

15. Gales, M.J.F. Maximum likelihood linear transformations for HMM-based speech recognition / M.J.F. Gales // Computer Speech and Language. – 1998.

16. Povey, D. FMPE: Discriminatively trained features for speech recognition / D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, G. Zweig // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings. – 2005. – Т. 1.

17. Червяков, Н. Методы обработки исходного речевого сигнала / Н. Червяков, Н. Кучукова // Научные ведомости. Серия Экономика. Информатика. – 2016. – Т. 40. – № 23(244). – С. 148-155.

18. Plahl, C. Neural Network based Feature Extraction for Speech and Image Recognition / C. Plahl. – Ph.D. diss. – Dept. Computer Science, RWTH Aachen, Aachen, Ger, 2014.

19. Grézl, F. Probabilistic and bottle-neck features for LVCSR of meetings / F. Grézl, M. Karafiát, S. Kontár, J. Černocký // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2007. – Т. 4.

20. Rabiner, L. A tutorial on hidden Markov models and selected applications in speech recognition / L. Rabiner // Proceedings of the IEEE, 77(2). – 1989. – С. 257-286.

21. Jelinek, F. Continuous Speech Recognition by Statistical Methods / F. Jelinek // Proceedings of the IEEE. – 1976.

22. Тампель, И. Автоматическое распознавание речи. Учебное пособие / И. Тампель, А. Карпов. – СПб: Университет ИТМО, 2016. – 138 с.

23. Ронжин, А. Автоматическое распознавание русской речи / А. Ронжин,

- И. Ли // Вестник Российской академии наук. – 2007. – Т. 77. – № 2. – С. 133-138.
24. Bishop, C. *Pattern Recognition and Machine Learning* / C. Bishop. – New York: Springer–Verlag, 2006. – 729 с.
25. Рабинер, Л. Скрытые марковские модели и их применение в избранных приложениях при распознавании речи: Обзор / Л. Рабинер // ТИИЭР. – 1989. – Т. 77. – № 2. – С. 86-120.
26. Dempster, A.P. Maximum Likelihood from Incomplete Data Via the EM Algorithm / A.P. Dempster, N.M. Laird, D.B. Rubin // *Journal of the Royal Statistical Society: Series B (Methodological)*. – 1977. – Т. 39. – № 1.
27. Povey, D. *Discriminative Training for Large Vocabulary Speech Recognition* / D. Povey. – Ph. D. dis. – Cambridge: Cambridge University Engineering Dept., 2003.
28. Juang, B.H. Minimum classification error rate methods for speech recognition / B.H. Juang, W. Chou, C.H. Lee // *IEEE Transactions on Speech and Audio Processing*. – 1997. – Т. 5. – № 3.
29. Povey, D. Minimum phone error and I-smoothing for improved discriminative training / D. Povey, P.C. Woodland // *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. – 2002. – Т. 1.
30. Zheng, J. Improved discriminative training using phone lattices / J. Zheng, A. Stolcke // *9th European Conference on Speech Communication and Technology*. – 2005.
31. Saon, G. Penalty function maximization for large margin HMM training / G. Saon, D. Povey // *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. – 2008.
32. Saon, G. Large margin semi-tied covariance transforms for discriminative training / G. Saon, D. Povey, H. Soltan // *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. – 2009.
33. Povey, D. The subspace Gaussian mixture model - A structured model for speech recognition / D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, R.C. Rose, P. Schwarz, S. Thomas // *Computer Speech and Language*. – 2011. – Т. 25. – № 2.
34. Boulard, H. Towards increasing speech recognition error rates / H. Boulard,



- H. Hermansky, N. Morgan // *Speech communication*. – 1996. – Т. 18. – № 3. – С. 205-231.
35. Пикалёв, Я.С. Глубинное обучение в задаче автоматического распознавания речи / Я.С. Пикалёв // *Интеллектуальные технологии и проблемы математического моделирования: материалы Всерос. науч. конф. (Дивноморское, 24 – 26 сентября 2018 г.)* / ред. Б.В. Соболев. – Ростов-на-Дону: ДГТУ, 2018. – С. 16-17.
36. Hinton, G. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups / G. Hinton, L. Deng, D. Yu, Dahl, George E, A.R. Mohammed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, B. Kingsbury // *IEEE Signal processing magazine*. – 2012. – Т. 29. – № 6. – С. 82-97.
37. Хайкин, С. Нейронные сети: полный курс / С. Хайкин; ред. Н. Куссуль. – 2е издание. – М.: Издательский дом Вильямс, 2008. – 1104 с.
38. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М.: Горячая линия – Телеком, 2006. – 452 с.
39. Rumelhart, D.E. Learning representations by back-propagating errors / D.E. Rumelhart, G.E. Hinton, R.J. Williams // *Nature*. – 1986. – Т. 323. – № 6088. – С. 533-536.
40. Yu, D. *Automatic Speech Recognition: A Deep Learning Approach* / D. Yu, L. Deng. – London: Springer, 2016. – 329 с.
41. Kipyatkova, I. DNN-based acoustic modeling for Russian speech recognition using Kaldi / I. Kipyatkova, A. Karpov // *International Conference on Speech and Computer*. – Springer, Cham, 2016. – С. 246-253.
42. Bottou, L. Online algorithms and stochastic approximations [Электронный ресурс]. – URL: <https://leon.bottou.org/publications/pdf/online-1998.pdf> (дата обращения: 09.06.2020).
43. Nesterov, Y. A method of solving a convex programming problem with convergence rate / Y. Nesterov // *Sov. Math. Dokl.* – 1983. – Т. 27. – № 2. – С. 372-376.
44. Kingma, D.P. Adam: A method for stochastic optimization / D.P. Kingma, J.L. Ba // *3rd International Conference on Learning Representations, ICLR 2015 – Conference Track Proceedings*. – 2015.

