Казахский национальный исследовательский технический университет имени К. И. Сатпаева
Институт информационных и вычислительных технологий КН МНВО РК

УДК 004.934

На правах рукописи

ОРАЛБЕКОВА ДИНА ОРЫМБАЕВНА

Разработка системы автоматического распознавания речи на основе интегрального подхода

8D06103 – Management information systems

Диссертация на соискание ученой степени
доктора философии (PhD)

Научные консультанты:
Мамырбаев О.Ж.
PhD, ассоциированный профессор
Othman Mohamed,
PhD, профессор
(Малайзия)

Республика Казахстан
Алматы, 2022

СОДЕРЖАНИЕ

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей диссертации использованы ссылки на следующие стандарты:

ГОСО РК 5.04.034-2011 «Государственный общеобязательный стандарт образования Республики Казахстан. Послевузовское образование. Докторантура. Основные положения» , утвержденный приказом Министра образования и науки Республики Казахстан от 17 июня 2011 года № 261;

Положение о диссертационном совете НАО «КазНИТУ имени К.И.Сатпаева». П 029-04-01.01 – 2021;

«Инструкция по оформлению диссертации и автореферата», МОН РК, Внешний аттестационный комитет. Алматы 2004.

ГОСТ 7.1-2003. Библиографическая запись.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

APP – автоматическое распознавание речи

CAPP – система автоматического распознавания речи

HMM – скрытые Марковские модели (Hidden Markov Model)

DNN – глубокие нейронные сети (Deep neural networks)

GMM – модель Гауссовых смесей (Gaussian Mixture Model)

GMM-HMM – модели гауссовых смесей, базированных на скрытых Марковских моделях

RNN – рекуррентные нейронные сети (Recurrent Neural Networks)

LSTM – сеть с долгой и кратковременной памятью (Long Short-Term Memory)

ЯМ – языковая модель

DTW – динамическое преобразование времени (Dynamic Time Warping)

MFCC – мел-частотные кепстральные коэффициенты (Mel-Frequency Cepstral Coefficients)

PLP – перцептивное линейное предсказание (Perceptual Linear Prediction)

CNN – сверточная нейронная сеть (Convolutional Neural Network)

АМ – акустическая модель

ИНС – искусственные нейронные сети

E2E – интегральное (end-to-end)

SGD – стохастический градиентный спуск (Stochastic Gradient Descent)

CTC – коннекционная временная классификация (Connectionist Temporal Classification)

GPU – графический процессор (Graphics Processing Unit)

CRF – условные случайные поля (Conditional Random Fields)

WER – коэффициент неверно распознанных слов (Word Error Rate)

CER – коэффициент неверно распознанных символов (Character Error Rate)

BLSTM – двунаправленный LSTM (Bidirectional LSTM)

TDNN – глубокая нейронная сеть с задержкой во времени (Time Delay Neural Network)

RNN-T – рекуррентный нейронный преобразователь (Recurrent Neural Network Transducer)

LER – коэффициент неверно распознанных меток (Label Error Rate)

LAS – Listen, Attend and Spell

RNA – Recurrent Neural Aligner

WSJ – Wall Street Journal

Seq2Seq – Sequence to Sequence

LC – управление задержкой (Latency-Controlled)

AMoChA –адаптивное монотонное групповое внимание (Adaptive Monotonic Chunk-wise Attention)

MEMM – Марковские модели с максимальной энтропией (Maximum-Entropy Markov Model)

SCRF – сегментные условные случайные поля (Segmental CRF)

ВВЕДЕНИЕ

Актуальность темы исследования. Прогресс информационных систем и вычислительной техники привел к улучшению процессов во многих технологиях машинного обучения, в частности в распознавании речи. Распознавание речи относится к технологии, которая позволяет машинам с помощью различных систем, программ и алгоритмов распознавать и понимать человеческую речь и преобразовывать ее в текст. Системы автоматического распознавания речи (САРР) нашли широкое применение в различных сферах деятельности, как голосовое управление автомобилем, домом и бытовой техникой, а также голосовой ввод в различных приложениях, навигационных системах и сервисах для людей с ограниченными возможностями. И это лишь некоторые примеры использования САРР. Существует традиционная система распознавания речи, которая обычно состоит из трех основных независимых элементов, и представляют из себя следующие модели, как акустические модели для прогнозирования контекстно-зависимых состояний субфонем из аудио, языковые модели и лексикон для сопоставления фонем к словам [1]. Такие системы состоят из различных элементов, которые обучаются независимо друг от друга, так классическая акустическая модель может быть обучена на основе смесей гауссовых распределений и скрытых марковских моделей, а языковые модели на основе n-gram. Долгое время в задаче распознавания речи широко применялась, и была основной технологией, модель на основе скрытых марковских моделей (HMM) [2]. HMM в основном используется для динамической деформации времени на уровне кадра и смеси гауссовских распределений плотностей вероятностей (GMM) применяется для представления распределений сигналов в промежутке фиксированного небольшого периода времени, который обычно соответствует единице произношения [3]. Продолжительное время модель HMM-GMM являлась общей структурой для распознавания речи. В последнее время глубокое обучение приносит значительные улучшения во многих исследованиях, и в развитие распознавания речи. Активное использование искусственных нейронных сетей на каждом элементе сценария классической системы распознавания речи увеличивает эффективность ее работы, что отразилось во многих исследовательских работах. С развитием технологий глубокого обучения глубоких нейронных сетей (DNN) начали применять в распознавании речи для акустического моделирования [4]. Роль DNN заключается в рассчитывании апостериорной вероятности состояний HMM, заменяя обычную вероятность наблюдения GMM [5]. Следовательно, модель HMM со смесью гауссовских распределений превращается в HMM с глубокой сетью, которая достигает конкурентноспособных результатов, и становится популярным подходом автоматического распознавания речи. В научной работе [6] было показано, что для получения эффективной акустической модели были прменены глубокие нейронные сети, а в [7, 8] с помощью рекуррентных нейронных сетей и сетей с долгой и кратковременной памятью (LSTM) были построены языковые модели и словарь соответственно. Также в

работе [9] сверточные нейронные сети (CNN) были применены для извлечения признаков из сигнала речи. Таким образом, многие опубликованные результаты показывают, что предложенный подход демонстрирует лучшую производительность среди всех современных систем распознавания речи, который является основой для применения различных архитектур искусственных нейронных сетей на всех стадиях систем распознавания речи.

В последнее время распространение получил интегральный метод распознавания речи с использованием методов машинного обучения. В таких системах модель реализована с использованием только одной нейронной сети. Интегральная реализация модели часто представляет лучшую производительность с позиции скорости и точности распознавания речи. Во многих исследовательских работах доказано, что прогресс полученных результатов интегральной системы зависит от увеличения объема тренировочных данных для обучения сети. В настоящее время популярные приложения, как Voice to Text Messenger, Google Listen, Attend, Spell, Baidu Deep Speech и другие работают на основе интегрального подхода. Основной принцип работы заключается в том, что современные интегральные модели обучаются на основе больших данных. Из вышеуказанного можно обнаружить основную проблему, это касается распознавания языков с ограниченными обучающими данными, таких как казахский, киргизский, турецкий и т.д. Для таких малоресурсных языков не существует большие корпусы обучающих данных. Также необходимо отметить, что не разработаны и не исследованы модели и методы интегральной архитектуры для малоресурсных языков. В настоящее время не существуют эффективных алгоритмов и программных средств интегрального распознавания для казахского языка.

Зарубежные ученые, как Dario Amodei, Chao Weng, William Chan (США), Xinhui Hu, Chiori Hori (Япония), Jinchuan Tian, Eric Chang, Jianlai Zhou, Jianwei Sun (Китай), Jan Chorowski (Польша) и другие исследователи добились высоких результатов в совершенстовании интегральных систем для распознавания популярных языков, как английский и китайский, и ученые из ближнего СНГ, а именно из Санкт-Петербургского института информатики и автоматизации Российской академии наук Карпов А.А., Кипяткова И.С. и их коллеги исследуют область распознавания речи уже многие годы и достигли хороших показателей в разработке и улучшении интегральных систем распознавания русской речи [10-14].

Но стоит отметить, что существуют разработки отечественных ученых по системам распознавания казахской речи на основе глубоких нейронных сетей. В работах ученых Евразийского национального университета им. Л.Н.Гумилева – Шарипбай А.А., Есенбаева Ж.А. [15-16], Казахского национального университета им. Аль-Фараби – Тукеева У.А., Рахимовой Д.Р. [17-18], Назарбаев университета – Хасанова Е. [19], а также в научных работах исследователей из Института информационных и вычислительных технологий – Амиргалиева Е.Н., Мусабаева Р.Р. [20-21] и др. были отражены разработки системы автоматического распознавания казахской речи на основе традиционной,

гибридных моделей HMM-DNN и интегральной модели CTC с Transformer. Результаты этих работ достигли хорошего уровня верного распознавания речи, но до сих пор данные системы требуют улучшений в сокращении ошибок распознавания слов до человеческого уровня и в увеличении объема данных для тренировки сети. Кроме того, не разработаны интегральные системы для распознавания казахской речи на основе модели кодер-декодера (Encoder-Decoder) с использованием механизма внимания, которая показывает перспективные результаты. Модель с механизмом внимания может хорошо работать как с языковыми моделями, так и без них, при этом демонстрируя низкую погрешность распознавания речи. Этот подход является перспективным направлением, который может быть использован для разработки систем распознавания речи при ограниченной обучающей выборки.

Исходя из вышеизложенного, можно прийти к выводу, что в настоящий момент необходимость в эффективных методах, алгоритмах и программном обеспечении для улучшения точности распознавания казахской речи с использованием интегральных моделей особенно *актуальна*.

Цель диссертационной работы. Разработка модели, метода и алгоритма для повышения точности распознавания казахской речи на основе интегральной архитектуры.

Задачи исследования. Для реализации поставленных целей исследования решаются следующие вопросы:

1) Анализ популярных методов распознавания речи на основе интегральных моделей.

2) Разработка речевого корпуса для интегрального распознавания казахской речи.

3) Разработка эффективного метода и алгоритма интегральной модели на основе кодер-декодера (Encoder-Decoder) с использованием механизма внимания для создания системы распознавания казахской речи.

4) Разработка интегральной архитектуры и программного обеспечения для распознавания казахской речи с помощью полученных в ходе исследований моделей и методов на основе интегрального подхода.

Объект исследования. Современные технологии и системы автоматического распознавания речи.

Предмет исследования. Технологии, алгоритмы, модели и методы, программное обеспечение для распознавания казахской речи на основе интегрального подхода.

Методы исследования. Методы машинного обучения, технологии и методы автоматического распознавания речи, теории вероятностей и математической статистики, методы разработки программного обеспечения, а также прикладная и компьютерная лингвистика.

Новизна исследовательской работы.

1) Разработан новый обучающий корпус для казахского языка.

2) Разработана интегральная модель с применением механизма внимания для распознавания казахской речи.

3) Разработаны метод и алгоритм для эффективного распознавания казахской речи на основе интегрального модуля.

Теоретическое и практическое значение работы. Теоретическая значимость исследовательской работы заключается в разработке и реализации методов и алгоритмов интегральных моделей для распознавания казахской речи, а также в разработке обучающего корпуса речи для казахского языка.

Практическая значимость исследовательской работы заключается в применении разработанных алгоритмов и программного обеспечения для дальнейшего использования в развитии других технологий, как синтез речи, машинный перевод, голосовая аутентификация и идентификация и т.д. Разработанная система автоматического распознавания казахской речи может быть внедрено в государственных структурах, ответственных за расширение области применения национальных языков на базе информационных технологий; в мобильных телефонах (увеличение числа потенциальных покупателей за счёт внедрения речевых технологий на национальном языке); в банках (call-центры с поддержкой голосовых функций, голосовая аутентификация); в сектор производства различных устройств с поддержкой голосовых функций.

Основное положение, выносимое на защиту. Для обучения сети был разработан корпус в объеме 2000 часов речи с транскрипциями. При создании корпуса учтены различные виды речи: подготовленная (чтение), спонтанная.

Были разработаны методы и модели интегральной архитектуры кодер-декодера с вниманием. С использованием нейронных сетей, как LSTM и BLSTM была создана архитектура данной модели. На основе проведенных экспериментов было выявлено, что модель с вниманием хорошо выполняет работу без применения языковых моделей для государственного языка и превзошла не только гибридные модели с HMM, но и другие разновидности интегральных моделей, обученные меньшим объемом текстовых и речевых данных и продемонстрировала высокие показатели распознавания речи на казахском по точности распознавания символов и слов.

Степень достоверности и апробация результатов. Исследования и результаты, относящиеся к теме диссертации, были представлены и обсуждены на различных конференциях и семинарах на основе следующих публикаций:

1) Мамырбаев О.Ж., **Оралбекова Д.О.** Онлайновые модели на основе внимания для интегральных (end-to-end) систем распознавания речи. V Международная научно-практическая конференция "Информатика и прикладная математика (29 сентября - 1 октября 2020 г., Алматы).

2) O. Mamyrbayev, **D. Oralbekova**, A. Kydyrbekova, T. Turdalykyzy and A. Bekarystankyzy, "End-to-End Model Based on RNN-T for Kazakh Speech Recognition," 3rd International Conference on Computer Communication and the Internet (ICCCI) (25-27 июня 2021 г., Токио).

3) Мамырбаев О.Ж., **Оралбекова Д.О.**, Othman M., Тулендиев Д.М., Жумажанов Б., Турдалыкызы Т. Распознавание казахской речи на основе интегральной модели RNN-T. VI Международная научно-практическая

конференция "Информатика и прикладная математика (29 сентября - 1 октября 2021 г., Алматы).

4) Мамырбаев О.Ж., **Оралбекова Д.О.**, Othman M., Тулендиев Д.М., Жумажанов Б., Турдалыкызы Т. Исследование интегральной модели на основе внимания для автоматического распознавания казахской речи. Международная научная конференция в области информационных технологий, посвященной 75-летию профессора У.А. Тукеева. (8 октября 2021 г., Алматы).

5) Мамырбаев О.Ж., **Оралбекова Д.О.**, Кыдырбекова А.С., Жумажанов Б.Ж. , Тұрдалықызы Т. Интегральная гибридная модель на основе CTC и механизма внимания для распознавания казахской слитной речи. Международная научная конференция «Сатпаевские чтения 2021» (12 апреля 2021 г., Алматы).

6) **D. Oralbekova**, O. Mamyrbayev, M. Othman, B. Zhumazhanov, K. Mukhsina. Development of the insertion-based speech recognition method. 7th International conference on Computer Science and Engineering (14-16 сентября **2022г., Стамбул, Турция**)

Личный вклад исследователя. Докторант самостоятельно выполнил и решил задачи диссертационной работы. Разработал и реализовал модель и алгоритм для распознавания казахской речи на основе интегральной архитектуры. Разработал и расширил корпус для казахского языка. Выполнил экспериментальную оценку разработанных моделей и алгоритмов.

Связь темы диссертации с планами научно-исследовательской работы. Проведенные научно-исследовательские работы по диссертации были выполнены в рамках двух проектов грантового финансирования: 1) «Разработка технологии мультиязычного автоматического распознавания речи с использованием глубоких нейронных сетей» (2018-2020, государственный регистрационный номер: 0118РК00139) 2) «Разработка интегральной (end-to-end) системы автоматического распознавания речи для агглютинативных языков» (2020-2022, государственный регистрационный номер: 0120РК00344) в Институте информационных и вычислительных технологий КН МНВО РК.

Публикация основных результатов диссертационного исследования. По теме диссертационной работы было получено 3 авторских свидетельства, 1 патент на изобретение и опубликовано 7 работ, из которых 3 статьи опубликованы в журналах, рекомендованных Комитетом по контролю в сфере образования и науки МОН РК, 4 статьи опубликованы в изданиях имеющие ненулевой импакт-фактор, индексирумых базой Scopus и Web of Science:

1) Mamyrbayev, O., **Oralbekova, D.,** Keylan, A. et al. A study of transformer-based end-to-end speech recognition system for Kazakh language. Sci Rep 12, 8337 (2022). https://doi.org/10.1038/s41598-022-12260-y (**Web of Science, квартиль Q1, IF=4,3**)

2) Mamyrbayev, O.Z., Oralbekova, D.O., Alimhan, K. et al. Hybrid end-to-end model for Kazakh speech recognition. International Journal of Speech Technology (2022). https://doi.org/10.1007/s10772-022-09983-8 (Scopus, IF 1.803, процентиль 93).

3) Mamyrbayev, O., Kydyrbekova, A., Alimhan, K., **Oralbekova, D.,** Zhumazhanov, B., Nuranbayeva, B. (2021). Development of security systems using DNN and i & x-vector classifiers. Eastern-European Journal of Enterprise Technologies, 4 (9 (112)), 32–45. doi: https://doi.org/10.15587/1729-4061.2021.239186 (**Scopus, процентиль 43**);

4) Mamyrbayev, O., Alimhan, K., **Oralbekova, D.,** Bekarystankyzy A., Zhumazhanov, B. (2022). Identifying the influence of transfer learning method in developing an end-to-end automatic speech recognition system with a low data level. Eastern-European Journal of Enterprise Technologies, 1(9(115), 84–92. https://doi.org/10.15587/1729-4061.2022.252801(**Scopus, процентиль 43**);

5) Mamyrbayev O., **Oralbekova D.** Modern trends in the development of speech recognition systems // News of the National academy of sciences of the republic of Kazakhstan. – 2020. – Vol. 4, № 332. - P. 42 – 51 // doi.org/10.32014/2020.2518-1726.64

6) Mamyrbayev O., **Oralbekova D.**, Alimhan K., Othman M., Zhumazhanov B. Realization of online systems for automatic speech recognition// News of the National academy of sciences of the republic of Kazakhstan. – 2021. – Vol. 6, № 340. - P. 66 – 72 // doi.org/10.32014/2021.2518-1726.103

7) Мамырбаев О.Ж., **Оралбекова Д.О.**, Алимхан К., Othman M., Жумажанов Б. Применение гибридной интегральной модели для распознавания казахской речи// News of the National academy of sciences of the republic of Kazakhstan. – 2022. – Vol. 1, № 341. - P. 58 – 68 // doi.org/10.32014/2022.2518-1726.117.

8) Авторское свидетельство "Система автоматического распознавания казахской речи на основе интегральной архитектуры" № 15501 от 25.02.2021, Авторы: О.Ж. Мамырбаев, **Д.О. Оралбекова**, А.С. Кыдырбекова, Б.Ж. Жумажанов, Т.Тұрдалықызы.

9) Авторское свидетельство "Система идентификации и аутентификации через речевые технологии" № 23323 от 4 февраля 2022. Авторы: **Оралбекова Д.О.**, Мамырбаев О.Ж., Алимхан К., Кыдырбекова А.С., Жумажанов Б.Ж., Турдалыкызы Т.

10) Авторское свидетельство "Система автоматического распознавания казахской слитной речи на основе модели с механизмом внимания" №24178 от 5.03.2022. Авторы: Мамырбаев О.Ж., **Оралбекова Д.О.**, Әлімхан Қ., Кыдырбекова А.С., Жұмажанов Б.Ж., Тұрдалықызы Т.

11) Патент на изобретение «Система и способ распознавания агглютинативной слитной речи на основе интегрального (end-to-end) подхода». № 35886 от 07.10.2022. Авторы: Мамырбаев О.Ж., **Оралбекова Д.О.**, Кыдырбекова А.С., Жұмажанов Б.Ж., Тұрдалықызы Т.

Структура и объем диссертационной работы. Диссертационная исследовательская работа состоит из введения, 4 разделов, заключения, списка литературы из 111 наименований и 5 приложений. Работа изложена на 108 страницах и содержит 25 рисунков, 6 таблиц.

1 СОВРЕМЕННЫЕ МЕТОДЫ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ РЕЧИ

1.1 Обзор современных технологий распознавания речи

Распознавание речи – это способность машины или программы распознавать слова и фразы на разговорном языке и преобразовывать их в машиночитаемый формат, т.е. в текст. (рис. 1.1) Автоматическое распознавание речи (АРР) в настоящее время находит широкое применение в повседневной жизни. Сегодня АРР быстро стал повсеместным как полезный способ взаимодействия с технологиями, значительно сокращая разрыв во взаимодействии человека и компьютера, делая его более естественным. АРР используется в таких областях как, автоматизированный пользовательский интерфейс, управление цифровыми устройствами, интеллектуальные системы, цифровые персональные помощники и т.д. [22].

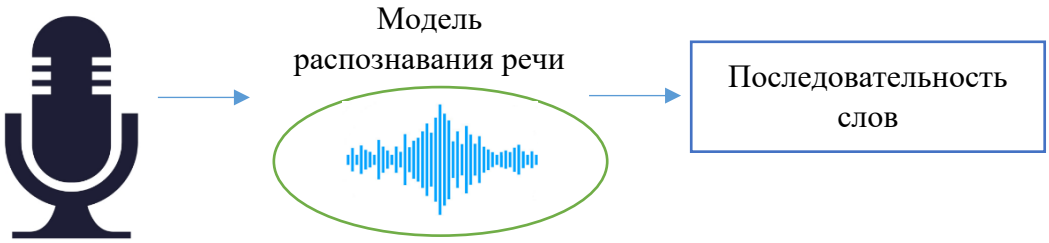


Рисунок 1.1 – Задача АРР - преобразование речевого сигнала в текст

Проще говоря, АРР можно описать следующим образом: учитывая входные аудиовыборки X из записанного речевого сигнала, примените функцию f, чтобы сопоставить его с последовательностью слов W, которые представляют расшифровку того, что было произнесено (1).

$$W = f(X) \tag{1}$$

Однако найти такую функцию $f(X)$ довольно сложно и требует последовательных задач моделирования для создания последовательности слов.

Эти модели должны быть устойчивыми к различным динамикам, акустической среде и контексту. Например, человеческая речь может иметь любую комбинацию изменений во времени (скорость говорящего), артикуляции, произношения, громкости говорящего и вокальных вариаций (хриплая или гнусавая речь) и все равно приводит к той же транскрипции.

С лингвистической точки зрения встречаются дополнительные переменные, такие как просодия (повышение интонации при задании вопроса), манеры, спонтанная речь, также известные как слова-заполнители («ммм» или «эээ»), все они могут означать разные эмоции или значения, даже если произносятся те же слова. Комбинирование этих переменных с любым количеством сценариев окружающей среды, таких как качество звука, расстояние до микрофона,

фоновый шум, реверберация и эхо, экспоненциально увеличивает сложность задачи распознавания.

Акустическая модель, лексикон, языковая модель и декодер являются основными модулями традиционной системы АРР. Для определения контекстно-зависимых состояний субфонем из аудио применяется акустическая модель, для сопоставления фонем к словам используется лексикон с языковой моделью. А декодер перевзвешивает все полученные данные от моделей и выводит результат [22]. Такие модели состоят из различных элементов, которые обучаются независимо друг от друга, так классическая акустическая модель может быть обучена на основе смесей гауссовых распределений и скрытых марковских моделей.

До появления интегральных моделей скрытые Марковские модели считались основной технологией в задаче распознавания речи. Даже сегодня модели на основе HMM показывают прогрессирующую производительность распознавания речи совместно с методами глубокого обучения (гибридные модели на основе HMM и глубоких сетей). HMM обычно применяют для динамической деформации времени на уровне кадра и смеси гауссовских распределений плотностей вероятностей (GMM) используются для описания распределений сигналов в промежутке фиксированного небольшого периода времени, который обычно соответствует единице произношения [23].

В последнее время глубокое обучение приносит значительные улучшения во многих исследованиях, и в развитие распознавания речи. Применение нейронных сетей на каждом этапе сценария традиционной системы распознавания речи повышает эффективность ее работы, что отразилась на многих исследовательских работах. С развитием технологий глубокого обучения глубоких нейронных сетей (DNN) стали применять в распознавании речи для построения акустического компонента [4]. В расчете апостериорной вероятности состояния HMM, DNN заменяет обычную вероятность наблюдения GMM [5]. В результате, модель на основе HMM-GMM перевоплощается в HMM с применением DNN, которая превосходит в точности распознавании, и становится популярной моделью АРР. В научной работе [6] было показано, что для получения эффективной акустической модели были применены глубокие нейронные сети, а в [7-8] с помощью рекуррентных нейронных сетей (RNN) и сетей долгой краткосрочной памятью (LSTM) были построены языковые модели (ЯМ) и словарь соответственно. Также в работе [9] ограниченные машины Больцмана были применены для выделения признаков из сигнала речи. Таким образом, многие опубликованные результаты показывают, что предложенный подход демонстрирует лучшую производительность среди всех современных систем распознавания речи, что является основой для использования различных архитектур нейронных сетей на всех этапах распознавания речи.

Методы глубокого обучения также стимулировали появлению альтернативы, которая является интегральной моделью. Интегральная модель, по сравнению с HMM, использует только одну нейронную сеть для непосредственного сопоставления звука со словами. Интегральная модель

представляет многоступенчатый процесс одной сетью, заменяя процесс проектирования на процесс обучения и не требует специальных знаний в этой области, поэтому данную модель легче реализовать и обучать. Именно эти преимущества повлияли на активное использование интегральных моделей в области распознавания речи.

Недавние успехи в машинном обучении показали, что системы можно обучать интегральным способом, то есть системами, в которых каждый шаг обучается одновременно, с учетом всех других шагов и конечной задачи всей системы [24]. В отличие от модели на основе HMM, которая состоит из нескольких модулей, интегральная модель заменяет ряд модулей глубокой сетью, реализуя прямое отображение акустических сигналов в последовательности меток без тщательно продуманных промежуточных состояний. Кроме того, нет необходимости выполнять последующую обработку на выходе. Такие архитектуры обычно состоят из многих уровней по сравнению с классическими системами [25].

Методам интегральных моделей для распознавания речи дается основная роль, так как их легче реализовать, чем обычные системы APP. Во многих современных системах, таких как DeepSpeech2, Voice to Text Messenger, Wav2letter++, Eesen, Google Listen, Attend, Spell, Speech to Translator TTS, была реализована мощная технология, а именно метод интегрального модуля, которая предполагает работу разветвленной многоуровневой сети нейронов, обрабатывающих огромное количество данных и обучающиеся на основе этих данных [26].

В данной главе приведен обзор и анализ существующих интегральных моделей, а также рассмотрены краткое сравнение между моделью на основе HMM и интегральной моделью, также представлена гибридная модель DNN–HMM. Показан анализ различных парадигм интегральных моделей и сравнение их преимущества и недостатки. Для начала рассмотрим традиционную модель распознавания речи.

1.2 Традиционные модели распознавания речи

Традиционный APP фокусируется на предсказании наиболее вероятной последовательности слов для речевого сигнала через аудиофайл или входной поток. Ранние подходы не использовали вероятностный фокус, стремясь оптимизировать последовательность выходных слов, применяя шаблоны для зарезервированных слов к входным акустическим характеристикам (это исторически использовалось для распознавания произносимых цифр). Динамическое преобразование времени (DTW) было одним из первых способов расширения этой стратегии создания шаблонов путем поиска «наименьшего ограниченного пути» для шаблонов. Этот подход позволял варьировать входную временную последовательность и выходную последовательность; однако было трудно придумать соответствующие ограничения, такие как метрики расстояния, выбор шаблонов и отсутствие статистической, вероятностной основы. Эти недостатки усложняли оптимизацию подхода к созданию шаблонов DTW.

Вскоре был сформирован вероятностный подход для преобразования акустического сигнала в последовательность слов. Традиционное распознавание последовательностей сделало акцент на оценке максимальной апостериорной вероятности. Формально этот подход представляет собой преобразование последовательности акустических речевых характеристик X в последовательность слов W. Акустические характеристики представляют собой последовательность векторов признаков длины T: $X = \{x_t \in R^D \mid t = 1,\ldots, T\}$, а последовательность слов определяется как $W = \{w_n \in V \mid n = 1,\ldots, N\}$, имеющей длину N, где V – словарь. Наиболее вероятную последовательность слов W^* можно оценить, максимизируя $P(W|X)$ для всех возможных последовательностей слов V^*(1.2). Данный процесс можно представить следующим выражением [27]:

$$W^* = \underset{W \in V^*}{argmax}\, P(W \mid X) \tag{1.2}$$

Следовательно, основная работа APP заключается в создании модели, которая может точно рассчитать апостериорное распределение p (W | X).

Решение этого количества является центром APP. Традиционные подходы факторизуют это количество, оптимизируя модели для решения каждого компонента, тогда как более современные методы интегрального глубокого обучения сосредоточены на оптимизации непосредственно для этого количества.