45. Igel, C. Improving the Rprop learning algorithm / C. Igel, M. Hüsken // Proceedings of the Second International Symposium on Neural Computation. – 2000.
46. Duchi, J. Adaptive subgradient methods for online learning and stochastic optimization / J. Duchi, E. Hazan, Y. Singer // COLT 2010 - The 23rd Conference on Learning Theory. – 2010.
47. Luo, L. Adaptive gradient methods with dynamic bound of learning rate / L. Luo, Y. Xiong, Y. Liu, X. Sun // 7th International Conference on Learning Representations, ICLR 2019. – 2019.
48. Zhang, M. Lookahead Optimizer: k steps forward, 1 step back / M. Zhang, J. Lucas, J. Ba, G.E. Hinton // Advances in Neural Information Processing Systems. – 2019. – C. 9593-9604.
49. Sutskever, I. Training Recurrent Neural Networks / I. Sutskever. – Toronto: University of Toronto, 2012. – 101 c.
50. Dahl, G. Phone recognition with the mean-covariance restricted Boltzmann machine / G. Dahl, M. Ranzato, A. Mohamed, G.E. Hinton // Advances in Neural Information Processing Systems. – 2010. – C. 469-477.
51. Hinton, G.E. A Fast Learning Algorithm for Deep Belief Nets / G.E. Hinton, S. Osindero, Y.-W. Teh // Neural Computation. – 2006. – T. 18. – № 7. – C. 1527-1554.
52. Bengio, Y. Greedy layer-wise training of deep networks / Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle // Advances in Neural Information Processing Systems. – 2007.
53. Su, H. Error back propagation for sequence training of Context-Dependent Deep NetworkS for conversational speech transcription / H. Su, G. Li, D. Yu, F. Seide // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2013.
54. Li, B. Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems / B. Li, K.C. Sim // Eleventh Annual Conference of the International Speech Communication Association. – 2010. – C. 526-529.
55. Gemello, R. Linear hidden transformations for adaptation of hybrid ANN/HMM models / R. Gemello, F. Mana, S. Scanzio, P. Laface, R. De Mori // Speech Communication. – 2010. – T. 49. – № 10-11. – C. 827-835.

56. Yao, K. Adaptation of context-dependent deep neural networks for automatic speech recognition / K. Yao, D. Yu, F. Seide, H. Su, L. Deng, Y. Gong // IEEE Spoken Language Technology Workshop. – 2012. – C. 366-369.

57. Chen, Y. Reconstructive discriminant analysis: A feature extraction method induced from linear regression classification / Y. Chen, Z. Jin // Neurocomputing. – 2012.

58. Seide, F. Feature engineering in Context-Dependent Deep Neural Networks for conversational speech transcription / F. Seide, G. Li, X. Chen, D. Yu // 2011 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2011, Proceedings. – 2011.

59. Xue, J. Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network / J. Xue, J. Li, D. Yu, M. Seltzer, Y. Gong // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2014.

60. Xiao, L. Regularized adaptation of discriminative classifiers / L. Xiao, J. Bilmes // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2006.

61. Yu, D. KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition / D. Yu, K. Yao, H. Su, G. Li, F. Seide // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2013.

62. Abdel-Hamid, O. Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code / O. Abdel-Hamid, H. Jiang // IEEE International Conference on Acoustics, Speech and Signal Processing. – 2013. – C. 7946-7946.

63. Saon, G. Speaker adaptation of neural network acoustic models using i-vectors / G. Saon, H. Soltau, D. Nahamoo, M. Picheny // 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013 - Proceedings. – 2013.

64. Rouvier, M. Speaker adaptation of DNN-based ASR with i-vectors: Does it actually adapt models to speakers? / M. Rouvier, B. Favre // Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. – 2014.

65. Senior, A. Improving DNN speaker independence with I-vector inputs / A. Senior, I. Lopez-Moreno // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2014.

66. Li, J. Factorized adaptation for deep neural network / J. Li, J.T. Huang, Y. Gong // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2014.

67. Liu, S. On combining DNN and GMM with unsupervised speaker adaptation for robust automatic speech recognition / S. Liu, K.C. Sim // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2014.

68. Dahl, G.E. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition / G.E. Dahl, D. Yu, L. Deng, A. Acero // IEEE Transactions on Audio, Speech and Language Processing. – 2012.

69. Abdel-Hamid, O. Exploring convolutional neural network structures and optimization techniques for speech recognition / O. Abdel-Hamid, L. Deng, D. Yu // Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. – 2013.

70. Abdel-Hamid, O. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition / O. Abdel-Hamid, A.R. Mohamed, H. Jiang, G. Penn // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2012.

71. Hochreiter, S. Long Short-Term Memory / S. Hochreiter, J. Schmidhuber // Neural Computation. – 1997.

72. Geiger, J.T. Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling / J.T. Geiger, Z. Zhang, F. Weninger, B. Schuller, G. Rigoll // Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. – 2014.

73. Гусак, Е.А. Применение специализированной нейросетевой архитектуры TDNN для распознавания речевых сигналов [Электронный ресурс]. – URL: [http://ea.donntu.edu.ua/bitstream/123456789/14270/1/4\\_Гусак.pdf](http://ea.donntu.edu.ua/bitstream/123456789/14270/1/4_Гусак.pdf) (дата обращения: 09.06.2020).

74. Deng, L. Deep convex net: A scalable architecture for speech pattern classification / L. Deng, D. Yu // Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. – 2011.

75. Hutchinson, B. A deep architecture with bilinear modeling of hidden representations: Applications to phonetic recognition / B. Hutchinson, L. Deng, D. Yu // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings. – 2012.

76. Zhang, Y. Towards end-to-end speech recognition with deep convolutional neural networks / Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, A. Courville // Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. – 2016.

77. Graves, A. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks / A. Graves, S. Fernández, F. Gomez, J. Schmidhuber // ACM International Conference Proceeding Series. – 2006.

78. Graves, A. Towards end-to-end speech recognition with recurrent neural networks / A. Graves, N. Jaitly // 31st International Conference on Machine Learning, ICML 2014. – 2014.

79. Yu, D. Feature Learning in Deep Neural Networks - studies on Speech Recognition Tasks / D. Yu, M. Seltzer, J. Li // ICLR. – 2013.

80. Hermansky, H. Tandem connectionist feature extraction for conventional HMM systems / H. Hermansky, D.P.W. Ellis, S. Sharma // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2000. – T. 3.

81. Sainath, T.N. Auto-encoder bottleneck features using deep belief networks / T.N. Sainath, B. Kingsbury, B. Ramabhadran // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2012.

82. Yu, D. Improved bottleneck features using pretrained deep neural networks / D. Yu, M.L. Seltzer // Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. – 2011.

83. Gehring, J. Extracting deep bottleneck features using stacked auto-encoders / J. Gehring, Y. Miao, F. Metze, A. Waibel // ICASSP, IEEE International Conference on

Acoustics, Speech and Signal Processing - Proceedings. – 2013.

84. Zhang, Y. Extracting deep neural network bottleneck features using low-rank matrix factorization / Y. Zhang, E. Chuangsuwanich, J. Glass // IEEE international conference on acoustics, speech and signal processing (ICASSP). – 2014. – С. 185-189.

85. Chen, S.F. Empirical study of smoothing techniques for language modeling / S.F. Chen, J. Goodman // Computer Speech and Language. – 1999.

86. Bell, T. Modeling for text compression / T. Bell, I.H. Witten, J.G. Cleary // ACM Computing Surveys (CSUR). – 1989.

87. Chen, S. Evaluation metrics for language models / S. Chen, D. Beeferman, R. Rosenfeld // Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. – 1998.

88. Mikolov, T. Recurrent neural network based language model / T. Mikolov, M. Karafiát, L. Burget, C. Jan, S. Khudanpur // Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010. – 2010.

89. Sundermeyer, M. LSTM neural networks for language modeling / M. Sundermeyer, R. Schlüter, H. Ney // 13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012. – 2012.

90. Vaswani, A. Attention is all you need / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin // Advances in Neural Information Processing Systems. – 2017.

91. Alammar, J. The Illustrated GPT-2 (Visualizing Transformer Language Models) [Электронный ресурс]. – URL: <http://jalammar.github.io/illustrated-gpt2/> (дата обращения: 09.06.2020).

92. Radford, Alec. Language Models are Unsupervised Multitask Learners | Enhanced Reader / Radford Alec, Wu Jeffrey, Child Rewon, Luan David, Amodei Dario, Sutskever Ilya // OpenAI Blog. – 2019.

93. Богданова, Н.В. Живые фонетические процессы русской речи: пособие по спецкурсу / Н.В. Богданова. – СПб: Филологический факультет СПбГУ, 2001. – 186 с.

94. Кривнова, О.Ф. Многофункциональный автоматический транскриптор русских текстов / О.Ф. Кривнова, Л.М. Захаров, Г.С. Строкин // Труды

международного конгресса «Русский язык: исторические судьбы и современность». – 2001. – С. 408-409.

95. Bisani, M. Joint-sequence models for grapheme-to-phoneme conversion / M. Bisani, H. Ney // *Speech Communication*. – 2008.

96. Novak, J.R. WFST-based Grapheme-to-Phoneme Conversion : Open Source Tools for Alignment , Model-Building and Decoding / J.R. Novak, N. Minematsu, K. Hirose // *10th International Workshop on Finite State Methods and Natural Language Processing*. – 2012.

97. Yanushevskaya, I. Russian / I. Yanushevskaya, D. Bunčić // *Journal of the International Phonetic Association*. – 2015.

98. Sutskever, I. Sequence to sequence learning with neural networks / I. Sutskever, O. Vinyals, Q. V. Le // *Advances in Neural Information Processing Systems*. – 2014.

99. Toshniwal, S. Jointly learning to align and convert graphemes to phonemes with neural attention models / S. Toshniwal, K. Livescu // *2016 IEEE Workshop on Spoken Language Technology, SLT 2016 - Proceedings*. – 2017.

100. Mohri, M. Speech Recognition with Weighted Finite-State Transducers / M. Mohri, F. Pereira, M. Riley // *Springer Handbooks*. – 2008.

101. Panayotov, V. Decoding graph construction in Kaldi: A visual walkthrough [Электронный ресурс]. – URL: <http://vpanayotov.blogspot.com/2012/06/kaldi-decoding-graph-construction.html> (дата обращения: 20.05.2017).