Используя теорему Байеса, статистическое распознавание речи определяется как (1.3):

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)} \tag{1.3}$$

Величина $P(W)$ представляет языковую модель (вероятность данной последовательности слов), а $P(X|W)$ представляет акустическую модель. Поскольку это уравнение приводит к максимизации числителя для достижения наиболее вероятной последовательности слов, цель не зависит от $P(X)$, и ее можно убрать (1.4):

$$W^* = \underset{W \in V^*}{argmax}\, P(X|W)P(W) \tag{1.4}$$

Таким образом, структуру традиционного APP представим следующей схемой (рис. 1.2):

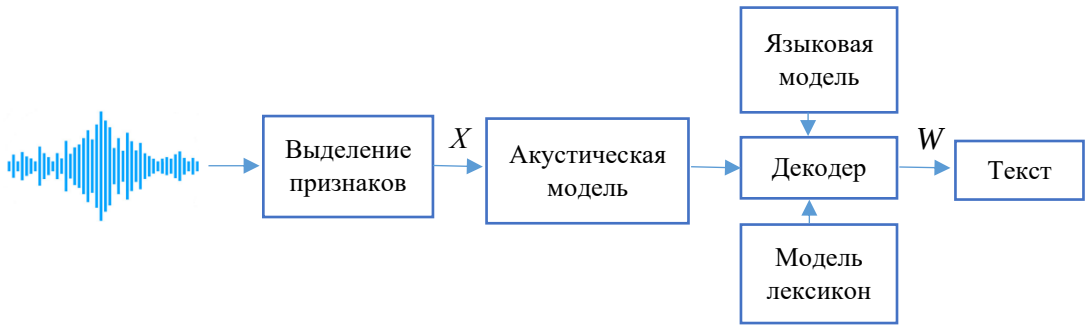


Рисунок 1.2 – Схема традиционной системы распознавания речи

Процесс традиционного метода распознавания речи состоит из последовательностей следующих процедур:

– Выделение из входной речи векторы признаков.

– Акустическое моделирование (определяет, какие именно звуки были произнесены, для последующего распознавания).

– Языковое моделирование (проверяет соответствие произносимых слов наиболее вероятным последовательностям).

– Декодирование последовательности слов, произнесенных человеком.

Алгоритмы извлечения признаков и методы распознавания являются важными частями системы APP. Извлечение признаков – это процесс, который получает небольшое количество данных, существенных для решения задачи [28]. Для извлечения признаков из звукового сигнала обычно используются алгоритмы мел-частотных кепстральных коэффициентов (MFCC) и перцептивное линейное предсказание (PLP) [29-30]. Популярным из них является MFCC.

В задаче распознавания речи входящий сигнал конвертируется в векторы признаков, после на основе этих признаков будет произведена классификация.

Уровень звука, улавливаемый человеческим ухом, связан с уровнем громкости и тембром, и поэтому в качестве единицы измерения высоты воспринимаемого звука был выбран Мел. Мел – единица высоты звука [31]. Обычно его применяют для решения задач анализа человеческой речи, и его использование приближает алгоритмы обработки данных к человеческим параметрам восприятия звука. Самым популярным методом выделения признаков является алгоритм мел-частотных кепстральных коэффициентов. Вычисление MFCC является копией слуховой системы человека, предназначенной для искусственной реализации принципа работы человеческого уха. Существует общеизвестный график зависимости мел-шкалы от частоты колебаний аудио сигнала (рис. 1.3) и вычисляется следующим выражением (1.5):

$$M=1125ln\ (1+f/700) \qquad (1.5)$$

Метод спектральный анализ применяется для изучения аудио сигнала и для нахождения спектра речевого сигнала используется быстрое преобразование

Фурье [32]. Преобразование Фурье – это функция, которая принимает на вход сигнал во временной области и выводит его разложение по частотам, а шкала Мела – это нелинейное преобразование частоты сигнала. После применения преобразования Фурье возникает проблема с выходным сигналом, т.к. будет иметь нелинейные искажения в местах стыков кадров. Для исключения возникшей проблемы полученные результаты быстрого преобразования Фурье нужно умножить на весовую функцию. Эти функции используются в технике цифровой обработки сигналов и являются окнами. В качестве примера можно применить оконную функцию Хеннинга (1.6)

$$X_f = \sum_{k=0}^{K-1} x_k e^{-\frac{2\pi i}{K} fk}, f = 0, \dots, K-1 \quad (1.6)$$

здесь K – размерность дискретного отрезка сигнала, x_k–амплитуда k-го сигнала, X_f – K амплитуд синусоидальных сигналов, которые представляют собой основной сигнал.

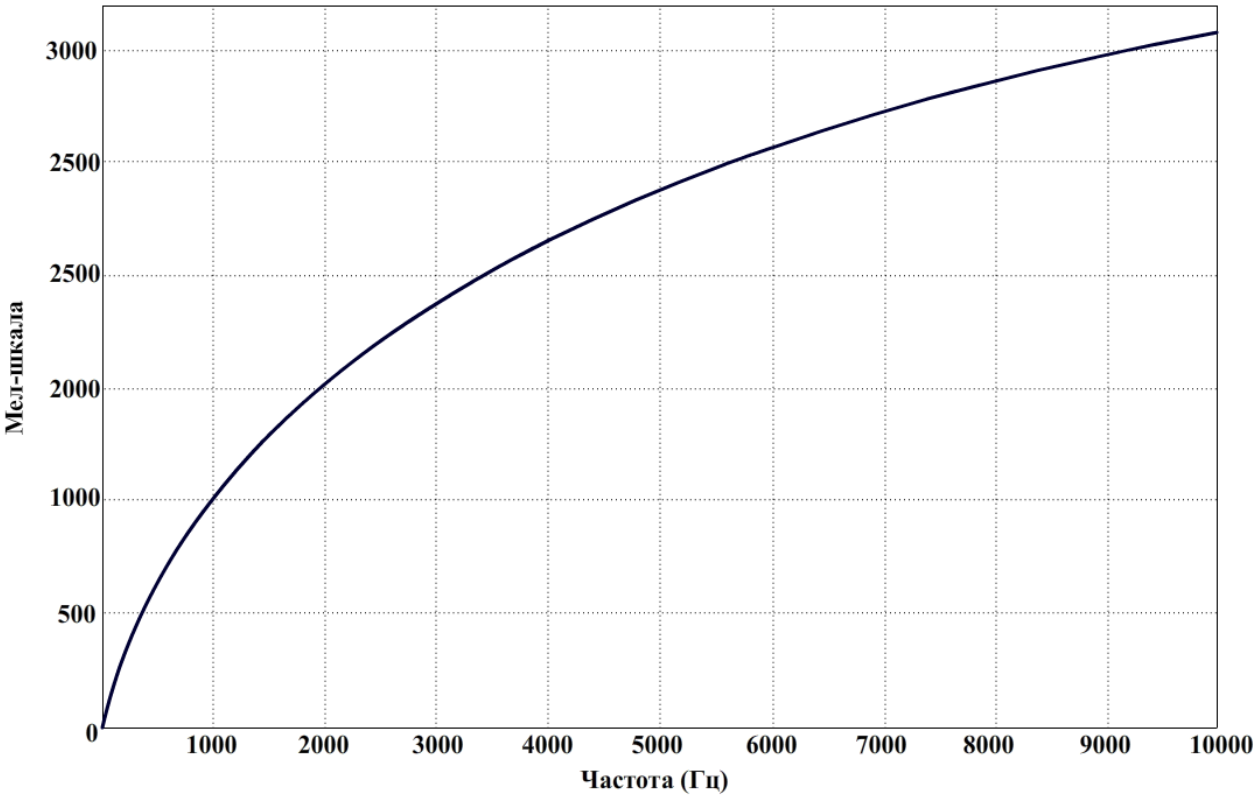


Рисунок 1.3 – График зависимости мел-шкалы от частоты колебаний

Однако при анализе сигналов данных, полученных от одного спектра сигнала недостаточно. В этом случае применяется кепстр (логарифм спектра от входящего сигнала), чтобы представить имеющий спектр в качестве самостоятельного сигнала, благодаря чему значимая спектральная информация представляется более компактно, что в значительной мере облегчает её анализ. Для извлечения акустических признаков применяется алгоритм мел-частотных

кепстральных коэффициентов. Для этого применяя шкалу перевода частоты сигнала в высоту в мелах, вычисляются векторы признаков, с которыми в дальнейшем предстоит работать нейронной сети.

Первым шагом метода вычисления мел-частотных кепстральных коэффициентов является деление входного сигнала на кадры, которые так же называются фреймами или окнами. Длина каждого фрейма прямо пропорционально влияет на результативность работы алгоритма. Самая оптимальная длина фрейма для решения задачи распознавания речи является 25 миллисекунд.

После этого для каждого фрейма вычисляется его спектр с помощью быстрого преобразования Фурье [32]. Полученные коэффициенты фреймов накладываются на мел-частотные окна (рис. 1.4). Человеческое ухо гораздо лучше воспринимает нижние частоты звука, поэтому данные окна сосредотачиваются ближе к низким частотам.

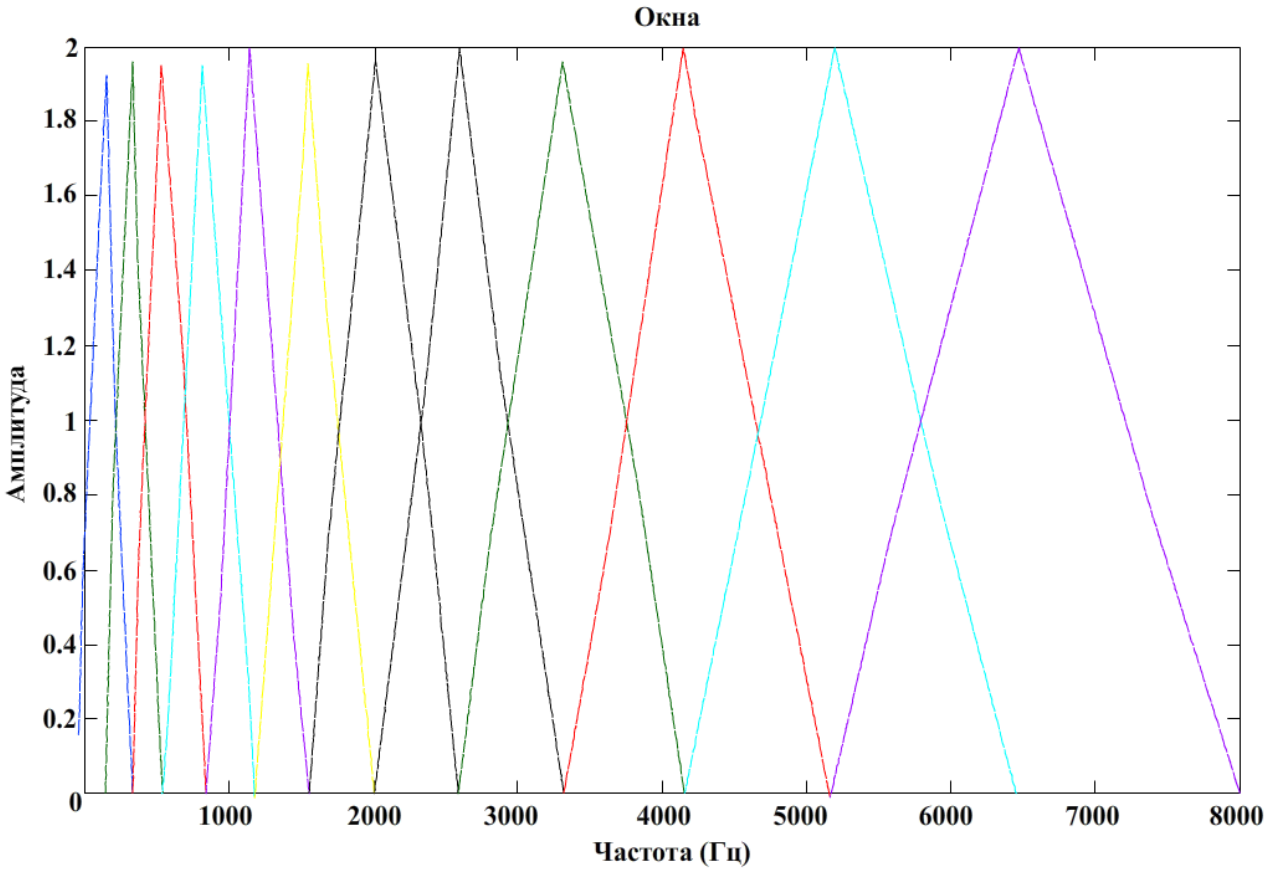


Рисунок 1.4 – Наложение окон на мел-шкалу

Затем применяется дискретное косинусное преобразование, результатом которого является многомерный вектор признаков сигналов, что и является мел-частотными кепстральными коэффициентами.

С помощью полученных векторов признаков, можно установить, какая фонема была произнесена в входящим сигнале. Скрытые марковские модели и

искусственные нейронные сети являются самыми популярными методами в области распознавания речи. [33].

1.2.1 Акустическая модель

Акустическая модель (AM) использует глубокие нейронные сети и скрытые марковские модели [34]. Глубокая нейронная сеть [35], сверточная нейронная сеть (CNN) [36] или долгосрочная краткосрочная память, которая является вариантом рекуррентной нейронной сетью [37] используется для отображения акустического кадра x_t в фонетическое состояние последующего f_t в каждый момент времени t ввода (1.7):

$$P(f_t \mid x_t) = AcousticModel(x_t) \tag{1.7}$$

Перед этой процедурой акустического моделирования выходные цели моделей нейронных сетей, последовательность фонетических состояний на уровне кадра $f_{1:T}$, генерируются HMM и GMM в специальных методах обучения. GMM моделирует акустический элемент на уровне кадра $x_{1:T}$, а HMM оценивает наиболее вероятную последовательность фонетических состояний $f_{1:T}$. Акустическая модель оптимизирована с учетом ошибки кросс-энтропии, которая представляет собой ошибку фонетической классификации на кадр.

1.2.2 Языковая модель

Языковая модель $p(w)$ моделирует наиболее вероятные последовательности слов независимо от акустики (1.8):

$$P(w_u \mid w_{<u}) = LanguageModel(w_{<u}) \tag{1.8}$$

здесь $w_{<u}$ – является предыдушим распознанным словом.

RNN или LSTM обычно широко используются для архитектуры языковых моделей, поскольку они могут фиксировать долгосрочные зависимости а не традиционные модели *n-gram*, которые основаны на марковском предположении и ограничены определенным *n*-диапазоном истории слов. Языковая модель также оптимизирована отдельно с другой целью, сложностью. Эта несвязанная процедура ограничивает другие компоненты, акустическая модель, не может полностью использовать лингвистическую информацию или информацию лингвистического контекста.

1.2.3 Модель лексикон

Модель лексикон сопоставляется между фонетической последовательностью f, которая является минимальной единицей звуков, и последовательностью слов w (1.9):

$$w = LexiconModel(f) \tag{1.9}$$

Например, английские слова представлены в виде 39 различных фонем, или трифонов, которые учитывают контексты левой и правой фонемы. Словарь произношения позволяет быстро находить слово из акустики, ограничивая пространство поиска декодера, однако предполагается, что несколько ограниченных фонем могут охватывать все варианты произношения слова (для всех различных акцентов, стилей речи и т. д.). Словарь произношения обычно составляется вручную с лингвистическими знаниями и редко обновляется, в случае редких слов или новых слов такое произношение недоступно. Хуже того, для языков второго и третьего уровня такие ресурсы произношения могут быть недоступны или ограничены.

1.2.4 Системы на основе скрытых марковских моделей

Ядро всех систем распознавания речи состоит из набора статистических моделей, представляющих различные звуки языка, которые необходимо распознать. Поскольку речь имеет временную структуру и может быть закодирована как последовательность спектральных векторов, охватывающих диапазон звуковых частот, скрытая марковская модель (HMM) обеспечивает естественную основу для построения таких моделей [38-39].

Распознавание речи с использованием HMM дает хороший результат благодаря сходству между архитектурой HMM и различными речевыми данными. Исследование показывает, что нейронная сеть хорошо работает при наличии большого количества обучающих данных, а точность распознавания высока, если рассматриваемое слово взято из набора обучающих данных. В то время как в HMM способность распознавания хороша для неизвестного слова. HMM является общей концепцией и используется во многих областях исследований.

Цель HMM состоит в том, чтобы сопоставить вектор признаков с некоторым представимым состоянием и выдать символ, конкатенация которого дает желаемую последовательность фонем.

В традиционных системах APP модель HMM можно применить и в акустическом моделировании, и в языковом. Но в основном, почти все исследовании посвящены в использовании HMM для акустического моделирования. В модели звук и его особенность являются наблюдением и скрытым состоянием HMM соответственно. Модель применяет теорему Байеса для ввода данных о каждом состоянии HMM $S = \{s_t \in \{1, \cdots, J\} \mid t = 1, \cdots, T\}$ по p (W | X) (1.10).

$$\begin{aligned} \underset{W \in \gamma^*}{argmax}\, p(W|X) &= \underset{W \in \gamma^*}{argmax}\, \frac{p(W,\ X)}{P(X)} \\ &= \underset{W \in \gamma^*}{argmax}\, p(W,\ X) \\ &= \underset{W \in \gamma^*}{argmax}\, \sum_S\ p(P,\ W,\ X) \\ &= \underset{W \in \gamma^*}{argmax}\, \sum_S\ p(X|S,\ W)\, p(S,\ W) \\ &= \underset{W \in \gamma^*}{argmax}\, \sum_S\ p(X|S,\ W)\, p(S|W)\, p(W) \end{aligned} \qquad (1.10)$$

$$W \in \gamma^*$$

После можно аппроксимировать выражение p(X|S, W) ≈ p(X|S) на основе условно-независимой гипотезы. Получаем следующее представление (1.11)

$$\underset{W \in \gamma^*}{argmax}\, p(L|X) \approx \underset{W \in \gamma^*}{argmax} \sum_S p(X|S)\, p(S|W)\, p(W) \quad (1.11)$$

В уравнении (1.11) параметры p(X|S) – совпадает акустической модели, p(S|W) – модель произношения и p(W) соответствует языковой модели.

– p(X|S) находит из скрытой последовательности S вероятность наблюдений X. Согласно гипотезе независимости наблюдения и правилу цепочки вероятностей в HMM, p(X|S) может быть представлена следующим выражением (1.12):

$$p(X|S) = \prod_{t=1}^{T} p(x_t|x_1, \dots, x_{t-1}, S) \approx \prod_{t=1}^{T} p(x_t|s_t) \propto \prod_{t=1}^{T} \frac{p(s_t|x_t)}{p(s_t)} \quad (1.12)$$

Акустическая модель p ($x_t|s_t$) является вероятностью наблюдений X, и вычисляется с помощью модели гауссовских смесей распределений, а через глубокие нейронные сети можно представить распределение апостериорной вероятности скрытого состояния p ($s_t|x_t$). Данные подходы вычисления акустического моделирования соответствуют моделям HMM-GMM и HMM-DNN. DNN находит апостериорную вероятность состояния HMM, заменяя обычную вероятность наблюдения GMM [40], следовательно, HMM-DNN, которая достигает конкурентноспособных результатов, чем HMM-GMM, становится современной моделью в области распознавания речи.

Проблема с распознаванием непрерывной речи заключается в том, чтобы определить границу слова. Это требует знания языковой конструкции и понимания регионального акцента. Другой проблемой является наличие шума в выборочных данных. Для точного распознавания непрерывной речи требуется эффективное решение этих двух проблем для модели HMM, которые являются актуальными до сегодняшнего дня.

1.3 Гибридные модели на основе DNN-HMM

GMM были популярным выбором, потому что они способны напрямую моделировать $P(x_t|s_t)$. Кроме того, они обеспечивают вероятностную интерпретацию входных данных, моделируя распределение для каждого состояния. Однако гауссово распределение в каждом состоянии само по себе является сильным предположением. На практике особенности могут быть сильно негауссовскими. DNN показали значительные улучшения по сравнению с GMM благодаря их способности изучать нелинейные функции. DNN не может напрямую предоставить условную вероятность. Покадровое апостериорное распределение используется для превращения вероятностной модели $P(x_t|s_t)$ в задачу классификации $P(s_t|x_t)$ с использованием трюка псевдоподобного

правдоподобия в качестве аппроксимации совместной вероятности (1.13). Применение псевдовероятности упоминается как «гибридный метод».

$$\prod_{t=1}^{T} P(x_t|s_t) = \prod_{t=1}^{T} \frac{P(x_t|s_t)}{p(s_t)} \quad (1.13)$$

Числитель – это классификатор DNN, обученный с помощью набора входных функций в качестве входного x_t и целевого состояния s_t. Знаменатель $P(s_t)$ – это априорная вероятность состояния s_t. Для обучения покадровой модели требуется покадровое выравнивание с x_t в качестве входных данных и s_t в качестве цели. Это согласование обычно достигается за счет использования более слабой системы согласования HMM/GMM или с помощью словарей, созданных человеком. Качество и количество меток выравнивания обычно являются наиболее значительными ограничениями гибридного подхода.

Архитектура. Глубокое обучение превзошло самые современные результаты во многих областях: распознавание изображений, распознавание речи, языковое моделирование, синтаксический анализ, поиск информации, синтез речи, перевод, автономные автомобили, игры и т.д. DNN могут обнаруживать и изучать сложную структуру очень больших наборов данных. Чтобы использовать классификации DNN в распознавании речи, необходимо найти способ решить задачу переменной длины в речевых сигналах.

Сочетание искусственных нейронных сетей (ИНС) и HMM в качестве альтернативной парадигмы для APP началась в конце 20-века. Это направление исследований было возобновлено в последнее время, после того как высокая производительность изучения представления DNN стала популярной.

Одним из подходов, который был доказан для сочетания DNN с HMM в системе под названием гибридные системы DNN-HMM (рис. 1.5.). В рамках данной архитектуры, динамика речевого сигнала моделируется с помощью HMM и вероятности наблюдения оцениваются через DNN. Каждый выходной нейрон DNN обучен оценить апостериорную вероятность состояния HMM непрерывной плотности учитывая акустические наблюдения. В дополнение к своей сути дискриминационного характера, DNN-HMM имеют два дополнительных преимущества: обучение может быть проведено с помощью встроенного алгоритма Витерби [41] и декодирования.

Наиболее ранние работы по гибридному подходу используют контекстно-независимые телефонные состояния в качестве меток для ИНС обучения и рассматриваемых небольших задач лексики. Наиболее ранние работы по гибридному подходу использовали контекстно-независимые звуковые состояния в качестве меток для ИНС обучения и рассматривали небольшие задачи словарного запаса. ИНС-HMM позже были расширены для моделирования контекстно-зависимых [42] звуков и были применены в лексике и некоторых крупных задачах словарного запаса APP [43], которые также используют разновидные структуры нейронной сети.

Однако в более ранней работе по гибридной архитектуре контекстно-зависимой ИНС-HMM [44], апостериорная вероятность контекстно-зависимого

звука (1.14) была смоделирована как

$$p(s_i, c_j|x_t) = p(s_i|x_t)p(c_i|s_j, x_t) \quad (1.14)$$

где x_t – акустическое наблюдение в момент времени t, c_j – один из сгруппированных классов контекста $\{c_1, \dots, c_j\}$, и s_i – или контекстно-независимый звук или состояние в контекстно-независимом звуке. ИНС были использованы для оценки $p(s_i|x_t)$ и $p(c_i|s_j, x_t)$. Хотя эти типы моделей контекстно-зависимой ИНС-HMM превзошли GMM-HMM для некоторых задач, улучшения были невелики.

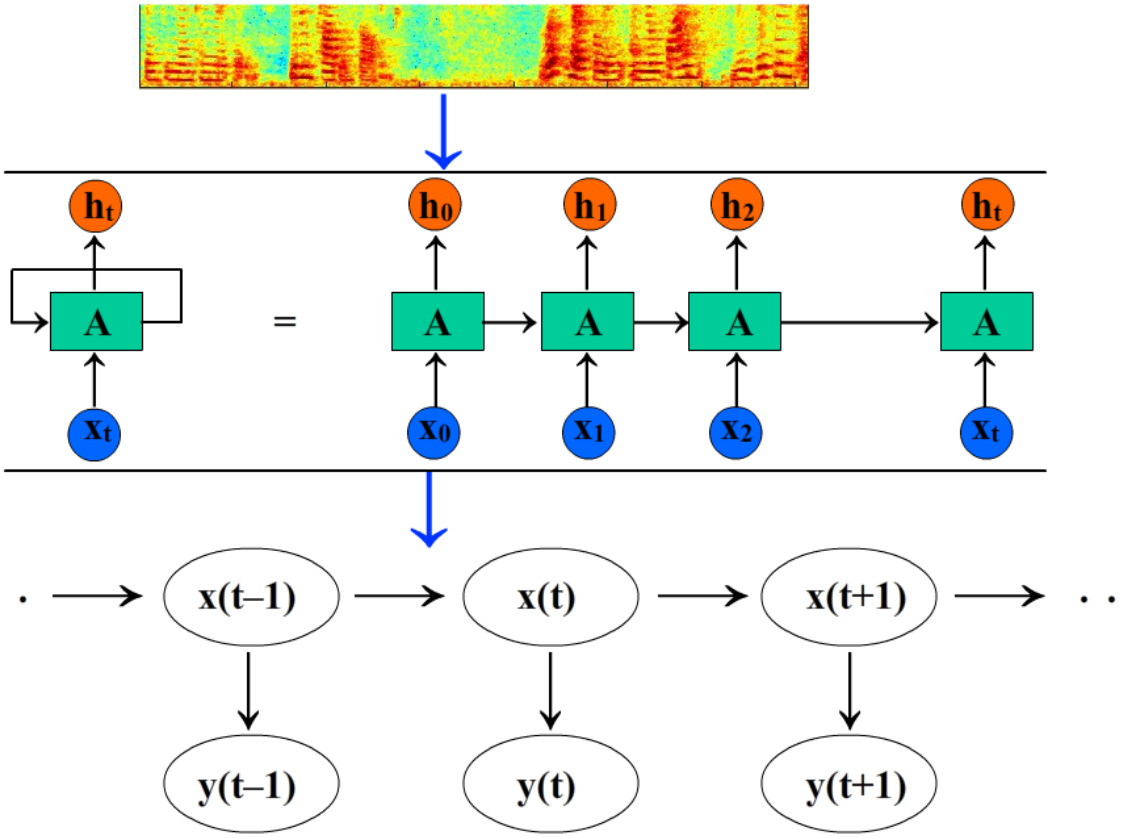


Рисунок 1.5 – Структура гибридной системы DNN-HMM.

Гибридные модели имеют некоторые важные ограничения. Например, ИНС с более чем двумя скрытыми уровнями редко применялись из-за вычислительных ограничений производительности, а также вышеописанная контекстно-зависимая модель учитывает многочисленные эффективные методы, разработанные для GMM-HMM.

Недавний прогресс [45] указывает на то, что значительное улучшение точности распознавания может быть достигнуто при замене традиционных неглубоких нейронных сетей на более глубокие, а также использовать фонем вместо состояния монофона в качестве вывода узлов нейронной сети.

Моделирование фонем также имеет два других преимущества: во-первых,

можно реализовать систему DNN-HMM только с минимальными модификациями к существующей системе GMM-HMM.

1.4 Интегральные модели распознавания речи

Интегральное (end-to-end, E2E) АРР – это новая технология в области машинного обучения, которая предлагает сложную систему обучения одной моделью, и представляет полную целевую систему, минуя промежуточные уровни, обычно присутствующие в традиционных системах. Подход E2E заключается в использовании единого критерия оптимизации для улучшения системы.

Для глобального вычисления $P(W \mid X)$ с помощью интегральных моделей распознавания речи, вход можно представить в виде последовательности речевых данных X, последовательность выходных данных как $У$, а последовательности слов или высказывания в виде W. Таким образом, сеть находит вероятности $P(\cdot|x_1),\ldots,P(\cdot|x_t)$, где входные параметры вероятностей являются некоторые представления последовательности слов, т.е. метки. Структура интегральной модели представлена на рисунке 1.6.

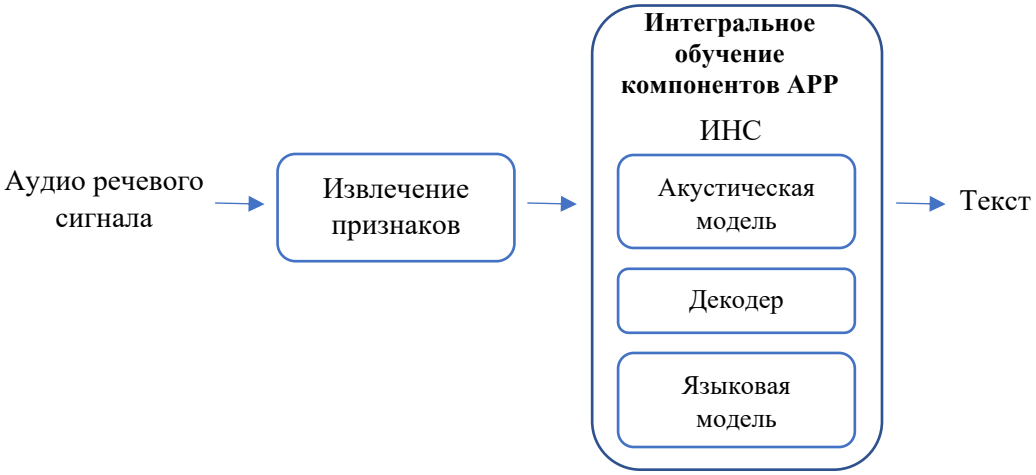


Рисунок 1.6 – Структура интегральной системы распознавания речи

Повышения производительности интегральных систем можно достичь, сосредоточив внимание на трех компонентах:

1) архитектуре модели,
2) больших помеченных наборах обучающих данных,
3) вычислительном масштабе.

Еще одно преимущество E2E-подхода заключается в том, что можно разработать хорошо работающую модель без глубоких знаний о проблеме, несмотря на ее сложность. В зависимости от архитектуры рекуррентных нейронных сетей модель давала лучшие прогнозы с небольшой или без существенной потери точности [40].

Для нахождения лучшей архитектуры модели, системы интегрального обучения значительно выигрывают от больших объемов обучающих данных. В исследовании [39] представлены варианты реализации системы сбора данных,

которые позволили создавать более крупные наборы данных, чем те, которые обычно использовались для обучения систем распознавания речи.

Обучение на больших данных обычно требует использования более крупных моделей. В отличие от предыдущих [10, 11] крупномасштабных подходов к обучению, которые использовали серверы параметров и асинхронные обновления, использовался синхронный стохастический градиентный спуск (SGD), потому что его было легче отлаживать при тестировании новых идей, а также он быстрее сходился для той же степени параллелизма данных. Помимо этого, для улучшения масштабируемости использовались методы оптимизации, обычно встречающиеся в высокопроизводительных вычислениях. Эти оптимизации включают быструю реализацию функции потери коннекционной временной классификации (CTC) на графическом процессоре и специальный распределитель памяти. Тщательно интегрированные вычислительные узлы использовались для ускорения взаимодействия между графическими процессорами (GPU). Благодаря такой масштабируемости и эффективности время обучения сокращается до 1-3 дней, что позволяет быстрее выполнять итерации по моделям и наборам данных. Системы были протестированы на нескольких общедоступных тестовых наборах, и результаты сравниваются с предыдущей интегральной системой. Цель состоит в том, чтобы в конечном итоге достичь производительности человеческого уровня не только в конкретных тестах, которые можно улучшить с помощью настройки для конкретного набора данных, но и в ряде тестов, которые отражают разнообразный набор сценариев.

Интегральная модель может быть разделена на три разные категории в зависимости от их реализаций (рис. 1.7).

Системы глубокого обучения сложно развернуть в любом масштабе. Большие нейронные сети требуют больших вычислительных ресурсов для оценки каждого высказывания пользователя, и некоторые сетевые архитектуры развертываются легче, чем другие. Посредством исследования модели были получены варианты реализации высокоточных развертываемых сетевых архитектур, которые описаны здесь [24]. Также была разработана и внедрена схема пакетной обработки, подходящая для аппаратного обеспечения графического процессора, что приводит к эффективной реализации в реальном времени варианта осуществления механизма на производственные серверы.

Рисунок 1.7 – Виды интегральных моделей

1.4.1 Модель на основе коннекционной временной классификации

Alex Graves является основателем модели коннекционной временной классификации (CTC). CTC – это функция потерь, которая обычно применяется для тренировки РНН для преобразования речь в текст без промежуточного выравнивания входных и выходных данных [46-56].

Во время обучения ИНС с функцией CTC выходной слой ИНС содержит для каждого символа блок выходной последовательности, и добавляется дополнительный блок для «пробела», который определяет бесшумную дорожку звука. Пусть у нас есть выходная последовательность $y = S_w(x)$. Допустим, что любой элемент выходного данного имеет вектора распределения вероятностей для каждой буквы V' в момент времени t. Следовательно, определяем y_k^t, который является вероятностью произнесения символа k из алфавита V' в момент времени t. Если μ – это последовательность символов и «пробела» по входному данному x, то вероятность $P(\mu|x)$ можно вычислить следующим образом (1.15):

$$\mathrm{P}(\mu|\mathrm{x}) = \prod_t y_{\mu t}^t \quad (1.15)$$

Уравнение (1.15) показывает, вывод выходных данных не зависят друг от друга. Это означает, что метка в каждом временном шаге не зависит от выхода последовательностей меток на других временных шагах.

Кроме того, есть необходимость введения дополнительного оператора, который будет удалять повторы букв и «пробелы». Обозначим его как B. Таким образом, полная вероятность выходной последовательности можно представить следующим выражением (1.16):

$$\mathrm{P}(y|\mathrm{x}) = \sum_{\mu \epsilon B^{-1}(y)} P(\mu|\mathrm{x}) \quad (1.16)$$

Приведенное уравнение определяет сумму по всем выравниваниям с использованием динамического программирования, и помогает обучать ИНС на неразмеченных данных (1.17):

$$CTC(x) = -log\ \mathrm{P}(y|\mathrm{x}) \quad (1.17)$$

Необходимо отметить, что ИНС может быть обучена на любом градиентном оптимизированным алгоритме. В архитектуре CTC в качестве шифратора может быть использована любая разновидность ИНС.

Для декодирования CTC-модели было представлено в [46] предположение (1.18):

$$argmax\mathrm{P}(y|\mathrm{x}) \approx B(\mu^{\circ}) \quad (1.18)$$

где $\mu^{\circ} = argmax\mathrm{P}(y|\mathrm{x})$. Как было показано, появление моделей CTC значительно упрощает структуру и обучение модели распознавания речи. Таким образом, нет необходимости создания различных словарей, CTC устраняет необходимость выравнивания данных и позволяет применять довольно много количество слоев, простую структуру сети для реализации модели, который отображает аудио в текст.

Сопутствующие работы

В результате исследований работы [47] было доказано, что модель CTC неплохо работает без языковых моделей для агглютинативных языков на примере казахского языка, но и ResNet показал самый лучший результат, и коэффициент неверно распознанных символов (CER) составил 11,52% и коэффициент неверно распознанных слов (WER), составил 19,57%, с использованием языковой модели. Также сеть двунаправленный LSTM (BLSTM) с помощью архитектуры шифратора-дешифратора с механизмом внимания (attention-based models), показала лучший результат CER, равный 8,01% и WER, равный 17,91%.

В работе [48] описывается метод обучения для акустических моделей с использованием CTC для интегрального автоматического распознавания японской речи. Был предложен интегральный APP на основе сетей двунаправленной кратковременной памяти (BLSTM). Рассматривался два подхода: сокращение количества измерений BLSTM и добавление символьных строк в метки выходного слоя. Результаты эксперимента показали, что модель, объединяющая эти два метода, позволила относительно уменьшить WER на 14,6%.

В работе [49] представлен подход интегральной модели, основанный на моделях с длительной кратковременной памятью (LSTM) и глубокой нейронной сетью с задержкой во времени (TDNN) для распознавания вьетнамской речи. Предложены две интегральные архитектуры, использующие временную коннекционную классификацию в качестве функции потерь. Объединенная модель, использующая TDNN + LSTM, дала лучший результат по сравнению с моделью на основе TDNN, и их производительность была конкурентоспособной по сравнению с традиционной моделью. Так что модель E2E определенно эффективна для вьетнамских систем APP.

В [50] представлена интегральная система транскрипции речи в текст для арабского языка с использованием рекуррентных нейронных сетей без лексики. Целевая функция CTC использовался для максимизации выходных последовательностей символов с учетом акустических характеристик в качестве входных данных. Был использован 1200-часовой корпус мультижанровых программ Aljazeera. В результате WER составил 12,03% для слитной речи.

Bataev и др. [51] представили простую систему на основе графем для распознавания речи с низкими ресурсами с использованием данных Babel для спонтанной турецкой речи и обучили модель на основе CTC с использованием сегментации. И получили частоту ошибок в слове 45,8%, что, является лучшим показателем результатов для систем на основе CTC по распознаванию турецкой речи.

В работе [52] рассмотрели применение наборов инструментов, как Kaldi и CMUSphinx для распознавания азербайджанской речи в системах обслуживания вызовов такси. Полученные результаты показали, что с помощью CMUSphinx можно получить быстрых результатов, а также быстро обучить и легко настроить систему, а Kaldi для получения точного инструмента с относительно низким уровнем дисперсии.

В [53] была построена многоязычная интегральная система распознавания речи для девяти индийских языков. Были анализированы несколько методов для улучшения системы, и в результате было показано что комбинирование обученных языковых слоев адаптера и обусловленного языкового вектора улучшает модель, и решает два основных проблем связанных с несбалансированными данными обучения и необходимость потоковой передачи для распознавания речи. Данная система была построена на основе модели рекуррентного преобразователя (RNN-T) и адаптерных модулей, который превосходит других традиционных систем распознавания речи.

Для распознавания речи на Бахасе в работе [54] были построены модели на основе инструментарий Mozilla DeepSpeech и Kaituoxu SpeechTransformer, которые построены на основе интегрального подхода. Для обучения сети был выбран небольшой речевой корпус Bahasa Indonesia, который состоит из 40 тысяч высказываний. Результаты исследований показали, что модель на Kaituoxu SpeechTransformer работает лучше чем Mozilla DeepSpeech, и WER равны 22% и 23,1% соответственно.

Khassanov и др. [19] представили 335-часовой корпус для казахского языка. В итоге эксперимента было показано, что достаточно большой набор обучающих данных значительно улучшают показатели системы распознавания речи на основе интегральной модели по сравнению с гибридными.

Amirgaliyev и др. [55] рассмотрели метод, который использует другую обученную на большом наборе данных схожего языка модель и применяет ее знания в качестве базы для построения своей модели в рамках метода трансферного обучения с использованием интегральной архитектуры. А для обучения своей модели был разработан корпус казахской речи в объеме 20 часов для обучения нейронной сети. Обученная модель использовал 2 нейронных сетей, как LSTM и BLSTM. Результаты показали, что модель BLSTM с внешней русскоязычной моделью очень хорошо улучшила производительность системы, снизив коэффициент неверно распознанных меток (LER) до 32%.

1.4.2 Модель на основе механизма внимания

Архитектура данной модели состоит из двух отдельных подсетей: подсеть кодера – преобразует последовательность акустических признаков в промежуточное представление, длина которого соответствует длине входной последовательности T; подсеть декодера – предсказывает последовательность меток из промежуточной информации, предоставляемой подсетью кодера. Длина декодированной метки L, обычно меньше, чем длина входного признака T. Декодер использует только соответствующую часть кодированных последовательных представлений для прогнозирования метки на каждом временном шаге с использованием механизма внимания [56].

В качестве кодера обычно используется двунаправленная LSTM. В [57] был использован пирамидальный BLSTM (pBLSTM), который снижает вычислительную сложность, т.е. в каждом последующем сложенном слое pBLSTM уменьшал временное разрешение в 2 раза, таким образом данная модель объединяла выходные данные на последовательные шаги каждого слоя перед передачей их следующему уровню.

Внимание – это «соединитель» между кодером и декодером, который предоставляет декодеру информацию о каждом скрытом состоянии кодера.

Механизм внимания при декодировании сосредотачивает свое внимание на входные данные и выделяет все входные последовательности для обновления скрытых состояний сети и для прогнозирования следующего выходного данного.

Механизмы внимания можно распределить на три вида: в зависимости от местоположения, контента и комбинированный. Комбинированная модель

является наиболее общим видом, в которой важно учитывать предыдущее выравнивание для определения короткой подпоследовательности, с помощью которого данный механизм выберет наиболее важные элементы в схожих частях речи.

Сопутствующие работы

В работе [57] была осуществлена «Listen, Attend and Spell» (LAS) модель, на основе модели с механизмом внимания, где кодер является BLSTM, а декодер использовал LSTM. Данная модель после декодирования использовал языковые модели. Тесты проводились на корпусе Google Voice Search, и лучший результат WER составил 10,3%.

В исследовательской работе [58] была разработана рекуррентная нейронная сеть кодер-декодер под названием Recurrent Neural Aligner (RNA), которая продемонстрировала свою конкурентоспособность. В кодере был изменен временная понижающая дискретизация и была введена мощная сверточная структура. В декодере был применен регуляризатор для сглаживание выходного распределения и совместного обучения с языковой моделью. Результат достиг 27,7% коэффициента ошибок по символам.

В работах [59, 60, 61] был представлен новый метод интегрального распознавания речи для улучшения устойчивости и достижения быстрой конвергенции с использованием совместной модели CTC и механизма внимания в рамках многозадачной обучающей структуры, тем самым смягчая проблему выравнивания. Объединение методов интегральных моделей показал лучший результат, чем использование их по отдельности. Сеть CTC находился поверх кодера и совместно обучался с декодером, основанным на внимании. В процессе поиска луча были объединены предсказания CTC и декодера на основе внимания и отдельно обученная модель языка LSTM. Гибридная модель достигла 5-10%-ного снижения ошибок. Исследование [60] на корпусах Wall Street Journal (WSJ) и CHiME-4 показали улучшение на 5,4-14,6% коэффициента ошибок символов.

В работе [62] предложили гибридную модель шифратор-дешифратор с CTC-вниманием, которая использует глубокие свёрточные сети для сжатия входных функций или спектрограмм сигналов. Также был применен Highway LSTM в качестве шифратора. С помощью предложенных методов в результате эксперимента получили улучшение точности распознавания слитной русской речи и показали лучшую производительность с позиции скорости декодирования речи с использованием метода оптимизации лучевого поиска.

Недавние исследования показали, что модели Sequence to Sequence (Seq2Seq) основанные на внимании, такие как LAS, дают сопоставимые результаты с современными системами APP в различных задачах. В работе [63] была разработана система APP и WER достигла 3,43% на тестовом чистом наборе задачи LibriSpeech для английского языка. Система LAS имеет сопоставимую производительность с системой Kaldi: немного хуже, чем Kaldi при чтении речи, и лучше в спонтанном тестовом наборе. Полученные результаты демонстрируют потенциал Seq2Seq моделей в задачах APP, не относящихся к английскому языку.

В [64] была представлена модель, схожая на Listen, Attend and Spell с применением свёрточных LSTM сетей с остаточными модулями и батч-нормализацией. В качестве обучающего корпуса был выбран WSJ. В результате получили показатель по WER, равный 10,53%.

В работе [65] была разработана онлайн модель LAS, механизм адаптивного монотонного группового внимания (AMoChA) с управлением задержкой (LC). На стороне кодера был использован двунаправленная структура с управлением задержкой, чтобы уменьшить задержку прямого вычисления, а также был предложен механизм адаптивного монотонного группового внимания для расчета распределения веса внимания. Данная модель показала 3,5% относительного снижения производительности по сравнению с базовой линией LAS.

1.4.3 Гибридная модель на основе CTC и внимание

Для повышения надежности и достижения быстрой сходимости системы была разработана модель, которая комбинирует механизм внимания и коннекционную временную классификацию.

Идея модели состоит в том, чтобы использовать целевую функцию CTC в качестве вспомогательной задачи для обучения кодер-декодер модели с вниманием (1.19). Сеть кодера используется совместно с моделями CTC и внимания (рис. 1.8.) [56].

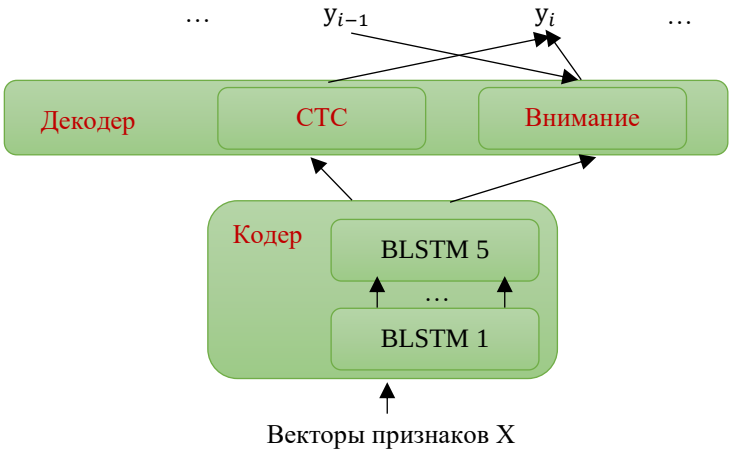


Рисунок 1.8 – Структура предлагаемой гибридной модели

В отличие от модели внимания, алгоритм прямого-обратного поиска в CTC может обеспечить монотонное выравнивание между последовательностями речи и меток. Поэтому данная структура модели будет более надежной в получении соответствующих выравниваний в шумных условиях. Еще одним преимуществом использования CTC в качестве вспомогательной задачи является быстрое обучение сети. Алгоритм прямого-обратного поиска в CTC помогает ускорить процесс оценки желаемого выравнивания без помощи грубых оценок выравнивания, который требует ручного труда [56].

$$L_{CTC/Att} = \tau L_{CTC}(x) + (1-\tau)L_{Att}(x) \quad (1.19)$$

здесь τ – настраиваемый параметр и удовлетворяет условие – $0 \leq \tau \leq 1$.

Сопутствующие работы

Полученные результаты, при совместном использовании модели CTC и кодер-декодер моделей на основе механизма внимания, отразились в следующих работах:

В работе [66] была предложена улучшения в механизме извлечения признаков и внимания в гибридной архитектуре, которая добавляет дополнительную потерю CTC к модели, основанной на внимании, что может привести к дополнительным ограничениям на выравнивание. Во-первых, совместная модель обучалась высокоуровневыми функциями на основе неотрицательной матричной факторизации. Затем был предложен гибридный механизм внимания, включающий внимание нескольких голов, который вычисляет оценки внимания по результатам многоуровневого анализа. Эксперименты с TIMIT показывают, что предложенный метод обеспечивает высочайшую производительность. Эксперименты с WSJ показывают, что метод демонстрирует WER, который всего на 0,2% хуже по абсолютной величине, чем метод, на который лучше всего ссылаются, который обучен на гораздо большем наборе данных, и превосходит все существующие интегральные методы. Другие эксперименты с LibriSpeech показывают, что реализованный метод также сопоставим с современной интегральной системой в WER.

Haoran Miao и другие [67] предложили основанную на Transformer онлайн-архитектуру CTC и внимание E2E APP, которая содержит кодер самовнимания фрагментов и декодер самовнимания на основе монотонного усеченного внимания. Во-первых, кодер самовнимания фрагментов разбивает речь на изолированные фрагменты. Чтобы снизить вычислительные затраты и повысить производительность, они предлагали многократное использование состояния кодера самовнимания фрагментов. Во-вторых, декодер самовнимания на основе монотонного усеченного внимания монотонно усекает речевые характеристики и уделяет внимание усеченным функциям. Для поддержки онлайн-распознавания они интегрируют блок повторного использования состояния-кодер самовнимания фрагментов и декодера самовнимания на основе монотонного усеченного внимания в онлайн-архитектуру CTC/внимания. Предлагаемые онлайн-модели на тесте HKUST Mandarin APP достигли 23,66% CER с задержкой 320 мс. И предложенная онлайн-модель дает всего 0,19% абсолютного ухудшения CER по сравнению с автономным базовым уровнем.

В работе [68] была улучшена гибридная модель, для начала было исследовано ограниченное по времени внимание с учетом местоположения CTC/внимание, устанавливая надлежащий размер ограниченного по времени окна внимания. Далее была введено ограниченное по времени самовнимание CTC/внимание, которое может лучше моделировать дальнодействующие зависимости между фреймами. Эксперименты с заданиями Wall Street Journal, расширенным многосторонним взаимодействием и коммутатором демонстрируют эффективность предлагаемого ограниченного по времени

самовнимания CTC/внимание. Наконец, чтобы исследовать устойчивость этого метода к шуму и реверберации, были объединены интерфейс обучающего нейронного формирователя луча с серверной частью CTC и внимание APP с ограниченным временем внимания в наборе данных CHIME-4. Уменьшение частоты ошибок по словам и повышение перцепционной оценки качества речи подтверждают эффективность этой структуры.

Такие гибридные интегральные модели также полезны при распознавании агглютинативных языков, в который входит и казахский язык.

В работе [69] была исследована онлайн-система APP для японского языка с использованием модели, основанной на однонаправленных LSTM, обученных с использованием CTC с локальным механизмом внимания. В результате был получен лучший показатель по CER 9,87%. В результате работы частота ошибок по слогам составила 10,5%.

Hosung Park и другие [70] реализовали метод интегрального распознавания речи на основе внимания и CTC, в которой в качестве единиц распознавания использовались корейские графемы. Чтобы предсказать результаты интегральной модели, в этом исследовании используются выходные структуры графемных единиц. Построение сети на основе графем обеспечивает эффективное обучение за счет сокращения числа прогнозируемых выходных параметров. В результате работы частота ошибок по слогам составила 10,5%.

В [71] представлен новый APP на основе CTC, CNN-LSTM и гибридный подход с механизмом внимания для арабского языка. Была добавлена языковая модель для достижения высоких результатов. Обучение и тестирование всех моделей было выполнено на корпусе, который содержит 7 часов современной стандартной арабской речи. Экспериментальные результаты показывают, что CNN-LSTM с фреймворком внимания превосходит обычный APP и совместный фреймворк CTC-внимание APP в задаче распознавания арабской речи. Это скорее всего объясняется с отсутствием достаточного объема данных для обучения модели.

В [72] документе предлагается гибридная CTC с архитектурой внимания и алгоритм слогового изменения для амхарской системы автоматического распознавания речи с использованием ее подсловных единиц, основанных на фонемах. Предложенная интегральная модель была обучена различным подсловам амхарского языка, а именно символам, фонемам, подсловам на основе символов и подсловам на основе фонем, созданным с помощью алгоритма сегментации кодирования пар байтов. Экспериментальные результаты показали, что контекстно-зависимые подслова на основе фонем, как правило, приводят к более точным системам распознавания речи, чем подслова на основе символов, фонем и символов. Был получен WER 18,38%.

1.4.4 Модель на основе условных случайных полей (CRF)

J. Lafferty et al. [73] предложили алгоритм оценки параметров для условных случайных полей и показали, что CRF имеет большее преимущество перед HMM и MEMM (maximum entropy Markov models) для данных на естественном языке. Модель CRF применяется для оценки измерения точности в задаче

фонетического распознавания, так и точности обнаружения границ между ними. Результаты показывают, что при использовании функций перехода в структуре распознавания на основе CRF производительность распознавания значительно улучшается за счет уменьшения количества удалений фонем. Показывают, что при использовании функций перехода в структуре распознавания на основе CRF производительность распознавания значительно улучшается за счет уменьшения количества удалений телефонов. Эффективность определения границ также улучшается, в основном, для переходов между фонетическими классами тишины, остановки и щелчков. Помимо этого, модель CRF дает более низкий уровень ошибок, чем модели HMM и Maxent в задаче обнаружения границ предложений в речи.

В распознавании речи распространение получили сегментные и линейные CRF [74]. В основе сегментного подхода лежит использование акустических последовательностей в качестве исходных данных и автоматическое построение универсального набора функций на уровне сегментов [75] (рис. 1.9). На рисунке 1.9 величины X и Y обозначены, как случайные величины для последовательностей данных и для них соответствующие последовательности меток.

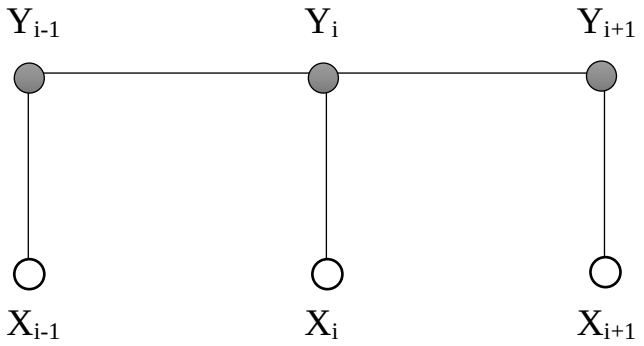


Рисунок 1.9 – Общая архитектура модели на основе CRF

В исследованиях [76-77] было обнаружено, что модель CRF улучшила показателей системы распознавания без интеграции языковой модели, чем методы на основе Марковских моделей с максимальной энтропией (MEMM), который является аналогом модели на основе условных случайных полей.

Сопутствующие работы

В настоящее время наиболее распространенной в распознавании речи являются модели линейного и сегментарного CRF. Данная модель чаще всего применяется для решения задач разметки и сегментации последовательностей.

Keyu An и его соавторы [78] продемонстрировали новый набор инструментов CAT, который представляет реализацию интегральных моделей CTC-CRF. Для эксперимента были применены китайские и английские тесты, как Switchboard и Aishell, таким образом были получены самые современные результаты среди существующих интегральных моделей с меньшим количеством параметров и конкурентоспособен по сравнению с гибридными

моделями DNN-HMM. Кроме того, эти же авторы в работе [79] предложили новый метод, называемый контекстуализированным мягким забыванием (contextualized soft forgetting), который позволяет инструменту CAT выполнять потоковую APP без снижения точности и показал хорошие результаты с ограниченными наборами данных по сравнению с существующими немодуляризованными моделями E2E. Однако необходимо понимать, что применяемые языки имеют достаточно большой корпус для обучения модели, что играет немаловажную роль.

В работе [80] была построена интегральная модель на основе сегментных условных случайных полей (SCRF) и коннекционной временной классификации (CTC). SCRF использует глобально нормализованную совместную модель меток и длительностей сегментов, а CTC классифицирует каждый кадр либо как выходной символ. Благодаря экспериментам с набором данных TIMIT, многозадачный подход к обучению повысил точность распознавания моделей CTC и SCRF. Помимо этого, было показано, что CTC можно использовать для предварительного обучения кодера RNN, ускоряя обучение совместной модели.

Hongyu Xiang и другие [81] разработали одноступенчатое акустическое моделирование на основе условного случайного поля с топологией состояния, основанной на CTC. Оценочные эксперименты проводились с наборами данных WSJ, Switchboard и Librispeech. При прямом сравнении модель CTC-CRF, использующая простые двунаправленные LSTM, последовательно превосходила lattice-free maximum-mutual-information по всем трем наборам данных эталонного тестирования.

В работе [82] были исследованы методы, позволяющие успешно применять недавно разработанные модули моделирования текстовых слов и нейронные сети Conformer в CTC-CRF. Эксперименты проводятся на двух английских наборах данных – Switchboard и Librispeech, и немецком наборе данных CommonVoice. Экспериментальные результаты показывают, что Conformer может значительно улучшить качество распознавания; словесные системы работают немного хуже по сравнению с телефонными системами для целевого языка с низкой степенью соответствия графемфонемы, в то время как обе системы могут работать одинаково хорошо, когда такая степень соответствия высока для целевого языка.

Yang, Li и др. [83] предлагают модель обработки текста после распознавания китайской речи, которая сочетает в себе двунаправленную сеть с долговременной краткосрочной памятью (LSTM) с моделью условного случайного поля (CRF). Необходимость обработки текста после распознавания связана с появлением проблемы с диалектом и акцентом, так как необходимо исправить текст после распознавания речи перед отображением. Задача разделена на два этапа: выявление текстовых ошибок и редактирование текстовых ошибок. В этой статье двунаправленная сеть с долговременной краткосрочной памятью и условное случайное поле используются на двух этапах обнаружения текстовых ошибок и исправления текстовых ошибок соответственно. Благодаря проверке и системному тестированию набора данных

SIGHAN 2013 Chinese Spelling Check экспериментальные результаты показывают, что модель может эффективно повысить точность текста после распознавания речи.

К сожалению, очень мало новых исследований по данной тематике именно в области распознавания речи. Но постаралась по максимуму рассмотреть научные работы, которые были в открытом доступе.

1.4.5 Рекуррентный нейронный преобразователь (RNN-T)

CTC определяет распределение по последовательностям фонем, которое зависит только от входной акустической последовательности. RNN-T объединяет CTC-подобную сеть [46] с отдельной RNN, которая предсказывает каждую фонему с учетом предыдущих, тем самым создавая совместно обученную акустическую и языковую модель [84-85]. В то время как CTC определяет выходное распределение на каждом входном временном шаге, RNN-T определяет отдельное распределение вероятностей для каждой комбинации входного временного шага и выходного временного шага. Модель решает проблемы, связанные с CTC, сохраняя при этом некоторые из своих преимуществ по сравнению с моделями внимания. Преобразователь является авторегрессивным: он принимает в качестве входных данных предыдущие выходные данные и создает функции, которые можно использовать для прогнозирования следующего вывода, например стандартную языковую модель.

В отличие от большинства моделей seq2seq, которые обычно должны обрабатывать всю входную последовательность для получения выходных данных, RNN-T непрерывно обрабатывает входные образцы и передает выходные символы, свойство, которое приветствуется для речевого диктанта. В реализации выходные символы — это символы алфавита. RNN-T выводит символы один за другим, с пробелами, где это необходимо. Он делает это с помощью цикла обратной связи, который возвращает в него символы, предсказанные моделью, для предсказания следующих символов, как показано на рисунке ниже (рис. 1.10).