102. Пикалёв, Я.С. О системах проверки правописания русского языка / Я.С. Пикалёв, А.С. Вовнянко // *Донецкие чтения 2018: образование, наука, инновации, культура и вызовы современности: Материалы III Международной научной конференции (Донецк, 25 октября 2018 г.)*. – Том 1: Физико-математические и технические науки/ под общей редакцией проф. С. В. Беспало. – Донецк: ДонНУ, 2018. – С. 243-247.

103. Ермоленко, Т.В. Классификация ошибок в тексте на основе глубокого обучения / Т.В. Ермоленко // *Проблемы искусственного интеллекта*. – 2019. – Т. 3. – № 14. – С. 47-57.

104. Smith T.F. Comparison of biosequences / T.F. Smith, M.S. Waterman //

Advances in Applied Mathematics. – 1981. – Т. 2. – № 4.

105. Шмырёв, Н.В. Свободные речевые базы данных voxforge.org / Н.В. Шмырёв // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Бекасово, 4–8 июня 2008 г.). – 2008. – № 7(14). – С. 585-588.

106. Ко, Т. Audio augmentation for speech recognition / Т. Ко, V. Peddinti, D. Povey, S. Khudanpur // Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. – 2015. – Тт. 2015-Janua.

107. Park, D.S. Specaugment: A simple data augmentation method for automatic speech recognition / D.S. Park, W. Chan, Y. Zhang, C.C. Chiu, B. Zoph, E.D. Cubuk, Q. V. Le // Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. – 2019. – Тт. 2019-Septe.

108. Пикалёв, Я.С. Применение аугментации для задачи автоматического распознавания речи / Я.С. Пикалёв, Т.В. Ермоленко // Материалы конференции Донецкие чтения 2019: образование, наука, инновации, культура и вызовы современности. Том 1 Физико-математические и технические науки. Часть 2, под общей редакцией проф. С.В. Беспаловой. – Донецк: ДонНУ, 2019. – С. 259-262.

109. Ко, Т. A study on data augmentation of reverberant speech for robust speech recognition / Т. Ко, V. Peddinti, D. Povey, M.L. Seltzer, S. Khudanpur // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2017.

110. Syrdal, A. TD-PSOLA versus harmonic plus noise model in diphone based speech synthesis / A. Syrdal, Y. Stylianou, L. Garrison, A. Conkie, J. Schroeter // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 1998. – Т. 1.

111. Пикалёв, Я.С. Применение систем синтеза речи / Я.С. Пикалёв // Материалы VII Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ-2016). – Донецк: ДонНТУ, 2016. – С. 136-142.

112. Пикалёв, Я.С. Применение систем синтеза речи / Я.С. Пикалёв // Электронные информационные системы. – 2016. – Т. 3. – № 10. – С. 51-56.



113. Laptev, A. You Do Not Need More Data: Improving End-To-End Speech Recognition by Text-To-Speech Data Augmentation [Электронный ресурс]. – URL: <https://arxiv.org/abs/2005.07157> (дата обращения: 06.06.2020).

114. Пикалёв, Я.С. Классификация текстовых документов при помощи иерархических нейросетей со свёрточным слоем / Я.С. Пикалёв // Восьмая международная конференция по когнитивной науке: Тезисы докладов. Светлогорск, 18–21 октября 2018 г. / ред. А.К. Крылов, В.Д. Соловьев. – М.: Изд-во «Институт психологии РАН», 2018. – С. 810-812.

115. TED Talks [Электронный ресурс]. – URL: <https://www.ted.com/talks/> (дата обращения: 25.05.2019).

116. Nair, V. Rectified linear units improve Restricted Boltzmann machines / V. Nair, G.E. Hinton // ICML 2010 - Proceedings, 27th International Conference on Machine Learning. – 2010.

117. Jurafsky, D. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (University of Colorado, Boulder) Upper Saddle River, NJ: Prentice Hall (Prentice / D. Jurafsky, J.H. Martin // Computational Linguistics. – 2000. – Т. 26. – № 4.

118. Carroll, J. Dependency Parsing Sandra Kübler, Ryan McDonald, and Joakim Nivre (Indiana University, Google Research, and Uppsala and Växjö Universities) Morgan & Claypool (Synthesis Lectures on Human Language Technologies, edited by Graeme Hirst, volume 2), 2009, xii+ / J. Carroll // Computational Linguistics. – 2010. – Т. 36. – № 1.

119. Syntaxnet Parsey McParseface wrapper for POS tagging and dependency parsing [Электронный ресурс]. – URL: <https://github.com/spoddtur/syntaxnet> (дата обращения: 20.09.2018).

120. UDPipe Models [Электронный ресурс]. – URL: <http://ufal.mff.cuni.cz/udpipe/models> (дата обращения: 20.09.2018).

121. Universal Dependencies [Электронный ресурс]. – URL: <https://universaldependencies.org/> (дата обращения: 25.09.2018).

122. Comparison of Treebank Statistics [Электронный ресурс]. – URL:

<https://universaldependencies.org/treebanks/ru-comparison.html> (дата обращения: 20.09.2018).

123. Пикалёв, Я.С. Разработка синтаксического анализатора русского языка на основе глубоких нейронных сетей / Я.С. Пикалёв // Донецкие чтения 2018: образование, наука, инновации, культура и вызовы современности: Материалы III Международной научной конференции (Донецк, 25 октября 2018 г.). – Том 1: Физико-математические и технические науки/ под общей редакцией проф. С. В. Беспало. – Донецк: ДонНУ, 2018. – С. 240-243.

124. Lourentzou, I. Adapting sequence to sequence models for text normalization in social media / I. Lourentzou, K. Manghnani, C.X. Zhai // Proceedings of the 13th International Conference on Web and Social Media, ICWSM 2019. – 2019.