Таким образом, с выводит последовательность данных в онлайн режиме, осуществляя потоковое распознавания речи, обновляя кодера и сеть прогнозирования в зависимости от состоянии метки . При появлении пустой метки вывод данных прекращается [88].

Наиболее вероятная последовательность меток во время логического вывода рассчитывается с применением поиска луча, с небольшим изменением, которое дает алгоритму использовать меньше ресурса без потери производительности модели [87].

В отличие от Neural Transducer , который используется для потокового распознавания речи, в RNN-T сеть прогнозирования не учитывает выходные данные от кодера [89]. Данное преимущество дает RNN-T предварительно тренировать декодера в текстовых данных в качестве ЯМ.

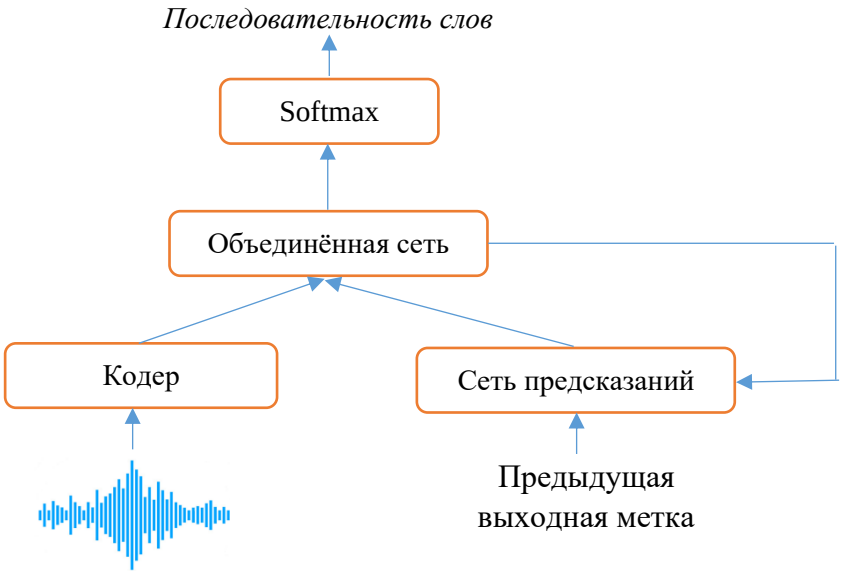


Рисунок 1.10 – Структура RNN-T

Сопутствующие работы

Ilya Sklyar и др. [90] разработали новую многооборотную модель RNN-T со стратегией расположения целей на основе перекрытия, которая обобщается на произвольное количество говорящих без изменений в архитектуре модели. Исследование проводилось на тестовом наборе LibriCSS и в результате улучшили WER на 28%.

Prabhavalkar R. и др. [91] провели детальную оценку обычной RNN-T и дополненной вниманием RNN-T. Примечательно, что каждая из этих систем напрямую предсказывает графемы в письменной области, не используя внешний лексикон произношения или отдельную языковую модель. Было обнаружено, что данные модели конкурентоспособны с традиционными современными подходами к наборам тестов для диктовки.

В [92] обнаружили, что производительность RNN-T можно повысить за счет использования единиц подслов, которые охватывают более длинный контекст и значительно уменьшают количество ошибок замены. RNN-T, двенадцатиуровневым LSTM-кодер с двухуровневым LSTM-декодером, обученный с использованием 30 000 фрагментов слов в качестве целей вывода, обеспечивает лучший уровень ошибок в словах на 8,5% при голосовом поиске и 5,2% при голосовой диктовке.

Wang S. и др. [93] исследовали обучение модели RNN-T для распознавания китайского языка. Было предложено несколько методов для ускорения обучения RNN-T, включая оттачивание стратегии снижения скорости обучения, отказ от предварительного обучения кодера путем добавления слоев CNN, ускорение обучения за счет правильной субдискретизации и инициализации ЯМ. В итоге результаты были улучшены на 2%.

В [94] была улучшена RNN-T: уменьшили потребление памяти при обучении RNN-T за счет эффективного сочетания выходных данных кодера и сети прогнозирования, чтобы избежать хранения в памяти нескольких больших

тензоров, а также была улучшена базовая структура модели RNN-T, в которой используются единицы LSTM, предлагая несколько новых структур. Все предлагаемые структуры используют концепцию траектории слоя, которая разделяет задачу классификации и задачу временного моделирования с использованием единиц глубины LSTM/GRU и единиц времени LSTM/GRU соответственно. Таким образом, были снижены показатели WER на 6,6%, 7,5% и 11,8% соответственно на тестовых наборах Cortana, Conversation и DMA по сравнению с базовой моделью RNN-T.

1.4.6 Архитектура Transformer

Модель Transformer был впервые создан для машинного перевода заменяя при этом рекуррентные нейронные сети в задачах обработки естественного языка. В данной модели полностью была устранена рекуррентность, вместо этого для каждого высказывания с помощью внутреннего механизма внимания строились признаки для выявления значимости других последовательностей для данного высказывания. Следовательно, созданные признаки для данного высказывания это и есть результат линейных преобразований признаков последовательностей, которые являются значимыми.

Модель Transformer состоит из одного большого блока, который в свою очередь состоит из блоков кодеров и декодеров (рис. 1.11).

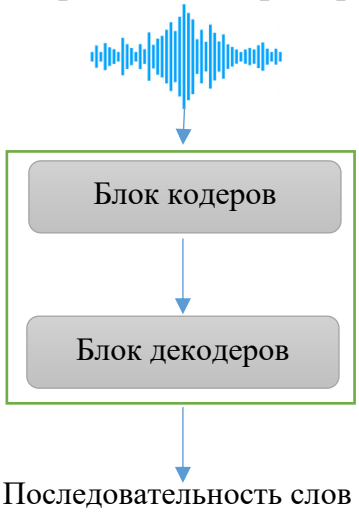


Рисунок 1.11 – Общая схема модели

Здесь кодер во вход принимает вектора признаков из аудиосигнала $X = (x_1, ..., x_T)$ и выводит последовательность промежуточных представлений. Далее на основе полученных представлений декодер воспроизводит выходную последовательность $W = w_m = (w_1, ..., w_M)$.

Каждый этап модели использует предыдущие символы для вывода следующего, т.к. является авторегрессивной. Архитектура Transformer (рис. 1.12) использует несколько слоев самовнимания в блоках кодера и декодера, которые взаимосвязаны друг с другом. Рассмотрим каждый блок по отдельности.

Сети кодера и декодера. Обычные интегральные кодер-декодер модели для задач распознавания речи состоят из одного кодера и декодера, механизма внимания. Кодер преобразует вектор акустических признаков в альтернативное представление, а декодер предсказывает последовательность меток из альтернативной информации, предоставляемым кодером, далее внимание выделяет значимые части фрейма для предсказания выхода. В отличие от этих моделей в том, что в моделе Transformer может быть несколько кодеров и декодеров, и каждый из них содержать свой внутренний механизм внимания.

Блок кодера состоит из наборов кодеров; в качестве исследования обычно берутся 6 кодеров, которые расположены друг над другом. Количество кодера не фиксированное, можно поэкспериментировать с произвольным количеством кодера в блоке. Все кодеры имеют одинаковые структуры, но имеют разные веса. На вход кодера поступают извлеченные векторы признаков из аудиосигнала, полученные с помощью мел-частотных кепстральных коэффициентов или сверточных нейронных сетей. Далее первый кодер трансформирует эти данные с помощью самовнимания в набор векторов, и через ИНС прямого распространения передает полученные выходы следующему кодеру. Последний кодер обрабатывает векторы и передает данные закодированных функций в блок декодера.

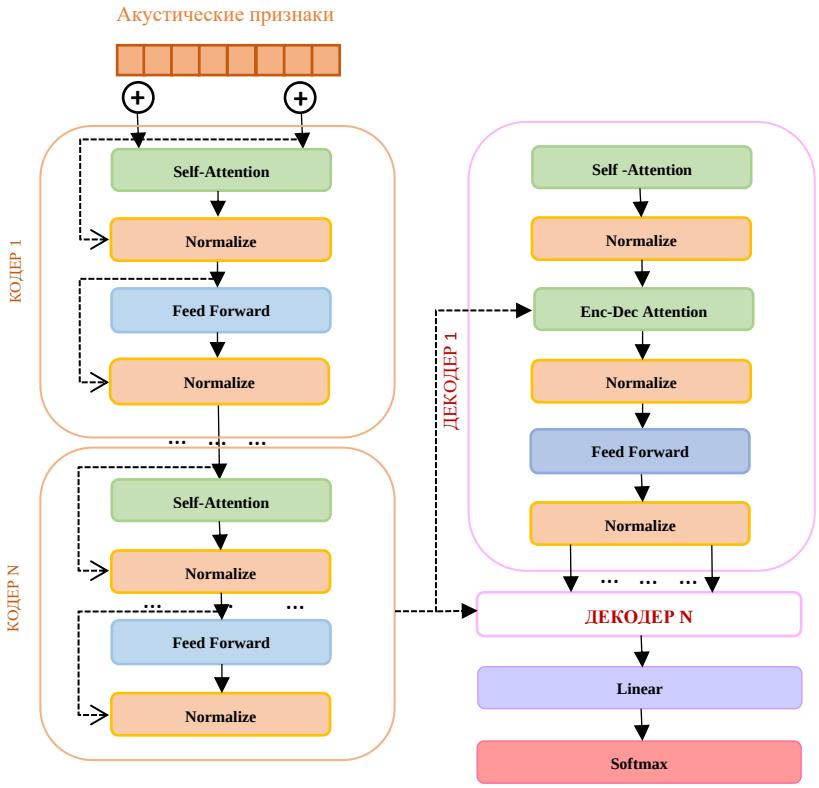


Рисунок 1.12 – Архитектура модели Transformer

Блок декодера это и есть набор декодеров, и их количество обычно идентично с количеством кодеров. Каждую часть кодера можно разделить на два подслоя: входные данные, поступающие в кодер, сначала проходит через слой

multi-head attention, помогающий кодеру посмотреть на другие слова во входящем предложении во время кодирования конкретного слова. Выход слоя внутреннего multi-head attention отправляется в нейронную сеть прямого распространения. Точно такая же сеть независимо применяется для каждого слова в предложении.

Декодер также содержит два этих слоя, но между ними есть слой внимания, который помогает декодеру фокусироваться на значимых частях входящего предложения, как и является схожим с обычным механизмом внимания в seq2seq моделях. Данный компонент будет учитывать предыдущие символы/слова и на основе этих данных на выход выдает апостериорные вероятности последующего символа/слова.

Self-attention mechanism. Модель Transformer включает в себя Scaled Dot-Product Attention [95]. Преимущество self-attention – это быстрое вычисление и сокращение пути между словами, а также потенциальная интерпретируемость. Данное внимание включает в себя 3 вектора: queries, keys и values и масштабирование (1.20):

$$Attention(U^Q, U^K, U^V) = softmax\left(\frac{U^Q U^{K\mathrm{T}}}{\sqrt{d_{attention}}}\right)U^V \qquad (1.20)$$

Эти параметры считаются полезными для вычисления внимания. Multi-head attention объединяет несколько self-attention карт в общие матричные вычисления (1.21):

$$MultiHeadAttention(Q, K, V) = Concat(s_1, \dots, s_h)U^{head} \qquad (1.21)$$

Здесь $s_h = Attention(QW_h^Q, KW_h^K, VW_h^V)$. h – это количество внимания в слое, $QW_h^Q, KW_h^K, VW_h^V, s_h$ – обучаемые матрицы весов.

Механизм многоголового внимания можно применить в качестве оптимизационной задачи. С помощью данного механизма можно обойти проблемы, связанные с неуспешной инициализацией, а также улучшить скорость обучения. Помимо этого, после обучения можно исключить некоторые части голов внимания, т.к. эти изменения никак не повлияет на качество декодирования. Количество голов в модели предназначено для регулирования механизмов внимания. Кроме того, данный механизм помогает сети с легкостью обратиться к любой информации вне зависимости от длины последовательности, т.к. это осуществляется легко вне зависимости от количества слов в наборе.

В архитектуре Transformer можно заметить элемент Normalize, который необходим для нормализации значений признаков, так как после использования механизма внимания данные значения могут иметь разные величины.

Выходы нескольких голов могут быть также разными, и в итоговом векторе разброс значений может быть большим. Для предотвращения данного случая был предложен подход, где значения на каждой позиции преобразовывают двухслойным перцептроном. После применения механизма внимания, значения

проецируются на большую размерность с помощью обучаемых весов, где затем происходит преобразование нелинейной функцией активации ReLU, а потом эти величины проецируют в исходную размерность, за которой происходит очередная нормализация.

Сопутствующие работы

Модель Transformer впервые была представлена в работе [95], с целью сокращения последовательных вычислений и количества операций для соотнесения сигналов входных и выходных позиций. Были проведены эксперименты по задачам машинного перевода, с английского на немецкий и с английского на французский язык. В результате было показано, что модель достигла хороших показателей по сравнению с существующими результатами. Более того, Transformer отлично работает и для других задач с большими и с ограниченными данными обучения, и является очень плодотворной для всевозможных seq2seq задач.

Применение Transformer для преобразования речи в текст также показало хорошие результаты и отразились в следующих исследовательских работах:

В исследовательской работе [96] была предложена модель на основе Transformer для потоковой передачи распознавания речи, что требует на вход целое речевое высказывание. Для генерации выхода после озвученного слова был применен ограниченное по времени самовнимание в кодере и запущенное (triggered) внимание для кодер-декодера с механизмом внимания. Архитектура модели достигла лучшего результата по интегральному распознаванию потоковой речи – 2,8% и 7,3% WER для «чистых» и «других» тестовых данных LibriSpeech.

В статье [97] был предложен метод подавления слабого внимания (Weak-Attention Suppression), который динамически вызывает разреженность вероятностей внимания. Данный метод подавляет внимание некритичных и избыточных непрерывных акустических кадров и с большей вероятностью подавляет прошлые кадры, чем будущие. Было показано, что предложенный метод приводит к снижению WER по сравнению с базовыми видами Transformer. В тесте LibriSpeech предлагаемый метод подавления слабого внимания снизил WER на 10% при тестировании-чистоте и на 5% в другом тесте для потоковых Transformer, что привело к появлению нового передового уровня среди потоковых моделей.

Была предложена система Speech-Transformer в работе[98], использующая механизм 2D-внимания, который совместно обрабатывает временную и частотную оси двумерных речевых входов, тем самым обеспечивая более выразительные представления для Speech-Transformer. В качестве тренировочных данных был использован корпус Wall Street Journal. Результаты эксперимента показали, что данная модель позволяет сократить время обучения и при этом может обеспечить конкурентоспособный WER.

В работе [99] была предложена Transformer с SLT-адаптацией – архитектура для разговорного языкового перевода, для обработки длинных входных последовательностей с низкой плотностью информации для решения задач APP.

Адаптация была основана на понижении дискретизации входных данных с помощью сверточных нейронных сетей и моделировании двумерной природы спектрограммы звука с помощью 2D-компонентов. Эксперименты показывают, что адаптированный Transformer превосходит базовый уровень на основе RNN как по качеству перевода, так и по времени обучения, обеспечивая высокую производительность по шести языковым направлениям.

В [100] была предложена архитектура Transformer, на основе которой было разработано контекстное окно, которая была обучена на сценариях монолога и диалога. Были применены тесты монолога на корпусе спонтанной японской речи и TED-LIUM3 и тесты диалога на SWITCHBOARD. В итоге были получены результаты превосходящие базовые интегральные APP с одним звуком и с i-векторами говорящих или без них.

В интегральной системе модель кодер-декодер на основе RNN была заменена архитектурой Transformer в [101]. И для того, чтобы использовать данную модель в маскирующей сети нейронного формирователя луча в многоканальном случае, был модифицирован компонент самовнимания так, чтобы он ограничивался сегментом, а не всей последовательностью, чтобы сократить объем вычислений. Кроме улучшений архитектуры модели, также была включена предварительная обработка внешней деревербераций, взвешенную ошибку предсказания (weighted prediction error, WPE), что позволяет модели обрабатывать реверберированные сигналы. Эксперименты с расширенным корпусом WSJ показывают, что модели на основе Transformer достигают лучших результатов в безэховых условиях в одноканальном и многоканальном режимах, соответственно.

Для реализации более быстрой и точной CAPP, были объединены Transformer и достижения APP на основе RNN [102]. Для построения модели была интегрирована коннекционная временная классификация с Transformer для совместного обучения и декодирования. Такой подход ускоряет обучение и способствует интеграции ЯМ. Предлагаемая система APP реализует значительные улучшения в различных задачах APP. К примеру, он снизил WER с 11,1% до 4,5% для Wall Street Journal и с 16,1% до 11,6% для TED-LIUM, внедрив интеграцию CTC и ЯМ в базовый уровень Transformer.

1.5 Выводы

Данная глава посвящена полному обзору и анализу моделей интегральных систем автоматического распознавания речи. В этой главе

1. приведена общая модель системы автоматического распознавания речи, которая представляет собой три основные независимые модули, такие как акустические модели для прогнозирования контекстно-зависимых состояний субфонем из аудио, языковые модели и лексикон для сопоставления фонем к словам;
2. описан стандартный процесс распознавания речи, который состоит из следующих шагов: выделение признаков из входного сигнала; акустическое

моделирование; языковое моделирование, лексикон; декодирование последовательностей;

3. рассмотрена модель на основе скрытых марковских моделей, которая долгое время была основной моделью распознавания слитной речи до появления глубокого обучения. Также приведены описание и архитектура гибридной модели на основе скрытых марковских моделей и глубоких нейронных сетей;

4. приведен обзор интегральных систем распознавания речи, как коннекционная временная классификация и рекуррентный преобразователь, модель кодер-декодер и архитектура Transformer с механизмом внимания, модель на основе условных случайных полей. Приведен обзор сопутствующих работ по рассмотренным моделям. А также приведен сравнительный анализ данных моделей с традиционными и гибридными моделями.

2 РАЗРАБОТКА РЕЧЕВОГО КОРПУСА ДЛЯ ИНТЕГРАЛЬНОГО РАСПОЗНАВАНИЯ КАЗАХСКОЙ РЕЧИ

При разработке корпуса необходимо охватить все возможные акустические данные, такие как акценты по регионам, жанры разного направления, вводные слова и т.д. Для оценки разработанного корпуса необходимо создать фонетический богатый текст. Полученный большой корпус в дальнейшем можно использовать в задачах идентификации и верификации диктора, распознавания языка, разработке искусственного интеллекта или в других задачах распознавания.

Для построения языкового корпуса казахского языка были собраны текстовые материалы с открытых интернет источников, такие как новостные сайты, информационные источники на казахском языке, а также применены другие доступные данные в электронном виде. Все собранные текстовые материалы продиктованы и озвучены дикторами разных полов и возрастов для исследования и расширения базы данных аудиозаписей.

2.1 Сбор речевого корпуса из открытых источников

Качество интегральной системы распознавания речи в большинстве случаев зависит от объема обучающих данных. Корпус в среднем должен содержать несколько тысяч часов речи [103]. Для сбора речевых и текстовых данных были рассмотрены новостные сайты (https://qazaqstan.tv/live, https://24.kz/kz/), you-tube каналы, а также сайты аудиокниг (https://kitap.kz/, https://e-history.kz/ru/audio/) и радио-станции как «Қазақ радиосы» (https://qazradio.fm/kz/) и Шалкар (https://shalkarfm.kz/kz), а также полезные сайты на казахском языке, как https://massaget.kz/, https://tilmedia.kz/, https://tilalemi.kz/ и другие. Кроме того, был применен корпус голосовых данных на казахском языке, разработанный Институтом умных систем и искусственного интеллекта Назарбаев Университета (ISSAI https://issai.nu.edu.kz/), включающий в себя около 300 часов записанной речи более двух тысяч человек.

В просторах интернета можно найти множество различных аудиокниг на казахском языке. Но их текстовые данные не всегда доступны, что приходится искать их в других доступных источниках. Несмотря на это, запись многих аудиокниг является чистой и корректной в произношении. В новостных сайтах и you-tube каналах можно получить только аудиоданные, что требует дополнительных усилий для ручного транскрибирования речи. Но тем не менее, преимущество you-tube каналов является разнообразие записанных голосов в различных зашумленных средах.

Все записи были сохранены в отдельные файлы с транскриптами. В итоге получилось собрать:

- аудиозаписи трансляций новостей и с you-tube каналов – более 50 часов аудиоданных;
- аудиозаписи с художественных аудиокниг – более 20 часов аудиоданных.

Эти аудиоданные не имеют речевых шумов и помех, и состоит только из чистой речи.

2.2 Создание и расширение речевого и текстового корпусов

Производительность и качество системы АРР почти полностью зависит от разнообразия и объема речевой базы данных. Есть много факторов, которые оказывают влияние на качество системы АРР. К ним относятся условия записи, окружающая среда, длительность аудио, устройства записи, пол, возраст и область проживания диктора и т.д. В зависимости от условия записи речевых данных, можно получить разные результаты системы автоматического распознавания речи.

Для расширения корпуса казахского языка участвовали 380 дикторов, носители казахского языка. Запись каждого диктора занимало в среднем 40-50 минут времени. Для каждого диктора был назначен отрывок текста, состоящий из 50-60 предложений. Каждая запись, с продолжительностью 15-20 секунд, была сохранена в отдельный файл. При подборе дикторов учитывались следующие признаки: возраст, пол, область проживания, уровень образования, время записи речи и т.д. (рис. 2.1).

Основные характеристики казахского речевого корпуса приведены в таблице 2.1.

Таблица 2.1 – Характеристики казахского речевого корпуса

Общая характеристика	Тип речевого материала	Состав фрагментов/ предложений	Дикторы /фрагменты
Текстовый материал	Прочитанная чистая речь; набор предложений и высказываний, полученных из газетных и новостных текстов на интернет-сайтах, аудиокниг	более 200 часов речи; более 185 548 фрагментов-предложений	380 дикторов: 140 мужчин и 240 женщин разного возраста
	Спонтанная смешанная речь	1800 часов записи; данные частично размеченные	Телефонные/мобильные диалоги людей разного возраста и пола

Текстовый материал корпуса включает 185 548 отдельных предложений. Были привлечены специалисты-лингвисты и языковеды для анализа и проверки корпуса в целях обеспечения высокого качества. С новостных сайтов и других

полезных сайтов на казахском языке (Егемен Қазақстан, Жас Алаш и т.д.) были собраны аудио данные и текстовые материалы для дальнейшего озвучивания и расширения корпуса. Эти речевые данные составили 7 часов аудиоданных. А также были отобраны некоторые материалы в электронном формате, как аудиокниги художественных литератур – 2 часа аудиоданных. Для транскрибирования аудиофайла была разработано правило составления текстовок.

Было собрано 2000 часов аудиоданных – записи дикторов, художественных аудиокниг и новостных лент сформировали около 200 часов, записи мобильной речи составили более 1800 часов. Записи дикторов (моносигнальный) и записи мобильной речи (стереосигнальный) были использованы для исследовательских целей.

После вышеперечисленных работ был создан один из важных элементов – словарная база для системы распознавания речи. Все записанные тексты собраны в одном файле и удалены повторяющиеся слова. Затем, были сортированы в алфавитном порядке.

Словарь произнесённых слов содержит 800805 неповторяющихся слов и их фонетическую транскрипцию. Акустическая база содержит около 2000 часов речи и 45 ГБ памяти на диске.

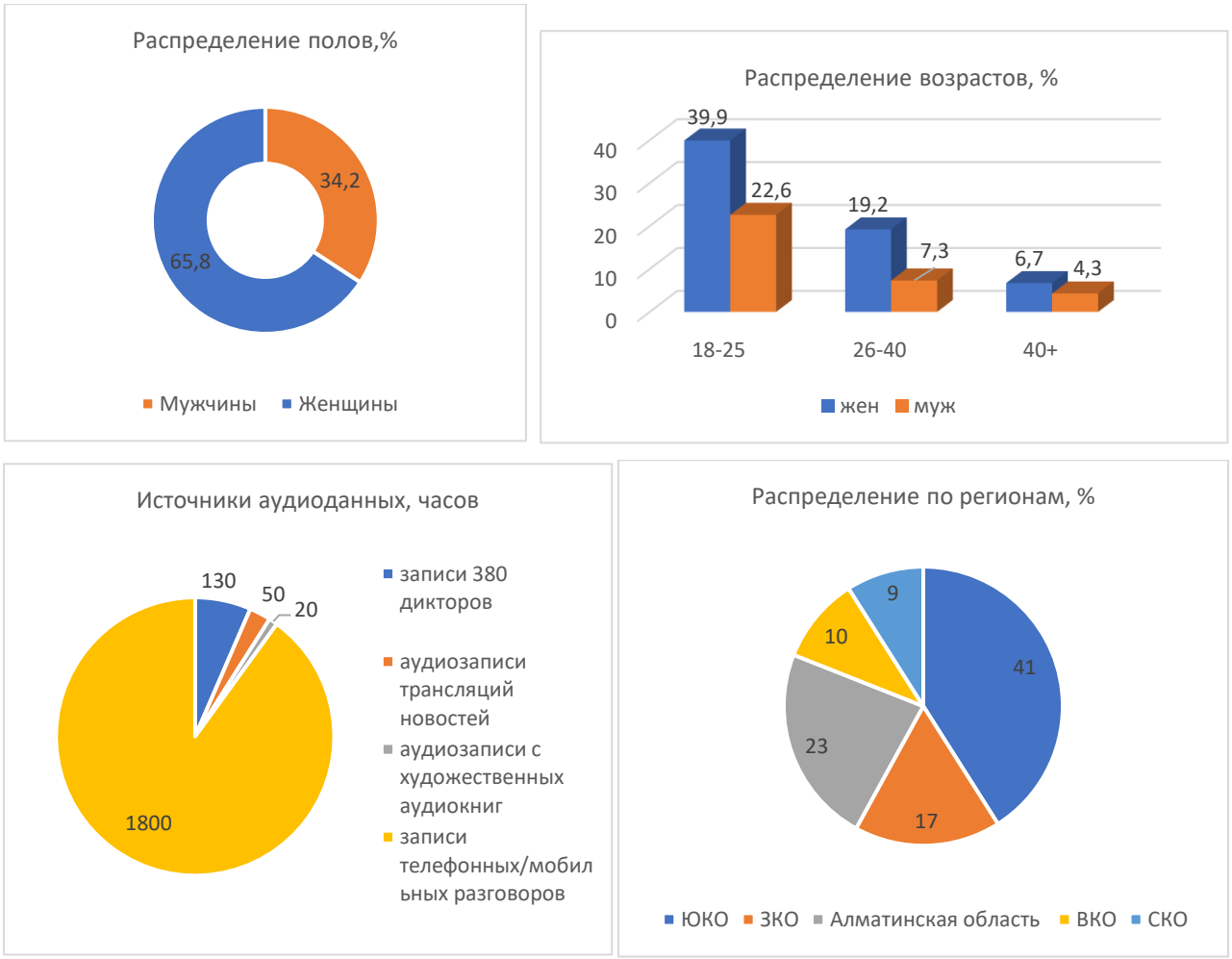


Рисунок 2.1 – Структура казахского речевого корпуса

В корпусе аудио файлы, полученные от дикторов, были разделены на две части: тренировочная и тестовая. При этом руководствовалась следующими соображениями. Тестовая часть корпуса должна составлять от 5 до 15% всего корпуса:

– запись одних и тех же дикторов не должна использоваться одновременно в обеих частях,

– каждый набор данных не должен содержать одинаковых предложений,

– тестовый набор должен обеспечивать полное покрытие фонемного инвентаря, достаточное разнообразие их фонетических контекстов и частоту встречаемости.

Каждое предложение сохранялся как отдельные файлы, а название состоялись из следующих идентификаторов: код региона, пол, год рождения, ФИО диктора, образование, номер текста, номер предложения в тексте.

Для разработки качественного, размеченного корпуса много средств и усилий было потрачено на ручную транскрибацию аудио файлов телефонной речи. Преимущество телефонной речи является то, что речь там спонтанная. Однако, было исследовано, что наличие неразмеченных данных положительно влияет на качество системы APP. Это дает возможность существенно увеличить количество дикторов и размеры текстового материала при разработке речевого корпуса.

2.3 Транскрибирование телефонных разговоров

Для транскрибирования телефонных разговоров была создана специальная методика по составлению текстовок, т.к. данная речь является спонтанной, и может содержать информацию на иностранном языке, а также речь может содержать ярко выраженный акцент и разного рода речевых шумов, неречевых шумов, как сигнал тонального набора, телефонный гудок, звуки, напоминающие удар, щелчок, а также звуки, которые служат для обдумывания следующего высказывания. Кроме того, могут быть неразборчивые слова, накладка нескольких дикторов и т.д. Необходимо заметить, что подбор текстовых массивов с заранее оговоренными статистическими требованиями на контекстное употребление фонем представляет собой очень трудоемкую задачу и заняла достаточно огромное количество времени. Для транскрибирования телефонных разговоров были вовлечены молодые волонтеры – это студенты последних курсов, магистранты и докторанты алматинских национальных университетов.