125. Anastasyev, D.G. Improving part-of-speech tagging via multi-task learning and character-level word representations / D.G. Anastasyev, I.O. Gusev, E.M. Indenbom // *Komp'yuternaja Lingvistika i Intellektual'nye Tehnologii*. – 2018. – Тт. 2018-Май.

126. Скобликова, Е.С. Правила русской орфографии и пунктуации. Полный академический справочник / Е.С. Скобликова; ред. В.В. Лопатин. – М.: Эксмо, 2006. – 480 с.

127. Бешенкова, Е.В. Объяснительный русский орфографический словарь--справочник / Е.В. Бешенкова, Л.К. Чельцова, О.Е. Иванова; ред. Е.В. Бешенкова. – М.: АСТ-Пресс, 2018. – 592 с.

128. Белошапкова, В.А. Современный русский язык / В.А. Белошапкова, Е.А. Брызгунова, Е.А. Земская, Л.А. Мирославский, М.В. Новиков. – М.: Высшая школа, 1989. – 256 с.

129. Валгина, Н.С. Современный русский язык: Учебник / Н.С. Валгина, Д.Э. Розенталь, М.И. Фомина; ред. Н.С. Валгина. – М.: Логос, 2002. – 528 с.

130. Князев, С.В. Современный русский литературный язык: Фонетика, орфоэпия, графика и орфография: Учебное пособие для вузов / С.В. Князев, С.К. Пожарицкая. – М.: Академический Проект; Гаудеамус, 2011. – 430 с.

131. Гируцкий, А.А. Введение в языкознание / А.А. Гируцкий. – Минск: ТетраСистемс, 2003. – 288 с.

132. Грищенко, А.И. Фонетика современного русского литературного языка (Фонетика. Фонология. Орфоэпия. Графика. Орфография) / А.И. Грищенко, М.Т. Попова. – М.: Московский педагогический государственный университет, 2018. – 136 с.
133. Бондаренко, А.В. Визильтер, Ю.В. Горемычкин В.И. Формальный метод транскрипции иностранных имен собственных на русский язык / В.И. Бондаренко, А.В. Визильтер, Ю.В. Горемычкин, Э.С. Клышинский // Программные продукты и системы. – 2010. – № 1. – С. 147-152.
134. Черепанова, О.Д. Озвучивание англоязычных словоупотреблений в системе русскоязычного синтеза «текст-речь» с помощью практической транскрипции / О.Д. Черепанова // Проблемы компьютерной лингвистики и типологии: Сборник научных трудов. – Воронеж: Изд-во ВГУ, 2017. – Т. 6.
135. Суперанская, А.В. Теоретические основы практической транскрипции / А.В. Суперанская. – М.: Наука, 1978. – 283 с.
136. Гитляревский, Р.С. Иностранные имена и названия в русском тексте. Справочник. / Р.С. Гитляревский, Б.А. Старостин. – М.: Высшая школа, 1985. – 304 с.
137. Ермолович, Д.И. Имена собственные на стыке языков и культур / Д.И. Ермолович. – М.: Валент, 2001. – 200 с.
138. Ермолович, Д.И. Методика межъязыковой передачи имён собственных / Д.И. Ермолович. – М.: ВЦП, 2009. – 86 с.
139. Рыбакин, А.И. Словарь английских фамилий / А.И. Рыбакин. – М.: Астрель, 2000. – 576 с.
140. Лидин, Р.А. Иностранные фамилии и личные имена. Практика транскрипции на русский язык. Справочник / Р.А. Лидин. – М.: ООО «Издательство Толмач», 2006. – 480 с.
141. Казакова, Т.А. Практические основы перевода / Т.А. Казакова. – СПб: «Издательство Союз», 2001. – 320 с.
142. Таранов, А.М. Русско-английский тематический словарь: Для активного изучения слов и закрепления словарного запаса. Британский английский. Кириллическая транслитерация. 15000 слов / А.М. Таранов. – T&P Books Publishing, 2011. – 388 с.

143. Crochemore, M. Direct construction of compact directed acyclic word graphs / M. Crochemore, R. V erin // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). – 1997. – Т. 1264.
144. Васюкова, И.А. Словарь иностранных слов: с грамматическими формами, синонимами, примерами употребления / И.А. Васюкова. – М.: АСТ-ПРЕСС, 1999. – 631 с.
145. Захаренко, Е.Н. Новый словарь иностранных слов / Е.Н. Захаренко, Л.Н. Комарова, И.В. Нечаева. – М.: ООО ИФ «Азбуковник», 2008. – 1040 с.
146. Егорова, Т.В. Словарь транскрипций для слов-исключений. Словарь иностранных слов современного русского языка / Т.В. Егорова. – М.: Аделант, 2014. – 800 с.
147. Гребнёва, Ю. Словарь омонимов, омоформ и омографов русского языка / Ю. Гребнёва. – М.: Мир и образование, 2019. – 656 с.
148. Тарасова, Л. Школьный словарь омографов. Кто? Что? / Л. Тарасова. – М.: 5 за знания, 2018. – 72 с.
149. Пикалёв, Я.С. Система автоматической генерации транскрипций русскоязычных слов-исключений на основе глубокого обучения / Я.С. Пикалёв, Т.В. Ермоленко // Проблемы искусственного интеллекта. – 2019. – Т. 4. – № 15. – С. 35-51.
150. Toolkit N.L. Stemmers [Электронный ресурс]. – URL: <http://www.nltk.org/howto/stem.html> (дата обращения: 15.05.2018).
151. Porter, M.F. Snowball: A language for stemming algorithms [Электронный ресурс]. – URL: <https://pdfs.semanticscholar.org/0d8f/907bb0180912d1e1df279739e45dff6853ee.pdf> (дата обращения: 20.05.2019).
152. Pascanu, R. On the difficulty of training recurrent neural networks / R. Pascanu, T. Mikolov, Y. Bengio // 30th International Conference on Machine Learning, ICML 2013. – 2013.
153. Черепанова, О.Д. Лингвистическое обеспечение речевых технологий: использование англо-русской практической транскрипции в системе русскоязычного синтеза «Текст - речь» / О.Д. Черепанова // Вестник Московского

университета. Серия 9. Филология. – 2017. – № 3. – С. 156-167.

154. Hermjakob, U. Name translation in statistical machine translation learning when to transliterate / U. Hermjakob, K. Knight, H. Dauré // ACL-08: HLT - 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference. – 2008.

155. Мещеряков, Р.В. Речевые технологии в задаче обучения студентов-носителей русского языка произношению на иностранном языке. / Р.В. Мещеряков, С.Д. Тиунов, Ю.М. Лирмак, Ш.А. Е. // «Анализ разговорной русской речи» (АРЗ- 2011): Труды пятого междисциплинарного семинара. – СПб: ГУАП, 2011. – С. 78-82.

156. Успенский, В.А. Труды по нематематике. В 5 книгах. Книга 3. Языкознание / В.А. Успенский. – М.: ОГИ, 2013. – 712 с.

157. Буркова, С.С. Англицизмы в современном русском интернет-языке / С.С. Буркова, А.И. Дергабузов // Актуальные проблемы филологии: материалы III Международной научной конференции (г. Казань, май 2018 г.). – Казань: Молодой ученый, 2018. – С. 11-13.

158. Vaswani, A. Tensor2tensor for neural machine translation / A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A.N. Gomez, S. Gouws, L. Jones, Ł. Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer, J. Uszkoreit // AMTA 2018 - 13th Conference of the Association for Machine Translation in the Americas, Proceedings. – 2018. – Т. 1.

159. Popel, M. Training Tips for the Transformer Model / M. Popel, O. Bojar // The Prague Bulletin of Mathematical Linguistics. – 2018. – Т. 110. – № 1.

160. Deep speech 2: End-to-end speech recognition in English and Mandarin / D. Amodei и др. // Proc. 33rd International Conference on Machine Learning, ICML 2016. – 2016.

161. Марковников, Н.М. Исследование методов построения моделей кодер-декодер для распознавания русской речи / Н.М. Марковников, И.С. Кипяткова // Информационно-управляющие системы. – 2019. – № 4 (101). – С. 44-53.

162. Пикалёв, Я.С. Исследование программного комплекса распознавания речи Kaldi ASR / Я.С. Пикалёв, В.Ю. Шелепов // Донецкие чтения 2017: Русский мир как цивилизационная основа научно-образовательного и культурного развития

Донбасса: Материалы Международной научной конференции студентов и молодых ученых (Донецк, 17-20 октября 2017 г.). – Том 1: Физико-математические и . – Донецк: ДонНУ, 2017. – С. 232-234.

163. Пикалёв, Я.С. Разработка системы автоматического распознавания слитной русскоязычной речи на основе дискриминативного обучения / Я.С. Пикалёв // Информатика и кибернетика. – 2018. – Т. 3. – № 13. – С. 61-68.

164. Chen, G. Pronunciation and silence probability modeling for ASR / G. Chen, H. Xu, M. Wu, D. Povey, S. Khudanpur // Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. – 2015. – Тт. 2015-Janua.

165. Меденников, И.П. Двухэтапный алгоритм инициализации обучения акустических моделей на основе глубоких нейронных сетей / И.П. Меденников // Научно-технический вестник информационных технологий, механики и оптики. – 2016. – Т. 16. – № 2. – С. 379-381.

166. Xue, J. Restructuring of deep neural network acoustic models with singular value decomposition / J. Xue, J. Li, Y. Gong // Interspeech. – 2013. – С. 2365-2369.

167. He, K. Deep residual learning for image recognition / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – 2016. – Тт. 2016-Decem.

168. Xu, K. Show, attend and tell: Neural image caption generation with visual attention / K. Xu, J.L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R.S. Zemel, Y. Bengio // 32nd International Conference on Machine Learning, ICML 2015. – 2015. – Т. 3.

169. Peddinti, V. A time delay neural network architecture for efficient modeling of long temporal contexts / V. Peddinti, D. Povey, S. Khudanpur // Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. – 2015. – Тт. 2015-Janua.

170. Graves, A. Hybrid speech recognition with Deep Bidirectional LSTM / A. Graves, N. Jaitly, A.R. Mohamed // 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013 - Proceedings. – 2013.

171. Chung, E. Lattice Rescoring for Speech Recognition using Large Scale

Distributed Language Models / E. Chung, H.-B. Jeon, J.-G. Park, Y.-K. Lee // COLING. – Mumbai: The COLING 2012 Organizing Committee, 2012. – С. 217-224.

172. OpenFst Library [Электронный ресурс]. – URL: <http://www.openfst.org/twiki/bin/view/FST/WebHome> (дата обращения: 15.06.2020).

173. Povey, D. The Kaldi speech recognition toolkit / D. Povey, and others // Proc. ASRU. – 2011.

174. Abadi, M. TensorFlow: A system for large-scale machine learning / M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng // Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016. – 2016.