Разработанная методика по составлению текстовок приведена в следующем в виде:

1) Прослушать произнесение
2) Удаляем файлы, в которых:

- только речь автоответчика;
- только телефонные сигналы или фоновая речь;
- только произнесение одного слова не до конца (обрывка слова);

- нельзя разобрать ни одного слова;
- только произнесение на иностранном языке (русском тоже);
- речь нескольких дикторов полностью накладывается;
- речь имеет ярко выраженный акцент;
- только шумы и речевые шумы;
- ничего нет (пустые файлы).

3) Если в произнесении есть разборчивая речь на казахском языке, то в текстовом поле в звуковом редакторе Wave Assistant начинаем писать текст после слов: Sentence=. В тексте нельзя использовать клавишу «Enter». Текст пишется в одну строку (пусть редактор сам переносит слова по мере набора текста).

4) Кодировка текстовых файлов должна быть: UTF-8 without BOM

5) Текст должен максимально соответствовать содержанию звукового файла. Не допускаются сокращения, пропуски слов, замена слов, упрощения речевых конструкций.

6) Слова записываем в соответствии с правилами орфографии.

7) Дефисы ставим в соответствии с правилами орфографии.

8) Все числовые обозначения должны записываться в буквенном виде. Пример: «отыз бірінші қаңтар».

9) Аббревиатуры так и записываются. Пример: «АҚ»

10) Имена собственные записываются с большой буквы.

11) Англоязычные слова типа «SMS» записываем так, как они должны быть в английском языке, однако в «слэшах» - «/» записываем слово так, как оно было произнесено: /эсэмэс/. ВАЖНО: не ставим пробел между словом-оригиналом и соответствующим ему словом-произнесением, записанным в «слэшах». Пример: «Он отправил мне SMS/эсэмэс/».

12) Из пунктуационных знаков ставим только точки. Не надо ставить: кавычки, тире, двоеточия, вопросительные и восклицательные знаки и пр. ВАЖНО: часть текста, отделяемая точкой, обладает смысловой завершенностью (это самостоятельное высказывание).

13) Между словами обязательно должен быть пробел.

14) В случае достаточно четкого произнесения «угу» и «ага» текстуем их как обычные слова. В случае если «угу»/«ага» произнесено без размыкания губ (специфическое произнесение для выражения подтверждения) – то текстуем (!ршум); в случае если специфически произнесено отрицание, также без размыкания губ – то текстуем (!ршум).

15) Если слово произнесено не до конца, то оно записывается в круглых скобках в нижнем регистре. Пример: «Мен Алматыдамын. Көп (ұзам) жолға шығамын. Жақсы (бопт). Рахмет. Аман бол. Сау бол. Кездескенше».

16) Если непонятно, что говорит диктор, то этот отрывок речи обозначаем в тексте (!белгісіз). Пример: «Төлқұжат керек (!белгісіз)».

17) Заполненные паузы хезитации (звуки «а», «э», «м», которые служат для обдумывания следующего высказывания), а также такие речевые шумы, как

кашель и смех, отмечаем: (!ршум). Если в паузе несколько подобных шумов идут подряд (разнородных или однородных), то отмечаем их одной меткой (!ршум).

18) Любые неречевые шумы (например, сигнал тонального набора, телефонный гудок, звуки, напоминающие удар, щелчок) отмечаем меткой (!шум). Если в паузе несколько подобных шумов идут подряд (разнородных или однородных), то отмечаем их одной меткой (!шум).

19) Русские слова (а также русские слова, к которым добавляются казахские окончания) отмечаем в тексте знаком #. Например: «мен #номер он жетінші #гимназия мектепке бардым және де он жетінші #гимназия мектебін бітірдім».

Разработанный корпус речи предназначен для работы с большими наборами данных, чтобы проверить установленные характеристики системы, а также исследовать влияние разнообразных данных на качество системы распознавания речи.

Все аудиоматериалы имели формат .wav. Количество аудиоканалов в мобильных разговорах является стерео, в записях дикторов – моно. Все аудиоданные были приведены в одноканальное. Для получения данных в цифровом формате был применен метод PCM. Разрядность 16 бит, дискретная частота 44,1 кГц.

2.4 Выводы

Во второй главе приведено описание речевого и текстового корпусов для казахского языка. В ней

1. рассмотрен сбор корпуса для казахского языка, который состоит из аудиоданных с их транскрипциями (текстовое представление аудио) и содержание корпуса. Для построения корпуса казахского языка были собраны текстовые материалы с открытых интернет источников. Все собранные текстовые материалы продиктованы и озвучены дикторами разных полов и возрастов для исследования и расширения базы данных аудиозаписей.
2. приведен общий объем собранного корпуса: 2000 часов аудиоданных – записи дикторов, художественных аудиокниг и новостных лент сформировали около 200 часов, записи мобильной речи составили более 1800 часов. Записи дикторов и записи мобильной речи были использованы для исследовательских целей.
3. для улучшения качества системы в корпус были добавлены аудиоданные телефонных разговоров. Так как речь в таких аудиоданных являются спонтанной, и может содержать информацию на иностранном языке, ярко выраженный акцент и разного рода речевых шумов, неречевых шумов, как сигнал тонального набора, телефонный гудок, звуки, напоминающие удар, щелчок, а также звуки, которые служат для обдумывания следующего высказывания, для таких случаев была разработана и приведена методика транскрибирования телефонных разговоров. Полученный большой корпус в дальнейшем можно использовать в задачах идентификации и верификации

диктора, распознавания языка, разработке искусственного интеллекта или в других задачах распознавания.

3 РАЗРАБОТКА ИНТЕГРАЛЬНОЙ МОДЕЛИ ДЛЯ РАСПОЗНАВАНИЯ КАЗАХСКОЙ РЕЧИ

3.1 Модель кодер-декодер с механизмом внимания

Интегральный метод, основанный на внимании, состоит из следующих элементов: подсеть кодера – преобразует последовательность акустических признаков в промежуточное представление, длина которого соответствует длине входной последовательности; подсеть декодера – предсказывает последовательность меток из промежуточной информации, предоставляемой подсетью кодера. Длина декодированной метки, обычно меньше, чем длина входного признака. Декодер с использованием механизма внимания выделяет значимую часть кодированных последовательных состояний для вывода метки на каждом временном шаге [104].

В модели с механизмом внимания кодер трансформирует входные данные *X* в последовательность промежуточных представлений *H* (3.1):

$$H = Encoder(X) \tag{3.1}$$

Декодер генерирует полученные промежуточные вектора в выходные последовательности (3.2):

$$P(y|x) = AttentionDecoder(H,y) \tag{3.2}$$

Архитектура модели представлена в рисунке 3.1.

В качестве кодера обычно используется двунаправленная LSTM. В [57] был использован пирамидальный BLSTM (pBLSTM), который снижает вычислительную сложность, т.е. в каждом последующем сложенном слое pBLSTM уменьшал временное разрешение в 2 раза, таким образом данная модель объединяла выходные данные на последовательные шаги каждого слоя перед передачей их следующему уровню.

Внимание – это «посредник» между кодером и декодером, который предоставляет декодеру информацию о каждом скрытом состоянии кодера.

Механизм внимания при декодировании сосредотачивает свое внимание на входные данные и выделяет все входные последовательности для обновления скрытых состояний сети и для прогнозирования следующего выходного данного (3.3-3.5):

$$att_i = Attend(l_{i-1}, att_{i-1}, H) \tag{3.3}$$

$$gl_i = \sum_{j=1}^{T} att_{i,j} h_j \tag{3.4}$$

$$y_i = Generate(l_{i-1}, gl_i) \tag{3.5}$$

где att_i – вектор весов внимания, l_{i-1} – (i – 1)-е состояние сети, gl_i – проблеск, с помощью которого сеть генерирует выход y_i, на основе определенных элементов H. Далее шаг завершается вычислением нового состояния сети $l_i = Recurrency(l_{i-1}, gl_i, y_i)$.

Механизм внимания был модифицирован и расширен, в котором учитывается предыдущее выравнивание att_{i-1} для определения короткой подпоследовательности H, с помощью которого данный механизм выберет наиболее важные элементы в схожих частях речи. Это модификация позволяет корректно распознавать схожие окончания в словах. Для вычисления вектора весов внимания функцию *Attend* можно представить в нижеприведенным выражением (3.6-3.7)

$$e_{i,j} = Score(l_{i-1}, H) \tag{3.6}$$

$$att_{i,j} = \frac{\exp(e_{i,j})}{\sum_{j=1}^{L} \exp(e_{i,j})} \tag{3.7}$$

где Score вычисляются в зависимости от механизмов внимания по расположению (3.8), по контенту (3.9) соответственно следующим образом [35]:

$$e_{i,j} = w^T \tanh(Wl_{i-1} + VH + b) \tag{3.8}$$

$$e_{i,j} = w^T \tanh(Wl_{i-1} + VH + Uf_i + b) \tag{3.9}$$

где W, V, f, U, b – обучаемые величины,

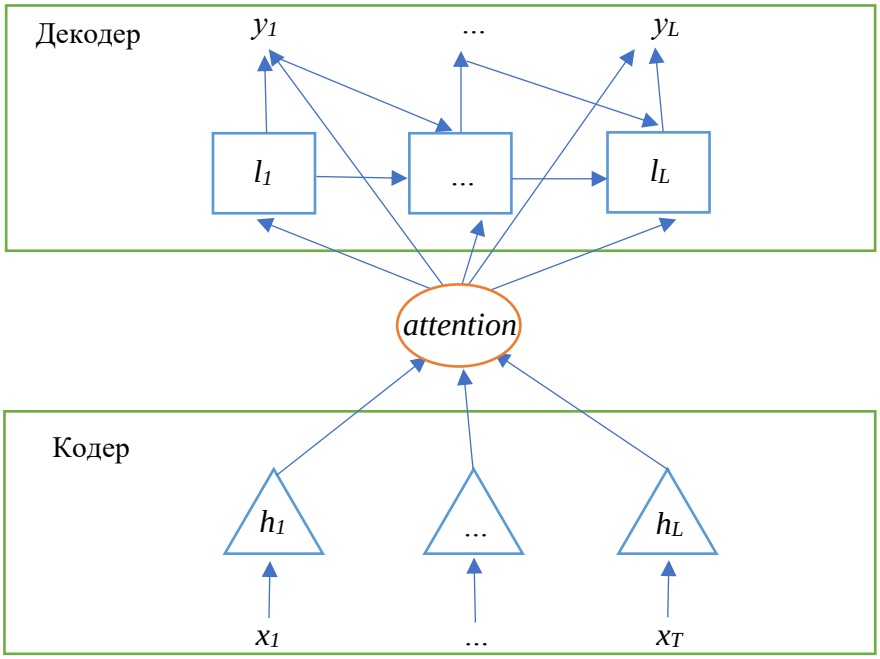


Рисунок 3.1 – Архитектура модели кодер-декодер на основе внимания

Для вычисления att_i возникают проблемы с нормализацией. Данная проблема решается путем метода сглаживания (3.10) [104].

$$att_{i,j} = \frac{\sigma(e_{i,j})}{\sum_{j=1}^{L} \sigma(e_{i,j})} \quad (3.10)$$

Далее декодер генерирует полученные через кодер промежуточные вектора представления H в выходные последовательности. В качестве декодера обычно использовались генераторы повторяющихся последовательностей на основе внимания (Attention-based Recurrent Sequence Generators) [64] и LSTM [57] соответственно.

3.1.1 Виды механизмов внимания

Функционально механизм внимания можно разделить на три типа: на основе контента, на основе местоположения и гибридный. Они используют разную информацию в процессе внимания и имеют разные характеристики.

На основе контента: он использует только входную последовательность функций H и предыдущее скрытое состояние l_{i-1} для расчета веса в каждой позиции. Его проблема в том, что он не использует информацию о местоположении (3.11). Для различного внешнего вида функции в последовательности функций внимание на основе контента придаст им одинаковый вес, что называется проблемой фрагмента речи схожести.

$$att_i = Attend(l_{i-1}, H) \quad (3.11)$$

В зависимости от местоположения: предыдущий вес att_{i-1} используется в качестве информации о местоположении на каждом этапе для расчета текущего веса att (3.12). Но поскольку он не использует входную последовательность объектов H , этого недостаточно для входных объектов.

$$att_i = Attend(l_{i-1}, att_{i-1}) \quad (3.12)$$

Гибрид: как следует из названия, он учитывает входную последовательность признаков H, предыдущий вес att_{i-1}, и предыдущее скрытое состояние l_{i-1} (3.13), что позволяет сочетать преимущества контекстного и по местоположению внимания.

$$Attend(l_{i-1}, att_{i-1}, H) \quad (3.13)$$

3.1.2 Архитектура модели кодер-декодера с механизмом внимания

В данной работе кодер-декодер модель с механизмом внимания была получена для распознавания казахской слитной речи. Для извлечения признаков были использованы мел-частотные кепстральные коэффициенты (MFCC). Размер окна было равно к 25 мс.

В предлагаемой модели в качестве кодера использовалась двунаправленная LSTM (BLSTM), которая содержала пять BLSTM-слоев; а декодером являлась обычная LSTM, которая содержала два LSTM-слоев. Был использован многослойный перцептрон (MLP) в качестве механизма внимания (рис. 3.2). В качестве эксперимента были использованы гибридные механизмы внимания для распознавания казахской речи.

Для улучшения результатов была использована функция softmax в декодере в качестве входных данных для предсказания следующего шага во время обучения.

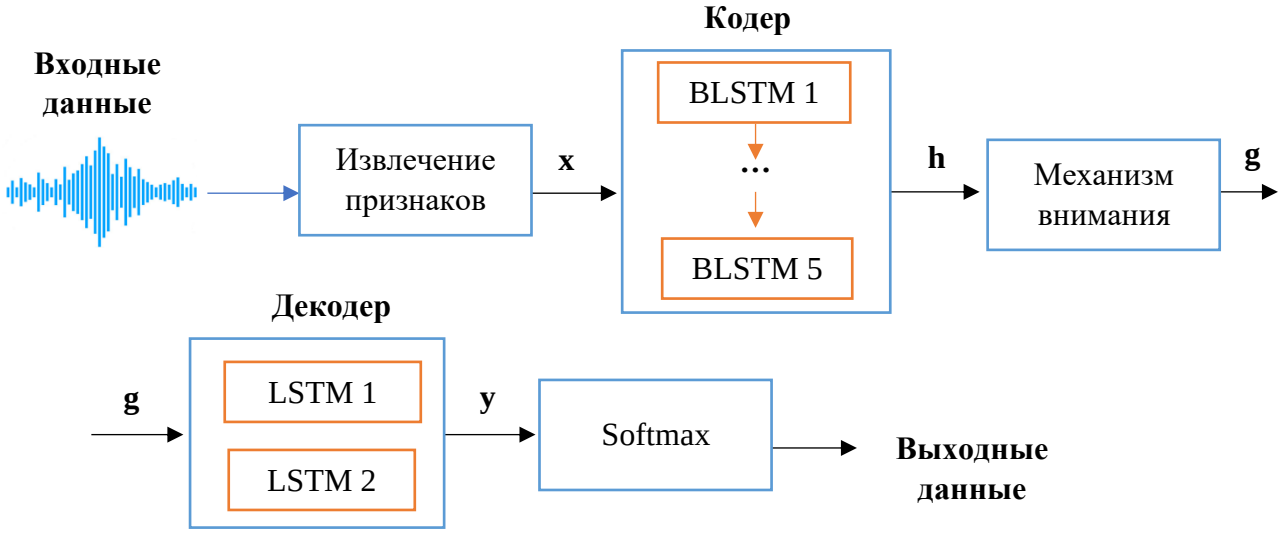


Рисунок 3.2 – Архитектура полученной модели

3.1.3 Предварительная настройка модели с механизмом внимания

Основные настройки элементов модели с вниманием представлены в таблице 3.1.

Таблица 3.1. Предварительная настройка модели

Кодер	**Механизм внимания**	**Декодер**
BLSTM	MLP	LSTM
5 слоев с 1024 единицами и выпадал с вероятностью удержания, равной 1,5	4 скрытых слоев с 1024 единицами, которая использует функцию активации ReLU с начальной скоростью обучения, равной 0,002, и коэффициентом уменьшения – 1,5	2 слоя с 1024 единицами в каждом с выпадением, равным 0,5, с начальной скоростью обучения, равной 0,001, и коэффициентом затухания, равным 1,5

Модель кодер-декодер на основе внимания сложно обучать на длинных входных данных, и было решено разделить данных на менее длинные предложения из обучающего корпуса.

3.1.4 Метрики оценки для распознавания речи

Для измерения качества системы распознавания казахской речи на основе внимания была использована метрика CER – количество неверно распознанных символов, потому что символы являются наиболее общими и простыми единицами вывода для генерации текстов. А также модель в этом исследовании оцениваются на основе коэффициента ошибок слов (WER), которая вычисляется с помощью расстояния Левенштейна [105] следующим образом (3.14):

$$WER/CER = \frac{D+S+I}{N} \cdot 100\% \quad (3.14)$$

здесь N — общее число слов или символов в данной последовательности, D — число удалений, S — количество замен, I— количество вставок.

3.2 Описание наборов данных, используемых в экспериментах

Для обучения модели был применен корпус в объеме 2000 часов смешанной речи. Для определения качества модели корпус был распределен на 2 части в соотношении 90% и 10% на тренировочную и тестоовую части соответственно. Использование повторных записей дикторов в тестовой части было исключено. В каждой части есть представители разного пола и возраста. Тренировочный и тестовый наборы не содержат идентичных предложений. Тестовый набор обеспечивает разнообразие слов и их фонетических контекстов и частоту встречаемости (таблица 3.2).

Таблица 3.2. Набор обучающих данных

Набор данных	**Продолжительность**
Общий объем корпуса	2000 часов
Тренировочная часть	1800 часов
Тестовая часть	200 часов

3.3 Алгоритм работы кодер-декодера с механизмом внимания

Этапы алгоритма интегрального распознавания речи, основанного на механизме внимания состоит из нескольких шагов, и представляют собой следующее:

1) выборка входных данных. На основе алгоритма MFCC звуковой сигнал преобразуется в вектора признаков $X = (x_1, \ldots x_n)$ в качестве входных данных системы.

2) обучение моделей, основанных на внимании. Во-первых, вводятся различные функции и механизмы для вычисления сходства или корреляции между входной последовательностью и целевой последовательностью. Затем вводится метод расчета softmax для преобразования оценки первого шага. С одной стороны, исходная рассчитанная оценка может быть нормализована до распределения вероятностей, где сумма весов всех элементов равна 1. С другой

стороны, можно выделить вес важных элементов с помощью внутреннего механизма softmax.

3) обучение и распознавание на основе интегральной модели кодер-декодера.

В модели внимания многомерные характеристики вводятся в нейронную сеть LSTM. В TensorFlow данные одного и того же пакета будут заполнены нулями в соответствии с самой длинной выборкой пакета, но, когда входные данные попадают в сеть LSTM, будет введена соответствующая длина последовательности каждой выборки в пакете. Таким образом, модуль LSTM будет вычислять только данные, соответствующие длине каждой выборки, а не последующей части нулевого заполнения, что минимизирует влияние нулевого заполнения на выходные данные и вес параметра нейронной сети. После того, как данные проходят через нейронную сеть LSTM, они проходят через слой полностью связанной нейронной сети, сопоставляя последнее измерение со всеми возможными метками классов. Затем данные поступают в модуль декодера, и предсказывают последовательность, соответствующую входным данным [106].

Алгоритм модели кодер-декодера с механизмом внимания

1: **procedure** EncAttDec(X, Z)
2: $U_0 \leftarrow$ {<sos>}
3: $U^{\wedge} \leftarrow \emptyset$
4: **for** l = 1 . . . Z **do**
5: $U_l \leftarrow \emptyset$
6: **while** $U_{l-1} \neq \emptyset$ **do**
7: hyp ← H (U_{l-1})
8: DECSEQ(U_{l-1})
9: **for each** cv ∈ V ∪ {<eos>} **do**
10: h ← hyp · cv
11: att(h, X)← att_i(h, X)
12: **if** cv = <eos> **then**
13: ENDSEQ($U^{\wedge}$, h)
14: **else**
15: ENDSEQ(Ul, h)
16: **return** arg $max_{h \in U^{\wedge}}$ att(h, X)
17: **end procedure**

Здесь U_l и $U^{\wedge}$ инициализируются в строках 2 и 3 алгоритма, которые реализованы как очереди, и принимают частичные гипотезы длины l и полные гипотезы соответственно. Каждая частичная гипотеза hyp в U_{l-1} расширяется каждой меткой cv в наборе меток V. Каждая расширенная гипотеза h оценивается в строке 11, где оценки внимания получаются с помощью att(). После этого, если cv =<eos>, гипотеза h считается завершенной и сохраняется в $U^{\wedge}$ в строке 13. Если с не пустая метка, то h сохраняется в U_l в строке 15. В строке 11 оценки модели внимания вычисляются для каждой частичной гипотезы.

Можно включить поиск луча, учитывая оценки внимания исключить частичные гипотезы с нерегулярным выравниванием и уменьшить количество ошибок поиска. Помимо этого, можно применить метод обнаружения конца, чтобы сократить вычисления, остановив поиск луча до того, как l достигнет Z.

Описание схемы распознавания речи на основе кодер-декодера с механизмом внимания приведена на рисунке 3.3.

Первый этап.

На вход поступает речевой сигнал, либо с микрофона либо заранее записанный аудиофайл формата .wav. Поступающие звуки предварительно сохраняются в отдельном файле в необходимом формате для дальнейшего распознавания.

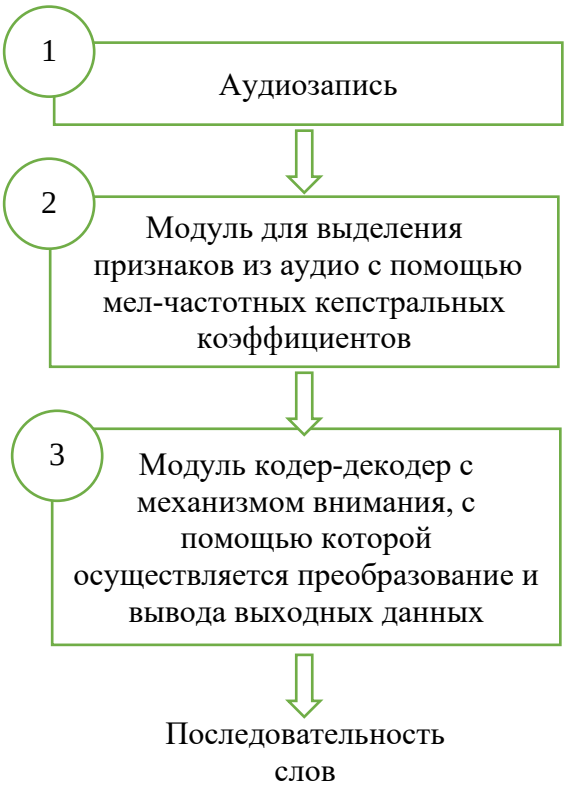


Рисунок 3.3 – Схема распознавания речи на основе кодер-декодера с механизмом внимания

Второй этап.

Входящий звук трансформируется в векторы признаков, а после этого будет произведена классификация. Данный этап включает в себя следующие действия:

– преобразование сигнала в цифровой формат;

– использование различных фильтров для устранения шумов;

– обозначение границ речи;

– извлечение признаков из сигнала.

Для выделения признаков применяется метод мел-частотных кепстральных коэффициентов (MFCC), который извлекает специфические параметры звуков речи посредством кепстрального анализа. Входной сигнал сначала формируется

и обрабатывается в виде фрейма, затем вычисляется с помощью преобразования Фурье, и величины результирующего спектра накладываются на шкалу Мел. Вычисляется энергия для каждого кадра. В итоге получаются значения мел-частотных кепстральных коэффициентов.

Полученный набор коэффициентов является входными данными для нейронной сети. При обучении сети, полученные данные передаются в блок обучения нейронной сети, где осуществляется обучение. После завершения обучения, нейронная сеть сохраняет свои навыки.

Второй этап алгоритма позволяет сократить время на преобразование в текст входящих аудиоданных.

Третий этап.

Используя полученные векторы признаков, нужно выяснить, какой звук или последовательность слов находилось в исходном сигнале. При работе нейронной сети в режиме распознавания данные переходят в блок интегральной модели кодер-декодера на основе механизма внимания, где происходит расшифровка речевого сигнала и вывод пользователю результата в текстовом виде.

Архитектура данной модели состоит из двух отдельных подсетей:

– подсеть кодера преобразует последовательность акустических признаков в промежуточное представление, длина которого соответствует длине входной последовательности;

– подсеть декодера предсказывает последовательность меток из промежуточной информации, предоставляемой подсетью кодера.

Длина декодированной метки, обычно меньше, чем длина входного признака. Декодер выбирает значимую часть векторов последовательных представлений для предсказания метки на каждом временном шаге с помощью механизма внимания и данный механизм предоставляет декодеру информацию о каждом скрытом состоянии кодера. Таким образом, декодер генерирует полученные промежуточные векторы в выходные последовательности. Результатом является слой softmax, вычисляющий распределение вероятностей по символам.

В качестве выхода извлекаются последовательности букв, складывающиеся в слова и фразы.

3.4 Программное и аппаратное обеспечения для реализации модели с вниманием

Программа была реализована на языке Python в среде Anaconda Navigator/Jupyter Notebook, который содержит программное обеспечение с открытым исходным кодом. Для проведения вычислений была применена библиотека программного обеспечения с открытым исходным кодом Tensorflow.

От характеристик центрального процессора и рабочей частоты оперативной памяти зависит скорость обучения модели. Для функционирования программы может быть использован процессор Pentium/AMD/Intel, при этом вычислительная машина, персональный компьютер или ноутбук должны

обладать оперативной памятью не менее 8 ГБ, объем свободного места на жестком диске – не меньше 500 МБ, операционные системы - Windows 8/10.

Учитывая размер набора данных и вычислительные требования архитектуры, для облегчения обучения потребовалось несколько графических процессоров (GPUs).

Для обучения модели на нескольких графических процессорах существуют два типа параллелизма: параллелизм данных и параллелизм моделей. Параллелизм данных фокусируется на сохранении копии архитектуры модели на каждом графическом процессоре, разделении большого пакета обучения между отдельными графическими процессорами, выполнении шагов прямого и обратного распространения для отдельных данных и, наконец, агрегировании обновлений градиента для всех моделей.

Параллелизм данных обеспечивает почти линейное масштабирование в зависимости от количества графических процессоров (это может повлиять на скорость сходимости из-за эффективного размера пакета). Второй тип параллелизма – это параллелизм моделей. Параллелизм моделей фокусируется на разделении слоев модели и распределении слоев по набору доступных графических процессоров. Включение параллелизма моделей может быть затруднено при работе с рекуррентными нейронными сетями из-за их последовательной природы. В архитектуре построенной модели был осуществлен параллелизм данных, распределив пакет данных для каждого графического процессора. Эти решения позволили тренироваться на 2000 часов звука и достичь самых современных результатов для распознавания казахской речи.

Программное обеспечение для распознавания речи не может правильно преобразовывать речь в текст, когда говорят 2 человека, особенно когда участники говорят одновременно и это приводит к наложению голосовых данных. В этом случае был применен микрофон SmartMike Duo, который может разделить это наложение на два отдельных аудиоканала (рис. 3.4).

Рисунок 3.4 – Интерфейс микрофона SmartMike Duo

Данный выбор микрофона обуславливается следующими преимуществами перед другими микрофонами (рис. 3.5):

• компактный и легкий - помещается в небольшие сумки;

• это единое устройство - подключи и работай, простое в использовании;

• обеспечивает естественное положение сидя (+ -45 ° / 0,5-1,5 м на расстоянии);

• естественное разделение голосовых динамиков (без искажений);

• индикация записи голоса с голосовым управлением;

• прочный корпус с глянцевой поверхностью и микрофонными решетками из нержавеющей стали;

• API для программного обеспечения для записи голоса для управления программным обеспечением или SmartMike.

SmartMike использует запатентованный искусственный интеллект для идентификации и разделения динамиков уже во время записи. Это обеспечивает четкую и точную расшифровку стенограммы для каждого человека, даже когда они говорили одновременно. Помимо записи встреч между коллегами, он также идеально подходит для любой ситуации с консультантом и клиентом, такой как юридическая консультация или встреча по заключению договора страхования.