175. Man, B. De. Evaluation of implementations of the EBU R128 loudness measurement / B. De Man // 145th Audio Engineering Society International Convention, AES 2018. – 2018.

176. Karpov, A. Large vocabulary Russian speech recognition using syntactico-statistical language modeling / A. Karpov, K. Markov, I. Kipyatkova, D. Vazhenina, A. Ronzhin // Speech Communication. – 2014. – Т. 56. – № 1.

177. ARPA Language models [Электронный ресурс]. – URL: <https://cmusphinx.github.io/wiki/arpaformat/>.

178. Verwimp, L. TF-LM: Tensorflow-based language modeling toolkit / L. Verwimp, H. Van Hamme, P. Wambacq // LREC 2018 - 11th International Conference on Language Resources and Evaluation. – 2019.

179. Decoders used in the Kaldi toolkit [Электронный ресурс]. – URL: <https://kaldi-asr.org/doc/decoders.html> (дата обращения: 27.08.2020).

180. Provilkov, I. Вре-dropout: Simple and effective subword regularization [Электронный ресурс]. – URL: <https://arxiv.org/abs/1910.13267>.

181. Text Normalization Challenge - Russian Language [Электронный ресурс]. – URL: <https://www.kaggle.com/c/text-normalization-challenge-russian-language>.

182. tf.data.Dataset [Электронный ресурс]. – URL: [https://www.tensorflow.org/api\\_docs/python/tf/data/Dataset](https://www.tensorflow.org/api_docs/python/tf/data/Dataset).

183. Lafferty, J. Conditional random fields: Probabilistic models for segmenting and labeling sequence data / J. Lafferty, A. McCallum, F.C.N. Pereira // ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning. – 2001. – Т. 8. – № June.

184. Misra, D. Mish: A Self Regularized Non-Monotonic Neural Activation Function [Электронный ресурс]. – URL: <https://arxiv.org/abs/1908.08681>.

185. Облако ЦРТ - технологии синтеза и распознавания речи [Электронный ресурс]. – URL: <https://cp.speechpro.com/service/asr>.

186. Speech-to-Text: Automatic Speech Recognition [Электронный ресурс]. – URL: <https://cloud.google.com/speech-to-text>.



## ПРИЛОЖЕНИЕ А. БАРК- И МЕЛ-ШКАЛЫ

Таблица А.1 Барк-шкала

№ полосы	Критическая полоса (диапаз.), Гц	Ширина критической полосы, Гц	Центральная частота критической полосы, Гц
0	0-00	100	50
1	100-200	100	150
2	200 – 300	100	250
3	300 – 400	100	350
4	400-510	110	450
5	510-630	120	570
6	630 – 770	140	700
7	770 – 920	150	840
8	920 – 1080	160	1000
9	1080-1270	190	1170
10	1270-1480	210	1370
11	1480-1720	240	1600
12	1720 – 2000	280	1850
13	2000-2310	320	2150
14	2320 – 2700	380	2500
15	2700-3150	450	2900
16	3150-3700	550	3400
17	3700-4400	700	4000
18	4400 – 5300	900	4800
19	5300 – 6400	1100	5800
20	6400 – 7700	1300	7000
21	7700 – 9500	1800	8500
22	9500-12 000	2500	10500
23	12 000-15 500	3500	13500