Создан для четкой транскрипции и распознавания речи:

• микрофон студийного качества для четкой записи голоса;

• интеллектуальный микрофон AI для четкого разделения динамиков;

• отдельные аудиопотоки идеально подходят для точной транскрипции и качества распознавания речи.

SmartMike поставляется со встроенным фильтром шумоподавления и двумя микрофонами студийного качества, обеспечивающими максимально четкую запись голоса и наиболее точные результаты распознавания речи.

Запатентованные алгоритмы искусственного интеллекта, работающие внутри устройства, позволяют разделить два динамика по линии даже в более шумной среде. Это обеспечивает более естественное качество голоса для отличных результатов распознавания речи.

SmartMike позволяет точно записывать данные с двух динамиков с помощью всего одного устройства. Микрофон можно расположить на столе естественным образом, чтобы он не мешал разговору. Благодаря компактным размерам его также можно легко транспортировать или брать с собой на встречи с клиентами.

Отдельные потоки аудио-голосовых данных позволяют легко и быстро транскрибировать, используете ли вы машинистку или преобразование речи в текст. Даже отрывки, в которых оба говорящих говорили одновременно, легко различимы. Это гарантирует, что важная информация не будет потеряна.

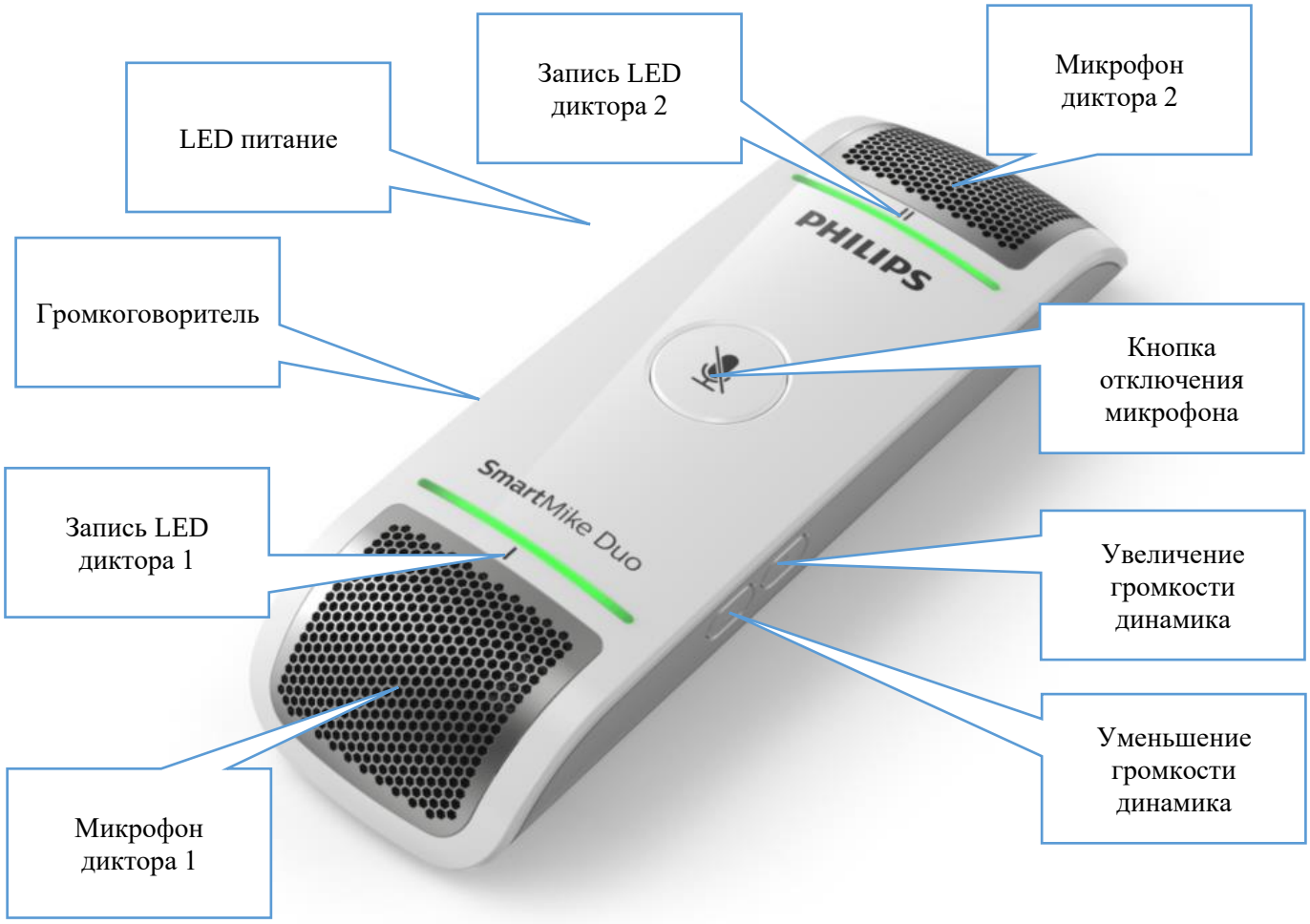


Рисунок 3.5 – Микрофон SmartMike Duo

Можно подойти ближе или дальше от микрофона без ущерба для качества. Микрофон позволяет поддерживать естественное и комфортное поведение при встрече и разговоре. Нет необходимости прямо говорить в микрофон.

SmartMike Duo оснащен двумя светодиодными индикаторами, которые четко сигнализируют, включен микрофон или отключен. Во время разговора зеленый свет сигнализирует о том, что голос соответствующего говорящего в данный момент записывается. Входящий в комплект громкоговоритель обеспечивает надежное и четкое воспроизведение.

3.5 Эксперименты с разными наборами данных с моделью внимания

Исследовательские работы проводились в лаборатории Компьютерной инженерии интеллектуальных систем при Институте Информационных и Вычислительных Технологий. Был использован суперкомпьютер с графическими процессорами AMD Ryzen 9 с GeForce RTX3090. Обучение заняло 1,5 дня. Обучающие данные хранились на 1000 GB SSD, для обеспечения быстрого потока данных во время обучения.

3.5.1 Настройка и обучение интегральной модели

Определяем гиперпараметры – размер пакета, начальная скорость обучения и градиентный спуск:

```
# Hyper-parameters
nm_epochs = 100
nm_hidden = 8
nm_layers = 3
batch_size = 64
init_learn_rate = 1e-2
dropout = 0.95
```

Для настройки процесса обучения доступны следующие основные гиперпараметры:

1. nm_epochs – число эпох. С увеличением числа эпох, веса нейронной сети изменяются все большее количество раз. Кривая с каждый разом лучше подстраивается под данные, переходя последовательно из плохо обученного состояния (последний график) в оптимальное (центральный график). Для различных датасетов оптимальное количество эпох будет отличаться. Но ясно, что количество эпох связано с разнообразием в данных.

2. nm_hidden – число нейронов в скрытых слоях нейронной сети. Данный параметр подбирается в зависимости от объема обучающих данных и его разнообразия.

3. nm_layers – число слоев нейронной сети.

4. batch_size – количество пакетов в одном батче для обучения сети. Выбирается максимальное значения исходя из объема аудиопамяти. Обычно используют значения от 8 до 64.

5. init_learn_rate – скорость обучения, позволяющий управлять величиной коррекции весов на каждой итерации

5. dropout_rate - вероятность исключения нейрона из вычислений во время шага обучения, что позволяет избежать перереобучения сети. Стандартное значение: 0.3-0.5.

Для улучшения модели были подобраны значения параметров, которые влияют на время и качества обучения сети. Для предотвращения переобучения модели и для ускорения обучения сети был установлен batch_size со значением 128, также были опробованы параметры batch_size со значениями 1, 4, 8, 16, 32, 64. Необходимо отметить, что минимальные значения batch_size не давали никакого эффекта при обучении. Коэффициент начальной скорости обучения был установлен 0,001. Использовался Dropout для каждого выхода рекуррентного слоя в качестве регуляризации и равен 0,5. Для нашей модели использовали алгоритм оптимизации градиентного спуска на основе Адама [107].

Словарь букв для казахского языка составила 44 символа с добавлением дополнительных меток, которые помогают определить начало и конец слова/предложения.

Мы применили модель к различным наборам данных. Модель была обучена на 126-часовом, 500-часовом, 2000-часовом корпусе казахской речи.

3.5.2 Полученные результаты в ходе исследования интегральной модели

В настоящее время, очень мало исследований по распознаванию казахской речи на основе интегральных моделей. Есть, конечно, работы, которые были реализованы на основе традиционных и гибридных моделей, но были использованы меньший объем тренировочных данных.

Были рассмотрены работы [108, 47, 55] связанные с распознаванием казахской речи на основе интегрального и гибридного подхода для сравнения результатов. В работе [47] была реализована модель CTC с интегрированной внешней языковой модели. Проведены эксперименты с разными архитектурами нейронных сетей. Объем обучающих данных составило 123 часов речи.

В работе [108] ИНС была обучена на основе 36-часового корпуса казахского языка. Применен гибридный модель, DNN с использованием 6 скрытых слоев.

Модель обучалась на данных 300-часового корпуса казахской речи в работе [19]. Была реализована на основе моделей Transformer с CTC. Также была использована внешняя RNN LM на этапе декодирования с помощью мелкого синтеза.

В [55] был реализован метод трансферного обучения, который берет предварительно обученную на русском корпусе модель для построения модели для распознавания казахской речи. Модель CTC была использована в качестве функции потерь. Объем корпуса составило 20 часов речи.

Результаты экспериментов по распознаванию речи с помощью разных моделей и модели с использованием кодер-декодера приведены в таблице 3.3.

В [109] был реализован кодер-декодер с механизмом внимания и применили модель к различным наборам данных. Разработанный корпус состоит из смешанного набора данных. Модель была обучена на 126-часовом, 500-часовом, 2000-часовом и расширенном корпусе казахской речи. Влияние размера набора данных можно увидеть из рисунка 3.6.

Для дальнейшего улучшения качества системы APP была реализована модель Transformer совместно с CTC. Обычные интегральные кодер-декодер модели для задач распознавания речи состоят из одного кодера и декодера, механизма внимания. А в модели Transformer может быть несколько кодеров и декодеров, и каждый из них может содержать свой внутренний механизм внимания. Обычно CTC используется в качестве функции потерь для обучения рекуррентных нейронных сетей для распознавания входной речи без предварительного выравнивания входных и выходных данных. Для достижения высокой производительности из модели CTC необходимо применять внешнюю языковую модель, так как прямое декодирование будет работать некорректно. Кроме того, казахский язык имеет довольно многообразный механизм словообразования, что использование ЯМ способствует к повышению качества распознавания казахской речи. Таким образом, использование CTC с LM при декодировании привело к быстрой сходимости модели, что уменьшило

количество времени для декодирования и улучшило показатели системы. CTC функция после получения выходных данных от кодера находит вероятность распределения меток для произвольного выравнивания между выходом кодера и выходной последовательности символов.

На основе проделанных экспериментов было выявлено, что совместное использование моделей Transformer и CTC способствовало улучшению показателей системы распознавания казахской речи при обучении и тестировании на основе чистых и полностью размеченных данных.

Таблица 3.3 – Полученные результаты модели на основе интегрального подхода

Model	**WER, %**	**CER, %**	**Объем данных, часов**
Mamyrbayev et al. (2020) (без ЯМ			
MLP	59.26	48.11	126
LSTM	46.51	36.43	
Conv+LSTM	39.31	34.92	
BLSTM	37.66	33.61	
ResNet)	36.57	32.52	
(с ЯМ			
MLP	63.26	39.11	
LSTM	46.51	24.43	
Conv+LSTM	39.31	22.92	
BLSTM	20.66	13.61	
ResNet)	19.57	11.52	
Mamyrbayev et al. (2019) (DNN-HMM) с ЯМ	31.78	–	76
Amirgaliyev et al. (2020) (LSTM)	–	24	20
(BLSTM)	–	32	
Khassanov et al. (2020) (Transformer + CTC) с ЯМ	12.4	3.9	300
Mamyrbayev, Oralbekova (2021) (Encoder-attention-decoder)	19.46	13.83	126
обученная расширенным корпусом			
(Encoder-attention-decoder) без ЯМ	15.46	9.83	500
(Encoder-attention-decoder) без ЯМ	12.24	7.21	2000
(Encoder-attention-decoder) с ЯМ	10.17	4.35	2000
Mamyrbayev, Oralbekova (2022) Transformer + CTC LM (обученная на основе чистых данных)	8.3	3.7	2000
Transformer + CTC LM (обученная на основе чистых и неразмечанных данных)	9.6	15.8	

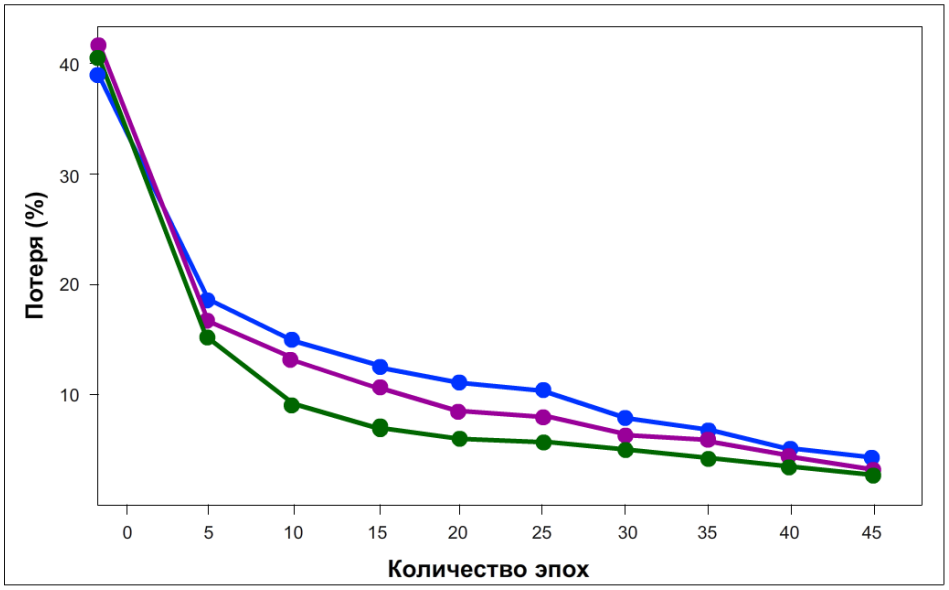


Рисунок 3.6. Сравнение training loss и CER с тремя наборами данных. Синим цветом выделено training loss на 126-часовом корпусе, фиолетовым цветом – на 500-часовом корпусе, а зеленым цветом – на расширенным основным корпусом.

3.5.3 Анализ и сравнение полученных результатов интегральной модели

Во время эксперимента имеющий корпус был расширен данными ISSAI Kazakh Speech Corpus https://issai.nu.edu.kz/kz-speech-corpus/, что привело к улучшению значений CER, и потребовалось больше времени для обучения системы. Сравнительный графический анализ полученных результатов (рис. 3.7) работ показывает, что модель кодер-декодер на основе механизма внимания без использования языковой модели не смогла превзойти модель на основе Transformer и CTC [19], это связано тем, что модель кодер-декодер с механизмом внимания был обучен на смешанном наборе данных, а для модели в [19] был применен чистый набор обучающих данных, что привело к уменьшению показателей CER и WER. А также необходимо отметить, что в [19, 47, 55] применили полностью размеченные и чистые наборы данных для обучения и тестирования системы APP.

Добавление не размеченных и частично размеченных данных, а также аудиоданных телефонных разговоров в корпус речи привело к ухудшению качества системы APP.

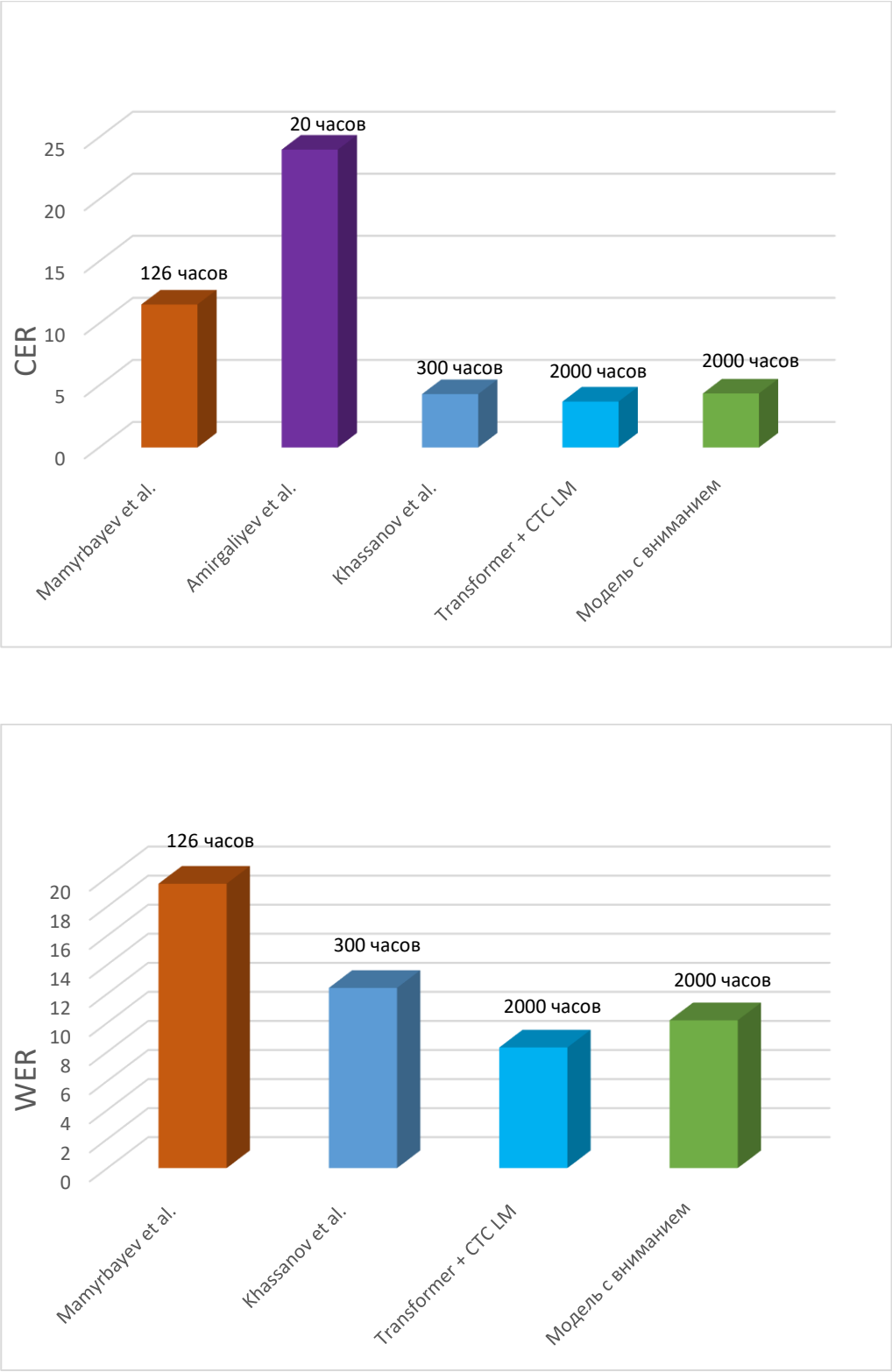


Рисунок 3.7 – Сравнительная диаграмма работ и разработанной модели по показателям CER и WER

В таблице 3.4 продемонстрированы некоторые результаты работы системы. Ошибки больше связаны с тем, что в качестве выхода были взяты последовательности букв, а не слов.

Таблица 3.4 – Выходные данные модели

Input	адамның заттарға қатынасы да аса маңызды
Decoded	адамның заттарға атынасы да аа маңызды
Input	бұл барыста ниет шамасында өзгерістер болуы мүмкін
Decoded	бұл барыста нет шамасында өзгерістер болу мүмккн
Input	ол кейде күшейюі кейде әлсіреуі мүмкін
Decoded	о кейде күшеюі кейде әсере мүмкін

Число итераций эксперимента было сначала установлено на 100 шагов, но, поскольку модель не сходилась на таком количестве шагов, было решено увеличение шагов до несколько ста итераций. Выходные данные были словами вместо символов и декодирование осуществлялось с помощью алгоритма beam search, перебирались параметры со значениями от 5 до 30. После применения необходимых действий, был получен результат, показанный в таблице 3.5. В лучшем случае система достигла 10,17% WER, что по сравнению с результатами других исследований, имеет самый лучший показатель без применения внешней языковой модели.

На основе полученных результатов можно сделать вывод, что модель кодер-декодер с механизмом внимания может увеличить качества распознавания и без использования внешних языковых моделей и показала конкурентноспособный результат для всех наборов данных на казахском языке.

Таблица 3.5. Выходные данные модели в расширенном наборе данных

Input	дегенмен бұлай түсіндіру салыстырмалы
Decoded	дегенмен бұндай түсіндіру салыстырмалы
Input	онда міндеттің өзі құштарлық және шаттық туғызады
Decoded	онда міндетті өз құштарлық және шаттық туғызады
Input	нақты және орташа күн тәулігі
Decoded	нақты және орташа күн тәулік

Данная модель не подходит для реализации модели распознавания потоковой речи, как RNN-Transducer и Neural Transducer. Но тем не менее предложенная модель может в дальнейшем улучшена в связи появлением новых методов и технологий.

3.6 Выводы

В этой главе подробно рассматривается модель кодер-декодер с механизмом внимания, а именно

1. описана архитектура и виды механизмов внимания модели кодер-декодер с вниманием. Для построения архитектуры кодер-декодер модели были применены нейронные сети, как LSTM, MLP и BLSTM;
2. описана предварительная настройка интегральной модели для кодера, декодера и механизма внимания, а также приведены метрики оценки для распознавания речи, как CER – количество неверно распознанных символов, и

WER – количество неверно распознанных слов, которые вычисляются с помощью расстояния Левенштейна;

3. приведено описание наборов данных, используемых в экспериментах. В корпусе звуковые файлы были разделены на тренировочную и тестовую части и составляет 90% и 10% всего корпуса соответственно;

4. разработан алгоритм работы кодер-декодер с механизмом внимания и приведено подробное описание алгоритма;

5. приведены программное и аппаратное обеспечения для реализации модели с вниманием. Программа была реализована на языке Python. Для проведения вычислений была применена библиотека программного обеспечения с открытым исходным кодом Tensorflow. Кроме того, приведено описание микрофона SmartMike Duo, который может разделить наложение двух отдельных аудиоканала, в случае если говорят 2 человека, особенно когда участники говорят одновременно и это приводит к наложению голосовых данных;

6. описана экспериментальная проверка предложенной интегральной модели. Для сравнения полученных результатов были отобраны исследовательские работы, связанные с распознаванием речи на казахском языке. Полученные данные исследований продемонстрировали, что реализованная модель улучшила качество распознавания и без применения языковых моделей для казахского языка и превзошла гибридные модели на основе глубоких нейронных сетей, и интегральные модели, которые были обучены меньшим объемом речевых данных. Модель с вниманием показала лучшие результаты по распознаванию казахской речи по точности распознавания символов на 4.35% без внешней ЯМ. А также было продемонстрировано, что увеличение обучающих данных намного улучшило качество системы распознавания казахской речи.

4 РЕАЛИЗАЦИЯ ИНТЕГРАЛЬНОЙ СИСТЕМЫ РАСПОЗНАВАНИЯ КАЗАХСКОЙ РЕЧИ

Система автоматического распознавания речи на казахском языке на основе интегрального подхода была на Python 3.8 в графическом интерфейсе Anaconda Navigator/Jupyter Notebook, с применением библиотеки Tensorflow.

4.1 Общее описание интегральной системы на основе внимания

Система, реализованная на основе модели кодер-декода с вниманием преобразует речевой сигнал в посделовательности слов без дополнительных вычислений на серверах в офлайн и онлайн режимах. Кроме того, технология учитывает различия в акценте, синтаксисе, местных выражениях и различных способах произнесения одних и тех же вещей в каждом отдельном языке. Данную систему APP можно использовать для диктовки в таких отраслях, как здравоохранение, где врачи могут делать точные голосовые заметки на государственном языке и т.д.

Программное обеспечение состоит из следующих частей:

– Модуль для обучения нейронных сетей;

– Модуль для валидации и вывода данных системы.

Модуль для обучения нейронных сетей представляет собой модель интегральной архитектуры с алгоритмами оптимизации градиентного спуска. Кроме того, данный модуль включает все настройки работы для эффективного обучения сети, который выявляет сложные зависимости между входными и выходными данными, а также выполняет обобщение.

Модуль для валидации системы обрабатывает набор аудио данных в текстовые данные и выдаёт показатель качества системы распознавания речи. Модель в этой системе оценивается на основе коэффициента ошибок слов (WER), которая вычисляется с помощью расстояния Левенштейна. Модуль вывода преобразовывает входящий сигнал в последовательность слов.

Подробное описание этих блоков приведены в следующих разделах.

4.2 Элементы интегральной системы распознавания речи

Система автоматически преобразует исходный аудиофайл в машиночитаемый формат, т.е. текст, без использования дополнительных компонентов и транскрипций. Данная система является одним из вариантов современных систем распознавания речи, разработанных с использованием интегрального глубокого обучения. Осуществление архитектуры интегральной модели происходит значительно проще, чем классические системы, которые строятся на основе независимых друг от друга компонентов. При построении интегральной системы можно обойтись без компонентов ручной обработки для моделирования фонового шума, реверберации или вариаций динамика.

Интегральные системы включают в себя хорошо оптимизированную систему обучения рекуррентной нейронной сети (RNN) [109], которая может использовать несколько графических процессоров, а также набор новых методов синтеза данных, которые позволяют эффективно получать большой объем

различных данных для обучения. Интегральные системы также могут справляться со сложной шумной средой лучше, чем широко используемые современные коммерческие традиционные речевые системы.

Для работы с программой распознавания речи на вход подается записанный аудиофайл в формате .wav. При выборе режима обучения нейронной сети, сначала происходит извлечение речевых признаков методом мел-частотных кепстральных коэффициентов, после чего преобразованный звук подается на вход нейронной сети, где происходит её обучение посредством корректировки весов. После завершения обучения, нейронная сеть сохраняет свои навыки. При работе модуля в режиме распознавания происходит загрузка обученной нейронной сети, а из поступившего аудиосигнала происходит извлечение речевых признаков методом мел-частотных кепстральных коэффициентов. Преобразованный сигнал интерпретируется нейронной сетью и расшифровывается с помощью кодер-декодер модели с механизмом внимания. Результат расшифровки выводится в виде текста на экран пользователя.

Основные технические характеристики

– Тип программы: 64-битовое приложение с графическим пользовательским интерфейсом;

– Название продукта: DuoConversationUSB ASR;

– Имя исполняемого файла: DuoConversationUSB.exe;

– Размер исполняемого файла: 7,51 Мбайт.

4.2.1 Модуль для обучения нейронных сетей

Данный модуль включает в себя: получение входной акустической последовательности, представляющей речь, и входной акустической последовательности, представляющей соответствующее представление акустического признака на каждом из первого числа временных шагов; обработку входной акустической последовательности с использованием первой нейронной сети для преобразования входной акустической последовательности в альтернативное представление для входной акустической последовательности, выделение вектора контекста начального внимания токена, при этом строка содержит набор букв алфавита. В модуле сформированная последовательность строк начинается с начала токена последовательности <sos> и заканчивается концом токена последовательности <eos>. Первая нейронная сеть является пирамидальной двунаправленной долгосрочной краткосрочной памятью (BLSTM); обработка входной акустической последовательности с использованием первой нейронной сети для преобразования входной акустической последовательности в альтернативное представление входной акустической последовательности включает: обработку входной акустической последовательности через нижний слой BLSTM для генерации выходного слоя BLSTM и обработку выходного слоя BLSTM через каждый из множества слоев пирамиды BLSTM, при этом последовательные выходные данные каждого пирамидного слоя BLSTM объединяются перед предоставлением следующему слою пирамиды BLSTM; обработка альтернативного представления для входной

акустической последовательности, осуществляемая второй нейронной сетью включает для начальной позиции в порядке выходной последовательности: генерацию вектора контекста внимания, обработку маркера начала последовательности токена и вектора контекста начального внимания токена для обновления скрытого состояния из исходного, генерацию набора оценок строк для начальной позиции с использованием вектора контекста; кроме того данный процесс включает выбор строки с наивысшей оценкой из набора оценок строки в качестве строки в начальной позиции в выходной последовательности строк, а обработка альтернативного представления для входной акустической последовательности включает: обработку строки в предыдущей позиции в порядке выходной последовательности и вектора контекста внимания для предыдущей для обновления скрытого состояния; генерацию вектора контекста внимания для позиции из альтернативного представления и скрытого состояния и генерацию набора оценок строк для позиции с использованием вектора контекста внимания для позиции и скрытого состояния. Модуль также включает выбор строки с наивысшей оценкой из набора оценок строки для позиции в качестве строки в позиции в выходной последовательности строк, а генерация вектора контекста включает: вычисление скалярной энергии для позиции с использованием альтернативного представления для позиции; преобразование вычисленной скалярной энергии в распределение вероятностей с использованием функции softmax, и использование распределения вероятностей для создания текстового вектора путем комбинирования альтернативного представления в различных позициях. Формирование набора подстрочных оценок для позиции с использованием вектора контекста для позиции включает: обеспечение скрытого состояния второй нейронной сети для позиции и генерируемого вектора контекста внимания для позиции в качестве входных данных для многослойного персептрона (MLP) с выходным слоем softmax; и обработку скрытого состояния генерируемого вектора контекста внимания для позиции с использованием MLP, при этом первую и вторую нейронные сети обучают совместно, обработка альтернативного представления для входной последовательности использование поискового декодирования луча слева направо, а обработка альтернативного представления для входной акустической последовательности осуществляется с использованием второй нейронной сети для генерации каждой позиции в порядке выходной последовательности набора оценок строки, включающей соответствующую оценку строк в наборе; при этом происходит генерация последовательности строк, представляющих транскрипцию высказывания [110].