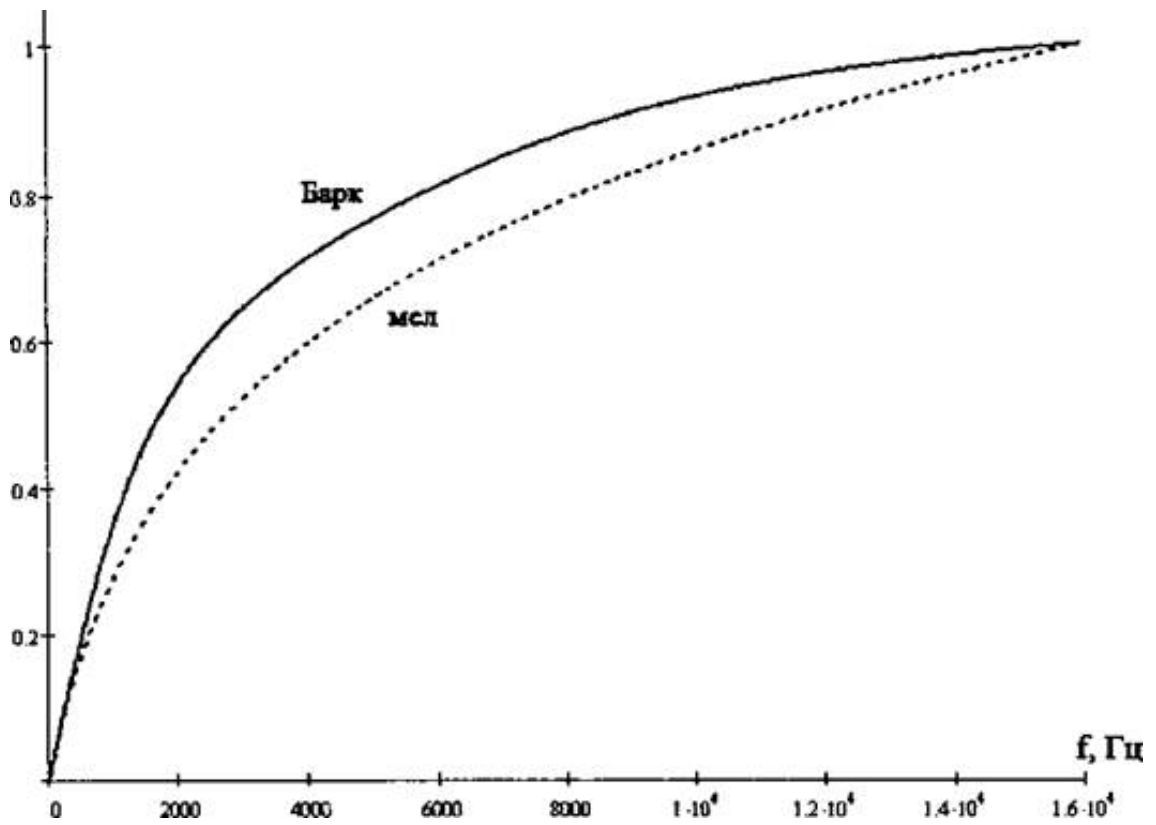


Рисунок А.1 – Соответствие Барков и Мелов Герцам

## ПРИЛОЖЕНИЕ Б. Схема обучения акустических моделей

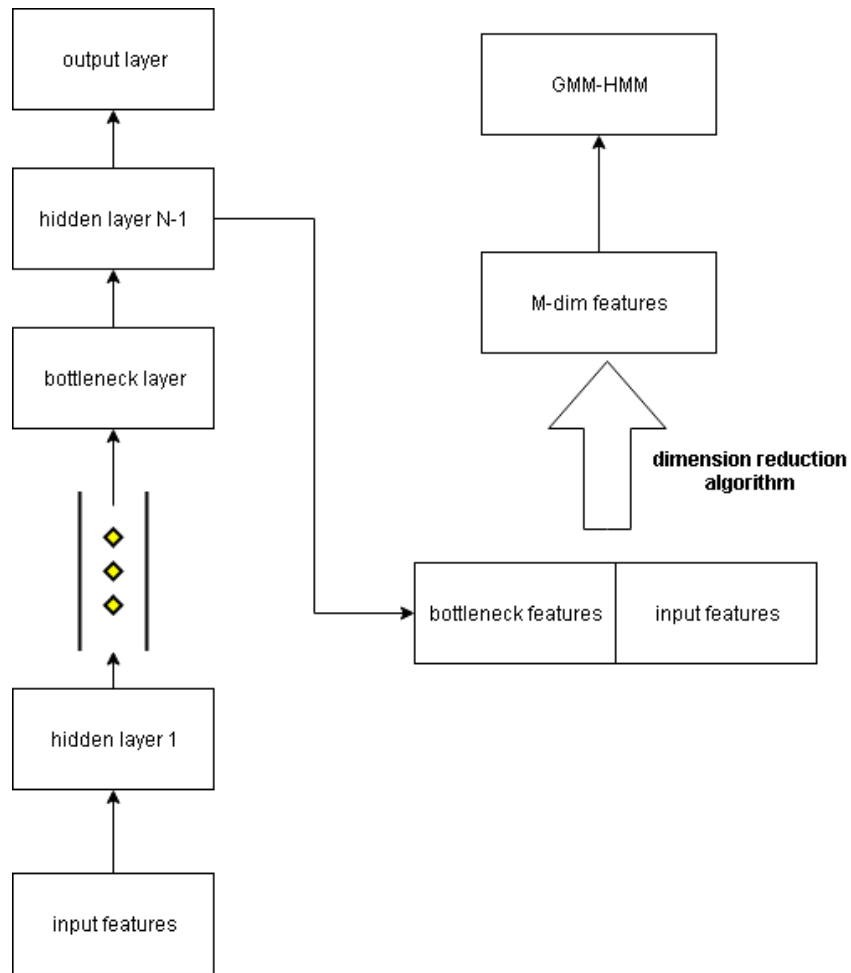


Рисунок Б.1 – Схема обучения акустических моделей на основе совмещённых признаков (M-dim features)



ПРИЛОЖЕНИЕ В. Документы, подтверждающие внедрение результатов  
диссертации



ДОНЕЦКАЯ НАРОДНАЯ РЕСПУБЛИКА  
ГОСУДАРСТВЕННЫЙ КОМИТЕТ ПО НАУКЕ И ТЕХНОЛОГИЯМ  
ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ  
«ИНСТИТУТ ПРОБЛЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»

83048, г. Донецк, ул. Артёма, 118-б, тел./факс: (062) 311-69-50, ☎ (062) 311-34-24, http://guiaidn.ru, e-mail: gu\_ipi@mail.ru

01.12.2020 г. № 347/01-01

Диссертационный совет Д 01.024.04

СПРАВКА

о внедрении результатов исследования диссертационной работы  
Пикалёва Ярослава Сергеевича на тему «Совершенствование методов и  
программных средств распознавания слитной русской речи»,  
представленную на соискание ученой степени кандидата технических наук  
по специальности 05.13.01 – Системный анализ, управление и обработка  
информации (технические науки)

Результаты, полученные при выполнении диссертационного  
исследования Пикалёва Я.С., использовались при выполнении  
фундаментальной научно-исследовательской работы «Исследование и  
разработка методов семантического анализа и интерпретации потоков  
данных интеллектуальными системами» (0118D000003) при разработке  
модулей:

- автоматической разметки дикторов в аудиосигнале;
- голосовой активности;
- нормализации текста;
- создания языковой модели/

Директор ГУ «Институт проблем  
искусственного интеллекта»



С.Б. Иванова