4.2.2 Модуль для валидации и вывода данных системы

Система содержит интерфейс, хранящий инструкции, для выполнения операций, включающие: получение входной акустической последовательности, входную акустическую последовательность, представляющую речь, и входную акустическую последовательность, содержащую соответствующее представление акустического признака на каждом из первого числа временных

шагов; обработка входной акустической последовательности с использованием первой нейронной сети для преобразования входной акустической последовательности в альтернативное представление для входной акустической последовательности, при этом альтернативное представление входной акустической последовательности содержит соответствующее альтернативное представление акустических характеристик для каждого из второго числа временных шагов, при этом второе число меньше первого числа; обработку альтернативного представления входной акустической последовательности с использованием второй нейронной сети для генерации для каждой позиции в порядке выходной последовательности набора оценок строки, включающего соответствующую оценку подстрок для каждой строки в набор строк; и генерацию последовательности строк представляющих транскрипцию высказывания, а набор строк содержит набор букв алфавита, который используется для записи одного или нескольких языков.

Структура системы распознавания казахской речи

Структура системы распознавания казахской слитной речи (рис. 4.1) на основе интегрального (end-to-end) подхода реализован следующим образом.

Блок 1 – На вход поступает речевой сигнал, либо с микрофона, либо заранее записанный аудиофайл формата .wav. Поступающие звуки предварительно сохраняются в отдельном файле в необходимом формате для дальнейшего распознавания.

Блок 2 – осуществляет преобразование аудиофрагмента в цифровую форму с параметром частоты дискретизации равным 16000 Гц для выделения векторов признаков из речевого сигнала (аудиофайла). Далее сигнал разбивают на короткие фреймы длиной 10 мс и применяют оконная функция Хеннинга. После этого, для получения спектра сигнала каждого фрейма применяют быстрое преобразование Фурье. Потом вычисляют кепстр и осуществляют наложение на него мел-частотных окон, а также вычисляют энергию для каждого кадра. Далее после применения функции дискретного косинусного преобразования, в итоге получают значения мел-частотных кепстральных коэффициентов [111]. Полученный набор коэффициентов является входными данными для нейронной сети.

Блок 3 – кодер преобразует последовательность акустических признаков в промежуточное представление. В качестве кодера обычно используется двунаправленная сеть LSTM – BiLSTM. Также использовался метод объединение между несколькими слоями BiLSTM для ускорения модели за счет объединения во времени, что может уменьшить длину входной последовательности. Временной шаг кодера между верхним и нижним уровнями был уменьшен на пропорцию 1/2, а декремент может быть реализован либо с помощью смежного суммирования, либо с помощью простой подвыборки.

Эта модификация требуется поскольку входные речевые сигналы могут иметь длину от сотен до тысяч кадров. Прямое применение BiLSTM для кодера работает медленно и не дает хороших результатов. Это связано с тем, что

механизм внимания с трудом извлекает релевантную информацию из большого количества временных шагов ввода.

Блок 4 – механизм внимания выявляет важные факторы из обучающей выборки, которые помогают декодеру снизить ошибку сети. Выявление релевантной информации реализовывается с помощью метода обратного распространения ошибки.

Элемент внимание генерирует матрицу весов важности. Когда мы обучаем сеть на данных, важность становится функцией вероятности того или иного исхода в зависимости от переданный на вход сети данных.

Механизм внимания помогает проектировать сеть, которая способна к изучению последовательностей на базе полносвязной сети, обучить её на GPU, применять dropout для устранения переобучения сети. И включение механизма в модель существенно повышает качество ее работы.

Блок 5 – декодер генерирует полученные через кодер промежуточные векторы представления в выходные последовательности.

Во время вывода необходимо найти наиболее вероятную последовательность символов с учетом входной акустики.

Декодирование выполняют с помощью простого алгоритма поиска луча слева направо. Мы поддерживаем набор частичных гипотез, начиная с токена <sos> в начале предложения. На каждом временном шаге каждая частичная гипотеза в луче расширяется со всеми возможными символами, и сохраняются только наиболее вероятные лучи. Когда токен <eos> встречается, он удаляется из луча и добавляется к набору полной гипотезы.

Блок 6 – для улучшения результатов была использована функция softmax в качестве входных данных для предсказания следующего шага. Затем баллы выравнивания нормализуются с использованием данной функции. Результатом является слой softmax, вычисляющий распределение вероятностей по символам.

Блок 7 – в качестве выхода извлекают последовательности букв, складывающиеся в слова и фразы [110].

Предлагаемые способ и устройство для его реализации решают проблемы с нормализацией:

Если имеется зашумленная информация, из которой извлекаются много незначимые векторы признаков, как известно, для механизма внимания сложно сосредотачивать свое внимание на нескольких соответствующих кадрах каждый раз при декодировании. И данный процесс требует, чтобы механизм внимания рассматривал все кадры каждый раз, когда он декодирует одиночный выходной сигнал при декодировании выходного сигнала, что приводит к вычислительной сложности.

Исследуется техника «окна», в котором механизм внимания рассматривает только подпоследовательность векторов признаков всей последовательности промежуточных представлений, заранее заданной шириной окна. Оценки для промежуточных представлений и подпоследовательность векторов признаков не вычисляются, что снижает вычислительную сложность системы. И данная техника используется во время обучения сети внимания.

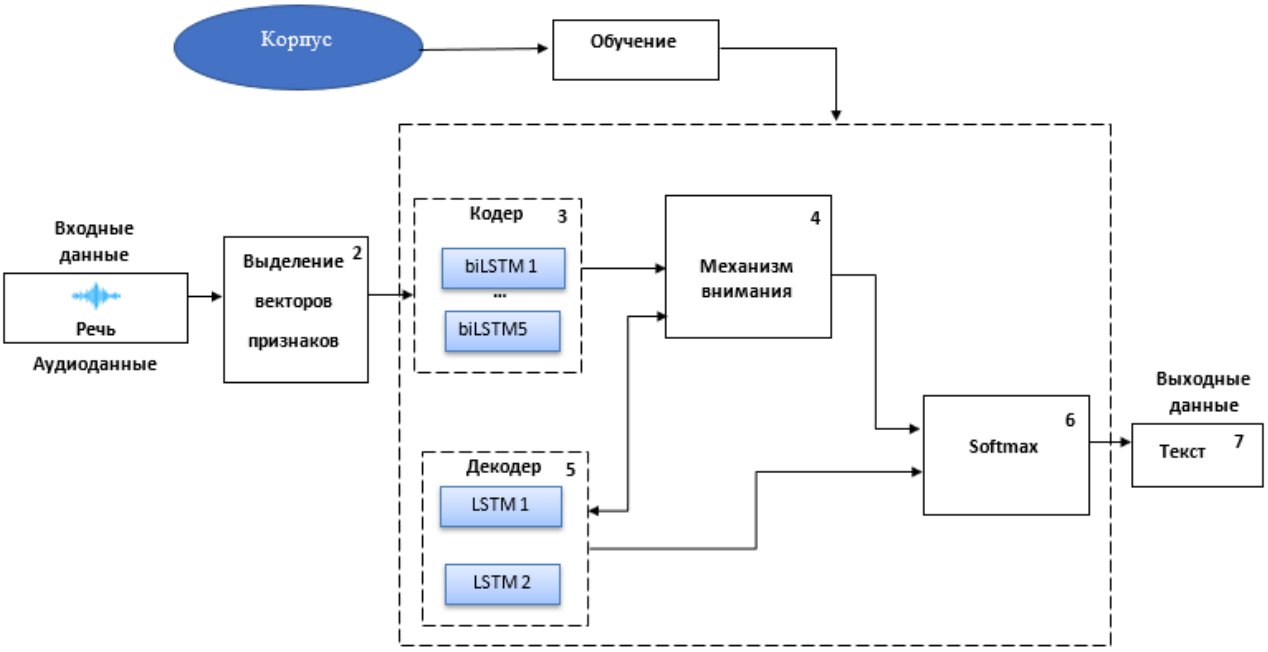


Рисунок 4.1 – Структура интегральной системы

Помимо этого, в моделях, где выбор кадра с наивысшей оценкой, отрицательно сказываются на характеристиках модели в стандартном наборе для разработки, который в основном состоит из коротких высказываний. Это позволяют нам предположить, что для модели полезно агрегировать выборки из нескольких фреймов с наивысшими оценками. Для этого экспоненциальная функция в механизме внимания была заменена логистическим сигмоидом, что является методом сглаживания.

Кроме того, исследования показывают, что гибридный механизм внимания является лучшим кандидатом для распознавания речи на агглютинативных языках, так как данный вид механизма устраняет те ограничения, связанные как с механизмами, основанными на содержании, так и на основе местоположения.

4.3 Описание и реализация интегральной системы распознавания речи

Интерфейс программы, приведенный на рисунке 4.1, имеет достаточно простой дизайн и состоит из двух блоков: блок 1, где находится кнопка записи речи, блок 2 – это Speaker 1 и Speaker 2.

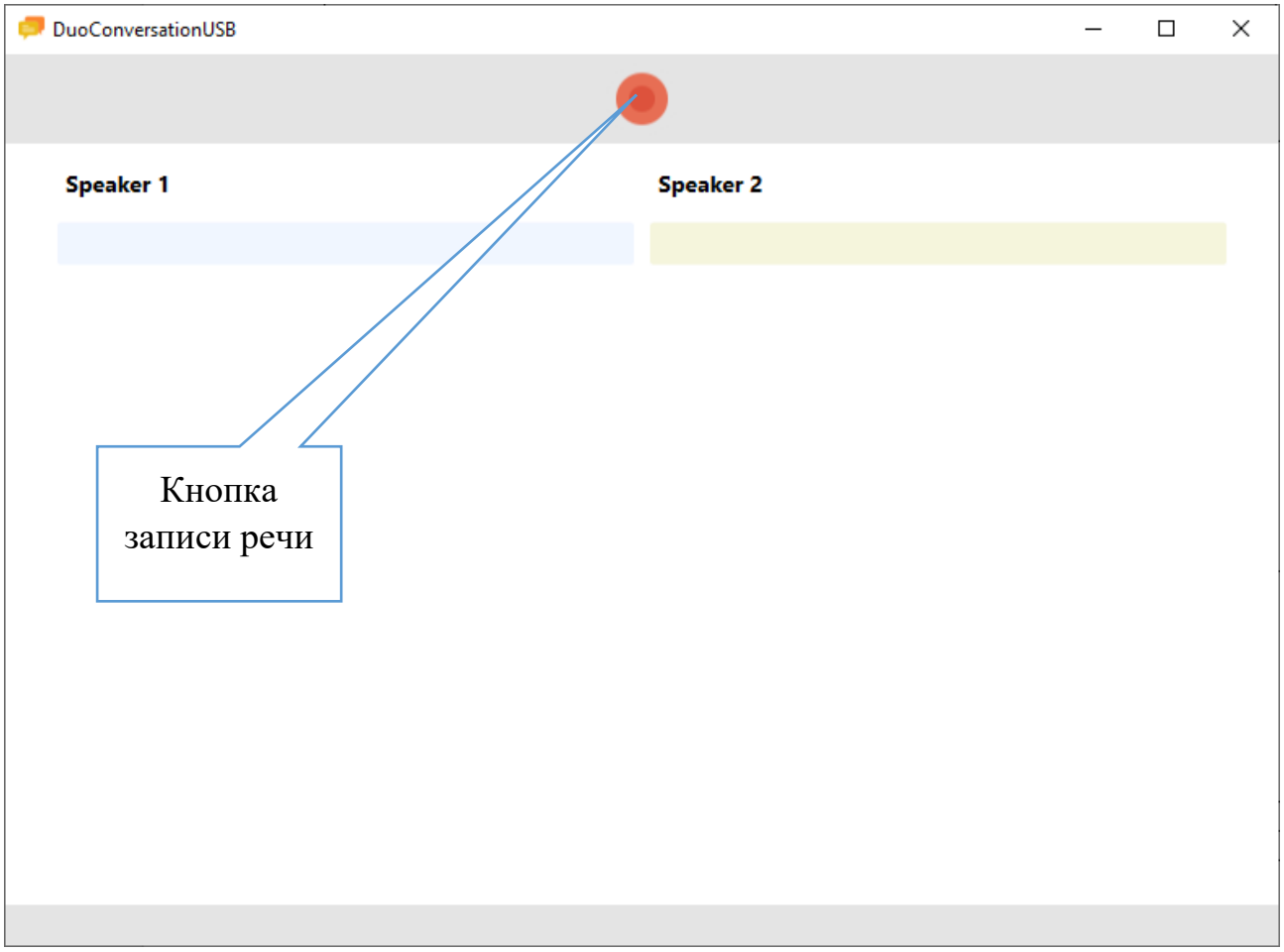


Рисунок 4.1 – Интерфейс программы DuoConversationUSB ASR

Для того чтобы записать речь необходимо нажать на красную кнопку, которая находится в центре на верхнем углу программы. После произнесенных слов программа автоматически преобразовывает речь в текст, и при этом разделяя речь дикторов по своему блоку (рис. 4.2.).

После этого могут быть записаны все сказанные дикторами аудиоречи в отдельной папке (рис. 4.3.).

Обработка речи в текст не занимает память компьютера и не требует больших вычислений на серверах. Данная программа работает локально и не требует подключения к Интернету.

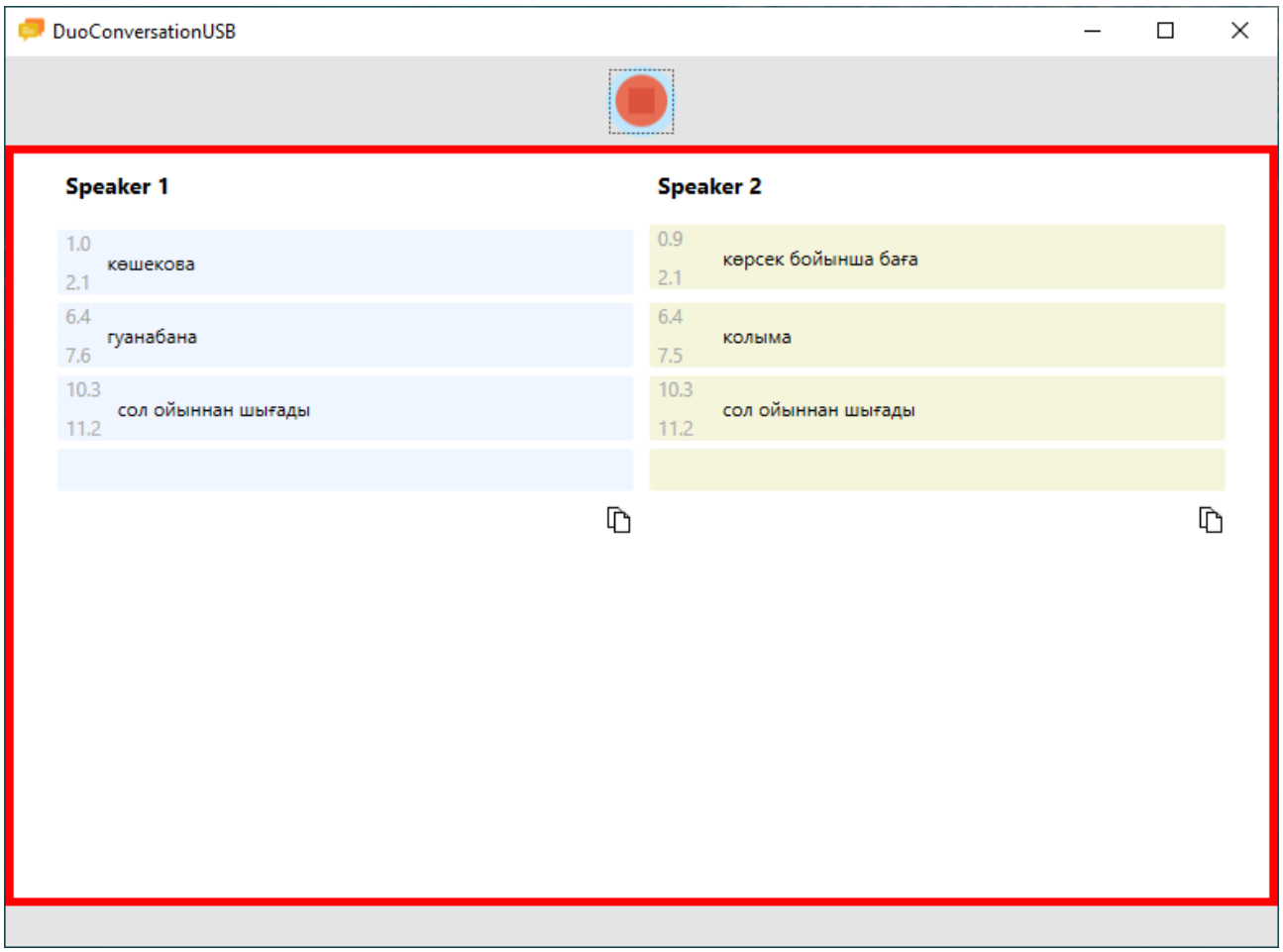


Рисунок 4.2 – Начало записи разговора дикторов

После нажатия кнопки все аудиоданные записываются в папке «Recordings» и в дальнейшем эти данные могут быть включены в корпус речи.

Процесс тестирования системы автоматического распознавания казахской речи и микрофона SmartMike можно найти по следующей ссылке: https://youtu.be/6eG1zRx-lEY.

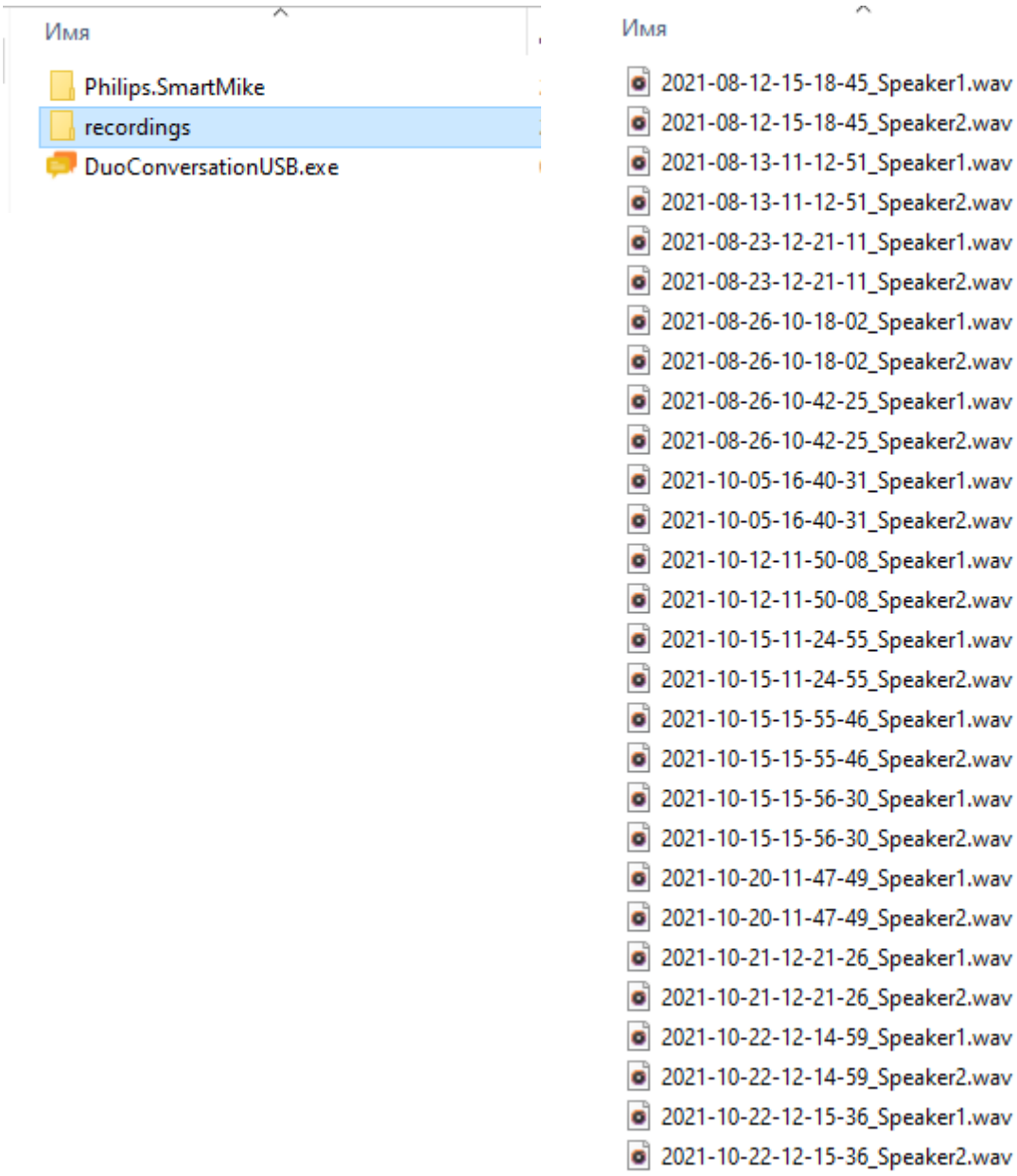


Рисунок 4.3 – Записи всех разговоров

4.4 Интеграция системы распознавания интегральной системы с микрофоном

Интеграция микрофона Wireless SmartMike Duo предоставляет уникальную возможность разработанной системе использовать два отдельных аудиоканала, что обеспечивает превосходную точность распознавания и расширенные возможности анализа речи (рис. 4.4).

Пример использования интеграции

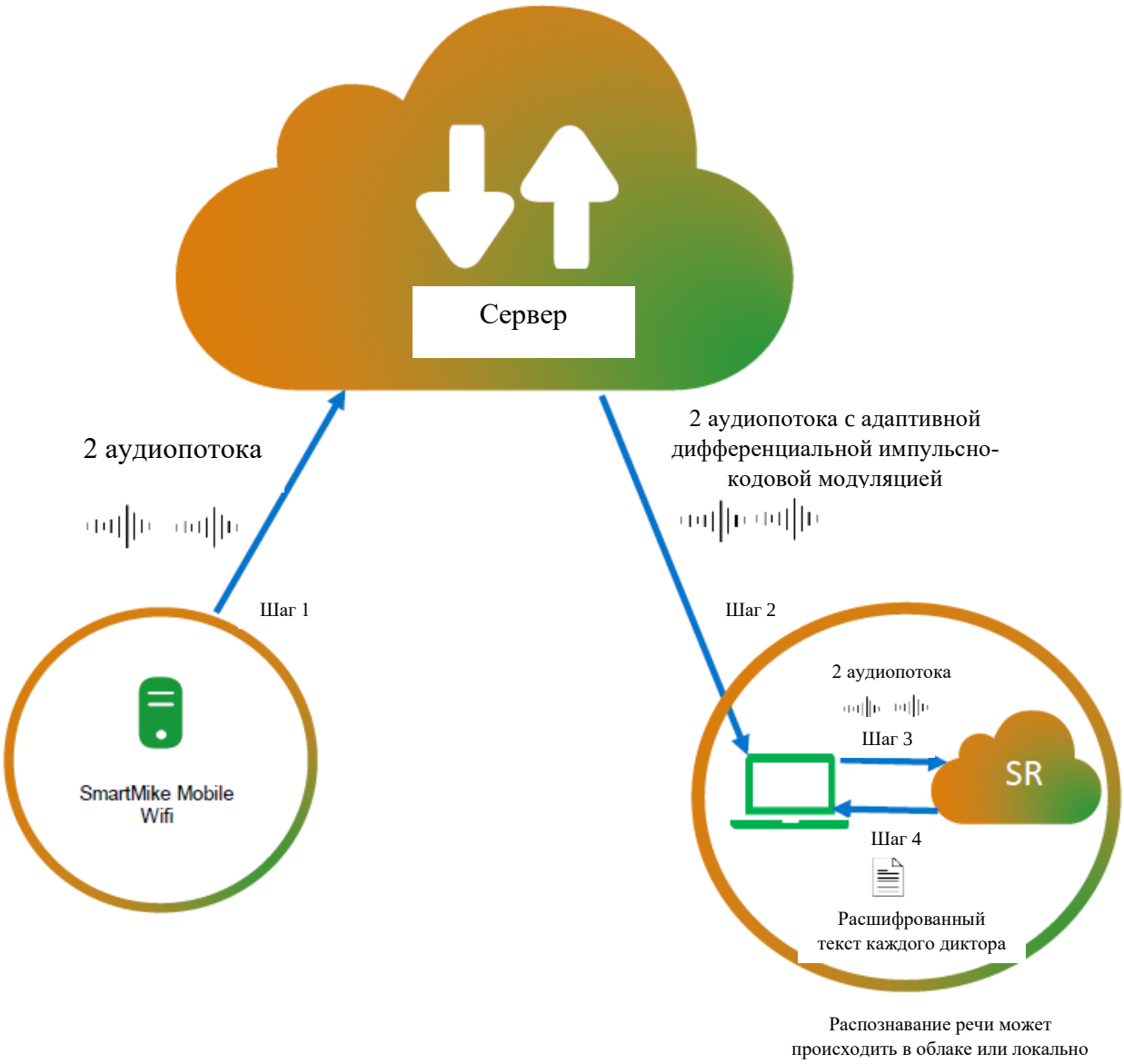
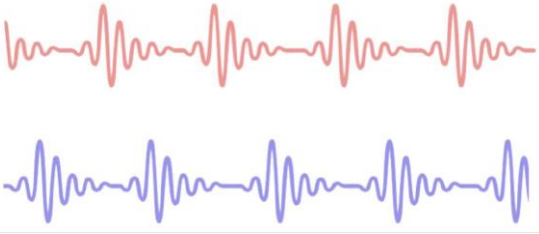


Рисунок 4.4 – Схема интеграции микрофона Wireless SmartMike с системой

2 человека разговаривают одновременно, что приводит к наложению голосовых данных:  .

С помощью устройства SmartMike Duo можно разделить это наложение на 2 отдельных аудиоканала: 

Всегда лучшее качество звука – даже с 2х динамиками (рис. 4.5):

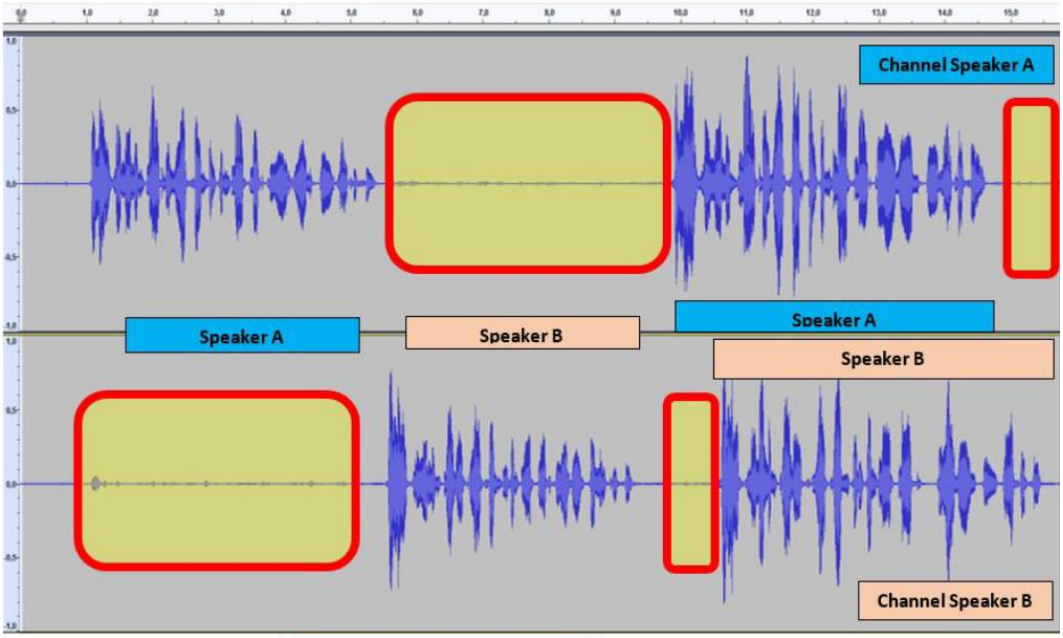


Рисунок 4.5 – Разделение аудио сигнала двух дикторов с SmartMike Duo

Для интеграции интегральной системы с микрофоном SmartMike Duo USB PSM1010 были проделаны некоторые действия:

1) Был загружен последний официальный пакет SDK по ссылке https://www.dictation.philips.com/us/products/workflow-software/sdk-for-dictation-hardware-lfh7445/. Для этого необходимо было создать учетную запись.

В пакете SDK есть полнофункциональное тестовое приложение и исходный код, который можно использовать в качестве справочника.

2) После этого создается новый проект в Visual Studio. Были выбраны WPF и C #. В этой работе было использовано Visual Studio 2019.

Нужно определиться со степенью интеграции. Любой процессор может не всегда работать должным образом. Была выполнена компиляция для x86, поскольку оно работает на всех платформах Windows.

Нужно установить элемент управления SmartMikeSDK COM в системе разработки, запустив SmartMikeCtrl32.msi из \ x86 \ Зарегистрированная установка \ Redist. Был установлен элемент управления COM с разрядностью, выбранной для проекта интеграции.

3) Добавила в свой проект ссылку на \ x86 \ Зарегистрированная установка \ Redist \ bin \ PIA.SmartMikeCtrl.dll. Можно также напрямую ссылаться на SmartMikeCtrl.dll, но рекомендуется использовать PIA.

4) Чтобы упростить задачу, в начало кода было добавлено следующее:

```
using PIA.SmartMikeCtrl;
```

5) Теперь можно объявить объект SDK

SmartMikeControl _ctrl;

и создать

```
_ctrl = new SmartMikeControl();
```

Нужно создать экземпляр try-catch для отображения потенциальных исключений. Все ошибки, которые может вызывать SDK, доставляются в виде исключений, поэтому нужно создать каждый вызов SDK блоком try catch.

6) Создаю кнопку для инициализации SDK. В обработчике событий кнопки инициализирую объект SDK следующим образом:

```
_ctrl.Initialize();
```

7) Запускаю свой код, подключаю SmartMike PSM1010, наблюдаю, как светодиоды меняют цвет с оранжевого на выключенный вскоре после того, как была нажата кнопка «Инициализировать». Теперь устройство разблокировано. Он останется заблокированным, если нет приложения, работающего с интеграцией SDK. Заблокировано означает, что устройство будет генерировать шум только в виде аудиопотоков. Разблокирован означает, что устройство воспроизводит один аудиоканал на динамик, идеально разделенный алгоритмом искусственного интеллекта, работающим на устройстве.

8) Далее добавляю простую конфигурацию устройства. Надо использовать программное обеспечение для отключения звука подключенного устройства. Создаем для этого кнопку. В обработчике добавляю следующее:

```
ISmmDeviceConfiguration devConfig = _ctrl.DeviceConfiguration[0];
devConfig.MicrophoneStatus = smmMicrophoneStatus.smmMicrophoneStatusMuted;
devConfig.UploadConfiguration();
```

9) Кроме того, мы хотим получать уведомления, как только устройство будет подключено. Для этого нужно указать это сразу после создания экземпляра:

```
_ctrl.SMMEventDeviceConnected += _ctrl_SMMEventDeviceConnected;
```

10) И создаю обработчик событий следующим образом:

```
private void _ctrl_SMMEventDeviceConnected(int InstanceID)
{
MessageBox.Show("Device connected with instance ID" + InstanceID);
}
```

11) Приложение, которое было создано, будет работать на любом устройстве, на котором установлен 32-битный SmartMike SDK. Интегратор может связать установку SDK со своей собственной установкой, используя \ [bitness] \ Registered installation \ Redist \ Windows Installer \ Philips_SmartMikeSDK_ [bitness] _COM.msm. Однако, если интегратор хочет использовать SDK параллельно, то нужно воспользоваться следующим решением: параллельная установка - это на самом деле не установка, а только копирование файлов, и она имеет смысл (или, лучше сказать, обязательна), если сама интеграция является параллельным приложением.

12) Чтобы интеграция работала параллельно, необходимо добавить манифест в проект Visual Studio. Для этого добавляю манифест из тестового приложения, представленного в \ [bitness] \ Side-by-side installation \ TestApplication \ CSharp \ TestSmartMikeCtrl \ app_ [bitness] .manifest.

13) Теперь нужно установить манифест в Project Properties -> Application -> Resources -> Manifest.

14) После создания решения помещаю папку с именем Philips.SmartMike, содержащую SmartMikeSDK, рядом с исполняемым файлом интеграции. Снова копирую папку из тестового приложения. Далее запускаю исполняемый файл.

15) При использовании SDK параллельно, все, что нужно сделать интегратору для развертывания SDK со своим приложением, - это скопировать папку Philips.SmartMike рядом с исполняемым файлом.

4.5 Выводы

Последняя глава диссертационной работы посвящена описанию системы автоматического распознавания казахской речи. В ней

1. рассмотрена подробная структура интегральной системы, описаны работы модулей для обучения нейронных сетей и для валидации и вывода данных системы;
2. приведен интерфейс программы, который имеет достаточно простой дизайн и состоит из двух блоков: блок 1, где находится кнопка записи речи, блок 2 – это Speaker 1 и Speaker 2;
3. описан процесс интеграции интегральной системы с микрофоном SmartMike Duo, который предоставляет уникальную возможность разработанной системе использовать два отдельных аудиоканала. Это обеспечивает превосходную точность распознавания и расширенные возможности анализа речи. Кроме того, с помощью устройства SmartMike Duo можно разделить звуковые потоки двух дикторов на 2 отдельных аудиоканала, что повышает точность распознавания казахской речи.

ЗАКЛЮЧЕНИЕ

В ходе выполнения диссертационной работы по исследованию и разработке модели, архитектуры и эффективного алгоритма для повышения точности распознавания слитной казахской речи на основе интегрального подхода были получены следующие научные и практические результаты:

1. Проведен аналитический обзор современных методов и подходов, состояние и перспективы развития распознавания речи на основе интегральной архитектуры. Выполнен сравнительный анализ интегральных моделей с традиционными и гибридными моделями.

2. Выполнен сбор информационных данных и запись казахской речи для корпуса языка, а также был расширен корпус казахской речи и корпус текстов на казахском языке. Разработана методика транскрибирования телефонных разговоров.

3. Разработана модель кодер-декодер с модифицированным механизмом внимания для корректного распознавания схожих окончаний в словах. Были получены результаты интегральной системы для распознавания казахской речи для разного набора данных. Был разработан эффективный алгоритм для повышения точности распознавания речи.

4. Разработано и реализовано программное обеспечение, с помощью которого можно получить качественную трансформацию речевой информации на казахском языке в последовательность слов, не прибегая к анализу звучания речи

По результатам диссертационной работы были получены 3 авторских свидетельства на программные обеспечения: 1) Система автоматического распознавания казахской речи на основе интегральной архитектуры. 2) Система идентификации и аутентификации через речевые технологии. 3) Система автоматического распознавания казахской слитной речи на основе модели с механизмом внимания. А также был получен акт о внедрении результатов диссертационного исследования в продукцию компании “Philips”. Помимо этого, был получен патент на изобретение «Система и способ распознавания агглютинативной слитной речи на основе интегрального (end-to-end) подхода».

Поставленные задачи диссертационной работы полностью решены, и полученные результаты могут быть использованы для компьютерного стенографирования, для голосового управления компьютером, робототехническими и автоматизированными системами на государственном языке, что позволит людям с ограниченными возможностями одновременно осуществлять несколько функций, не связанных с устройствами ввода в машину.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Karanasou P. Phonemic variability and confusability in pronunciation modeling for automatic speech recognition. – Paris: Université Paris Sud-Paris XI, 2013. – P. 83-86.

2 Juang, B., Rabiner, L. Hidden Markov Models for Speech Recognition // Technometrics. – 1991. – Vol. 33, № 3. – P. 251-272.

3 Brown J., Smaragdis P. Hidden Markov and Gaussian mixture models for automatic call classification // The Journal of the Acoustical Society of America. – 2009. – V. 125, № 6. – P. 221-224.

4 Hinton G., Deng L., Yu D., Dahl G., Mohamed A., Jaitly N., Senior, A., Vanhoucke V., Nguyen P., Sainath T., Kingsbury B. Deep Neural Networks for Acoustic Modeling in Speech Recognition // Signal Processing Magazine, IEEE. – 2012. – Vol. 29, № 6. – P. 2-17.

5 Ghaffarzadegan, S., Boril, H., Hansen, J. Deep neural network training for whispered speech recognition using small databases and generative model sampling // International Journal Speech Technology. – 2017. – Vol. 20. – P. 1063–1075.

6 Yu D., Li J. Recent progresses in deep learning based acoustic models // IEEE/CAA Journal of Automatica Sinica. – 2017. – Vol. 4, № 3. – P. 396-409.

7 Ko W., Tseng B., Lee H. Recurrent Neural Network based language modeling with controllable external Memory // 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA. – 2017. – P. 5705-5709.

8 Sundermeyer M., Schluter R., Ney H. LSTM neural networks for language modeling // Interspeech. – 2012. P. 256-268.

9 Jaitly N., Hinton, G. Learning a better representation of speech sound waves using restricted Boltzmann machines // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. – 2011. – P. 5884 - 5887.

10 Amodei D. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin // Proceedings of the 33rd International Conference on Machine Learning, in Proceedings of Machine Learning Research. –2016. – Vol. 48. – P.173-182.

11 Shan C., Weng C., Wang G., Su D., Luo M., Yu, D., Xie L. Investigating end-to-end speech recognition for mandarin-English code-switching // 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2019. – P. 6056-6060.

12 Тампель И.Б., Карпов А.А. Автоматическое распознавание речи. Учебное пособие. – Санкт-Петербург: Университет ИТМО, 2017. – 152 с.

13 Кипяткова И. С., Карпов А. А. Исследование нейросетевых моделей русского языка для систем автоматического распознавания слитной речи // Автоматика и телемеханика. – 2017. – № 5. – С. 110-122.

14 Марковников Н.М., Кипяткова И.С. Аналитический обзор интегральных систем распознавания речи // Тр. СПИИРАН. – 2018. – Т. 58. – С. 77-110.

15 Шарипбаев А.А., Бекманова Г.Т., Алтынбек Г., Адалы Е., Жеткенбай Л., Каманур У. Единый морфологический анализатор для казахского и турецкого языков // Turklang. – Астана, 2017. – С. 234-240.

16 Есенбаев Ж.А. Распознавание казахской речи по определенной словарной базе в условиях шумов: автореф.: 6D060200 – Информатика. – Евразийский национальный университет им. Л. Н. Гумилева. – Астана, 2014. – 175 с.

17 Арпачиев А.К., Тукеев У.А. Предварительная обработка речевых сигналов для выделения информативных признаков в системах распознавания речи // Вестник КазНУ им. аль-Фараби. – Алматы: Қазақ университеті, 2002, – № 3 (31). – С. 132-136.

18 Zhumanov Z., Madiyeva A., Rakhimova D. New Kazakh Parallel Text Corpora with On-line Access // Computational Collective Intelligence. Lecture Notes in Computer Science. – 2017. – Vol. 10449. – P. 501-508.

19 Khassanov Y., Mussakhojayeva S., Mirzakhmetov A., Adiyev A., Nurpeiissov M., Varol, H.A. A crowdsourced open-source Kazakh speech corpus and initial speech recognition baseline //Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics. – 2021. – Main volume. – P. 697-706.

20 Amirgaliyev Y., Hahn M., Mussabayev T. The speech signal segmentation algorithm using pitch synchronous analysis // Open Computer Science. – 2017. – P. 1-8.

21 Mussabayev R., Kalimoldayev M., Amirgaliyev Y., Mussabayev T. Automatic speech segmentation using throat-acoustic correlation coefficients // Open Engineering. – 2016. – Vol. 6. – P. 335-346.

22 Казачкин А. Е. Методы распознавания речи, современные речевые технологии // Молодой ученый. – 2019. – № 39. – С. 6-8.

23 El-Emary I., Fezari M., Atoui H. Hidden Markov model/Gaussian mixture models (HMM/GMM) based voice command system: A way to improve the control of remotely operated robot arm TR45 // Scientific research and essays. – 2011. – Vol. 6. – P. 341-350.

24. Hannun A., Case C., Casper J., Catanzaro B., Diamos G., Elsen E., Prenger R., Satheesh S., Sengupta S., Coates A., Ng A. DeepSpeech: Scaling up end-to-end speech recognition // ArXiv. – 2014. – Vol. 2. – P. 1-12.

25 Mamyrbayev, O., Oralbekova, D., Keylan, A. A study of transformer-based end-to-end speech recognition system for Kazakh language // Scientific Reports. – 2022. – Vol. 12. – P. 183–194.

26 Glasmachers T. Limits of End-to-End Learning // ACML. – 2017. –Vol. 77. – P. 17-32.

27 Ронжин А.Л., Карпов А.А., Ли И.В. Речевой и многомодальный интерфейсы // М.: Наука, 2006. – 173 с.

28 Ibrahim M., Walid I. Khedr, Osama M. Elkomy, Al-Zahraa M.I. Abdalla. Recognition of phonetic Arabic figures via wavelet-based Mel Frequency Cepstrum using HMMs // HBRC Journal. – 2014. – Vol. 10, № 1. P. 49–54.

29 Bhadragiri Jagan M., Ramesh Babu N. Speech recognition using MFCC and DTW // 2014 International Conference on Advances in Electrical Engineering (ICAEE). – 2014. – P. 1–4.

30 Dave N. Feature extraction methods LPC, PLP and MFCC in speech recognition // International Journal For Advance Research in Engineering And Technology (ISSN 2320-6802). – 2013. – Vol. 1, № 6. – P. 1-5.

31 Hu Y., Guizhong L. Dynamic characteristics of musical note for musical instrument classification // 2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). – 2011. – P. 1–6.

32 Ernawan F., Abu N., Suryana N. Spectrum analysis of speech recognition via discrete Tchebichef transform // Proceedings of SPIE - The International Society for Optical Engineering. – 2011. – Vol. 8285. – P. 1-8.

33 Pan S., Chen C., Zeng J. Speech recognition via Hidden Markov Model and neural network trained by genetic algorithm // 2010 International Conference on Machine Learning and Cybernetics. – 2010. – P. 2950–2955.

34 Li L. Hybrid Deep Neural Network--Hidden Markov Model (DNN-HMM) Based Speech Emotion Recognition // 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction. – 2013. – P. 312–317.

35 Deng L. Deep learning: From speech recognition to language and multimodal processing // APSIPA Transactions on Signal and Information Processing. – 2016. – Vol. 5 – P. 7-22.

36 Abdel-Hamid O., Mohamed A., Jiang H., Deng L., Penn G., Yu D. Convolutional Neural Networks for Speech Recognition // IEEE/ACM Transactions on Audio, Speech, and Language Processing. – 2014. – Vol. 22, №. 10. –P. 1533-1545.

37 Hasim Sak Andrew W. Senior Kanishka Rao Françoise Beaufays. Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition // CoRR. – 2015. – Vol. 1. – P. 42-47.

38 Lu, L.; Zhang, X.; Cho, K.; Renals, S. A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition // In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association. – Dresden, Germany, 2015. – P. 3249–3253.

39 Rao K., Sak H., Prabhavalkar R. Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer // Proceedings of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). – Okinawa, Japan, 2017. – P. 193–199.

40 Оралбекова Д.О., Мамырбаев О.Ж. Современные методы распознавания речи // Новости науки Казахстана. – 2021. Т. 1, № 148. – С. 20-35.

41 Vegesna V.V.R., Gurugubelli K., Vydana H.K., Pulugandla B., Shrivastava M., Vuppala A.K. DNN-HMM Acoustic Modeling for Large Vocabulary Telugu Speech Recognition // Lecture Notes in Computer Science. – 2017. – Vol. 10682. – P. 3-19.

42 Edmondo T., Marco G. A survey of hybrid ANN/HMM models for automatic speech recognition // Neurocomputing. – 2001. – Vol. 37, № 1–4. – P. 91–126.

43 Zhang Y., Zhang P., Li T. Yan Y. An unsupervised vocabulary selection technique for Chinese automatic speech recognition //2016 IEEE Spoken Language Technology Workshop (SLT). – 2016. – P. 420–425.

44 Deroo O., Christophe R., Stéphane D. Context dependent hybrid HMM/ANN systems for large vocabulary continuous speech recognition system // EUROSPEECH – 1999. – P. 211-222.

45 Oludare I., Aman J., Abiodun E., Kemi V., Nachaat A., Humaira A. State-of-the-art in artificial neural network applications: A survey // Heliyon. – 2018. – Vol. 4, № 11. P. 2405-8440.

46 Graves A., Fernández S., Gomez F., Schmidhuber J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks // Proceedings of the 23rd International Conference on Machine Learning. – 2006. – P. 369-376.

47 Mamyrbayev O., Alimhan K., Zhumazhanov B., Turdalykyzy T., Gusmanova F. End-to-End Speech Recognition in Agglutinative Languages // Proc. of the 12th Asian Conference on Intelligent Information and Database Systems (ACIIDS). – Phuket, Thailand, 2020. – Vol. 12034 – P. 391-401.

48 Ito H., Hagiwara A., Ichiki M., Mishima T., Sato S., Kobayashi A. End-to-end speech recognition for languages with ideographic characters // 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). – 2017. – P. 1228-1232.

49 Nguyen V. An End-to-End Model for Vietnamese Speech Recognition // 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF). – 2019. – P. 1–6.

50 Ahmed A., Hifny Y., Shaalan K.F., Toral S. End-to-End Lexicon Free Arabic Speech Recognition Using Recurrent Neural Networks. Computational Linguistics // Speech and Image Processing for Arabic Language. – 2018. – P. 93-97.

51 Bataev V., Korenevsky M., Medennikov I., Zatvornitsky A. Exploring End-to-End Techniques for Low-Resource Speech Recognition // SPECOM. – 2018. – P. 32-41.

52 Rustamov S., Akhundova N., Valizada A. Automatic Speech Recognition in Taxi Call Service Systems. In: Miraz M., Excell P., Ware A., Soomro S., Ali M. Emerging Technologies in Computing // Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. – 2019. – Vol. 285. – P. 243-253.

53 Kannan A., Datta A., Sainath, T., Weinstein E., Ramabhadran B., Wu Y., Bapna A., Chen Z., Lee S. Large-Scale Multilingual Speech Recognition with a Streaming End-to-End Model // Interspeech – 2019. – P. 2130–2134.

54 Suyanto S., Arifianto A., Sirwan A., Rizaendra A. P. End-to-End Speech Recognition Models for a Low-Resourced Indonesian Language // 2020 8th International Conference on Information and Communication Technology (ICoICT). – Indonesia: Yogyakarta. – 2020. – P. 1–6.

55 D. Kuanyshbay, Y. Amirgaliyev, O. Baimuratov. Development of Automatic Speech Recognition for Kazakh Language using Transfer Learning // International

Journal of Advanced Trends in Computer Science and Engineering. – 2020. – Vol. 9, P. 5880-5886.

56 Мамырбаев О.Ж., Оралбекова Д.О., Кыдырбекова А.С., Жумажанов Б.Ж., Тұрдалықызы Т. Интегральная гибридная модель на основе CTC и механизма внимания для распознавания казахской слитной речи // Международная научно-практическая конференция "Сатпаевские чтения - 2021". – Алматы, 2021. – С. 48-52.

57 Chan W., Jaitly N., Le Q., Vinyals O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition // 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016 – P. 4960-4964.

58 Dong L., Zhou S., Chen W., Xu B. Extending Recurrent Neural Aligner for Streaming End-to-End Speech Recognition in Mandarin // ArXiv – 2018. – Vol. abs/1806.06342. – P. 11-16.

59 Hori T., Watanabe S., Zhang, Y., Chan, W. Advances in Joint CTC-Attention based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM // Interspeech – 2017. – P. 17-22.

60 Kim S., Hori T., Watanabe S. Joint CTC-attention based end-to-end speech recognition using multi-task learning // 2017 IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP. – 2017. –P. 4835–4839.

61 Miao H., Cheng G., Zhang P., Yan Y. Online Hybrid CTC/Attention End-to-End Automatic Speech Recognition Architecture // IEEE/ACM Transactions on Audio, Speech, and Language Processing. – 2020. – Vol. 28. – P. 1452–1465.

62 Markovnikov N., Kipyatkova I. Investigating Joint CTC-Attention Models for End-to-End Russian Speech Recognition // Speech and Computer. – 2019. – P. 337-347.

63 Yan Y., Prieto R., Wang B., Zhou J., Gu Y., Liu Y., Lin H. Attention-based sequence-to-sequence model for speech recognition: development of state-of-the-art system on LibriSpeech and its application to non-native English // ArXiv. — 2018. – Vol. abs/1810.13088. – P. 42-56.

64 Bahdanau, D.; Chorowski, J.; Serdyuk, D.; Brakel, P.; Bengio, Y. End-to-end attention-based large vocabulary speech recognition // Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 2016. – P. 4945–4949.

65 Ruchao F., Zhou P., Chen W., Jia J., Liu G. An Online Attention-based Model for Speech Recognition // ArXiv. – 2019. Vol. abs/1811.05247 – P. 56-67.

66 Qin C., Zhang, W., Qu D. A new joint CTC-attention-based speech recognition model with multi-level multi-head attention // Audio speech music proc. – 2019. – Vol. 18. – P. 19-32.

67 Miao, H., Cheng, G., Gao, C., Zhang, P., Yan, Y. Transformer-Based Online CTC/Attention End-To-End Speech Recognition Architecture // 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2020. – P. 6084–6088.

68 Wu L., Li T., Wang L., Yan Y. Improving Hybrid CTC/Attention Architecture with Time-Restricted Self-Attention CTC for End-to-End Speech Recognition // Applied Science – 2019. – Vol. 9, 4639. – P. 1-14.

69 Chen J., Nishimura R., Kitaoka N. End-to-end recognition of streaming Japanese speech using CTC and local attention // APSIPA Transactions on Signal and Information Processing. – 2020. – Vol. 9. – P. 1-7.

70 Park H. Korean Grapheme Unit-based Speech Recognition Using Attention-CTC Ensemble Network // 2019 International Symposium on Multimedia and Communication Technology (ISMAC). – 2019. – P. 1-5.

71 Alsayadi H., Abdelhamid A., Hegazy I., Fayed Z. Arabic speech recognition using end-to-end deep learning // IET Signal Processing. – 2021. – P. 521-534.

72 Emiru E., Li Y., Fesseha A., Diallo M. Improving Amharic Speech Recognition System Using Connectionist Temporal Classification with Attention Model and Phoneme-Based Byte-Pair-Encodings. Information. – 2021. – Vol. 12. 62. – P. 1-22.

73 Lafferty J., McCallum A., Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data // Proceedings of the International Conference on Machine Learning (ICML'01). – Williamstown, MA, USA, 2001. – P. 282–289.

74 Fosler-Lussier E., He Y., Jyothi P., Prabhavalkar R. Conditional random fields in speech, audio, and language processing // Proceedings of the IEEE. – 2013. –Vol. 101, № 5. – P. 1054–1075.

75 Марковников Н.М., Кипяткова И.С. Аналитический обзор интегральных систем распознавания речи // Тр. СПИИРАН. – 2018. – Том 58. – С. 77–110.

76 Hifny Y., Renals S. Speech recognition using augmented conditional random fields // IEEE Transactions on Audio, Speech, and Language Processing. – 2009. – Vol. 17, № 2. – P. 354–365.

77 Tang H. End-to-End Neural Segmental Models for Speech Recognition // IEEE Journal of Selected Topics in Signal Processing. – 2017. – Vol. 11, № 8. – P. 1254–1264.

78 An K., Xiang H., Ou Z. CAT: CRF-based ASR Toolkit // ArXiv. – 2019. Vol. abs/1911.08747. – P.1-15.

79 An K., Xiang H. CAT: A CTC-CRF based ASR Toolkit Bridging the Hybrid and the End-to-end Approaches towards Data Efficiency and Low Latency // Interspeech. – 2020. – P. 1-9.

80 Lu L., Kong L., Dyer C., Smith N. A. Multitask Learning with CTC and Segmental CRF for Speech Recognition // Interspeech. – 2017. – P. 1-12.

81 Xiang H., Ou Z. CRF-based single-stage acoustic modeling with CTC topology // 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2019. – P. 5676–5680.

82 Zheng H., Peng W., Ou Z., Zhang J. Advancing CTC-CRF Based End-to-End Speech Recognition with Wordpieces and Conformers // ArXiv. – 2021. – Vol. abs/2107.03007 – P. 1-8.

83 Li Y., Li Y., Wang J., Tang Z. Post Text Processing of Chinese Speech Recognition Based on Bidirectional LSTM Networks and CRF // Electronics 8. – 2019. – Vol. 11. – P. 1-17.

84 Graves A., Mohamed A., Hinton, G. Speech Recognition with Deep Recurrent Neural Networks // ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing. – 2013. – P. 6645-6649.

85 Graves A. Sequence transduction with recurrent neural networks // ArXiv. – 2012. – Vol. 1211.3711. – P. 1-9.

86 He Y. Streaming End-to-end Speech Recognition for Mobile Devices // ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – Brighton, United Kingdom, 2019. – P. 6381–6385.

87 Prabhavalkar R., Rao K., Sainath T. N., Li B., Johnson L., Jaitly N. A comparison of sequence-tosequence models for speech recognition // Interspeech. – 2017. – P. 1-13.

88 Wang S., Zhou P., Chen W., Jia J., Xie L. Exploring RNN-Transducer for Chinese speech recognition // 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). – Lanzhou, China, 2019. – P. 1364–1369.

89 Battenberg E. Exploring neural transducers for end-to-end speech recognition // 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) – 2017. – P. 206–213.

90 I. Sklyar, Piunova A., Zheng X., Liu Y. Multi-turn RNN-T for streaming recognition of multi-party speech // ArXiv. – 2021. – Vol. 2112.10200. – P. 1-5.

91 Prabhavalkar R., Rao K., Sainath T., Li B., Johnson L., Jaitly N. A Comparison of Sequence-to-Sequence Models for Speech Recognition // Interspeech. – 2017. – P. 939–943.

92 Rao K. Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer // 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE. – 2017. – P. 193–199.

93 Wang S., Zhou P., Chen W., Jia J., Xie L. Exploring RNN-Transducer for Chinese speech recognition // 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) – Lanzhou, China, 2019. – P. 1364–1369.

94 Li J., Hu H., Gong Y. Improving RNN Transducer Modeling for End-to-End Speech Recognition // 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) – 2019. – P. 114–121.

95 Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. Attention is all you need //Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc. – NY, USA: Red Hook, 2017. – P. 6000–6010.

96 Moritz N., Hori T., Le J. Streaming Automatic Speech Recognition with the Transformer Model // ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2020. – P. 6074–6078.

97 Shi Y., Wang Y., Wu C., Fuegen C., Zhang F., Le D., Yeh C., Seltzer M.. Weak-attention suppression for transformer-based speech recognition // ArXiv. – 2020. – Vol. abs/2005.09137. – P. 1-15.

98 Dong L., Xu S., Xu B. Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition // 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) – 2018. – P. 5884–5888.

99 Di G., Negri M., Cattoni R., Dessì R., Turchi M. Enhancing Transformer for End-to-end Speech-to-Text Translation // Proceedings of Machine Translation Summit XVII: Research Track. – 2019. – P. 21–33.

100 Hori T., Moritz N., Hori C., Le Roux J. Transformer-based Long-context End-to-end Speech Recognition // Interspeech. – 2020. – P. 5011-5015.

101 Chang X., Zhang W., Qian Y., Le Roux J., Watanabe S. End-to-end multi-speaker speech recognition with transformer // ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) – 2020. – P. 6134-6138.

102 Karita S., Soplin N., Watanabe S., Delcroix M., Ogawa A., Nakatani T. (2019). Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration // Proceedings of the Annual Conference of the International Speech Communication Association, Interspeech. – 2019. – P. 1408–1412.

103 Georgescu, AL., Pappalardo, A., Cucu, H. Performance vs. hardware requirements in state-of-the-art automatic speech recognition // J AUDIO SPEECH MUSIC PROC. – 2021. – Vol. 28. – P. 1-30.

104 Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. Attention-based models for speech recognition // Advances in Neural Information Processing Systems. – 2015. – P. 577-585

105 Levenshtein V.I. Binary codes capable of correcting deletions, insertions, and reversals // Soviet physics. Doklady. – 1996. – Vol. 10. – P. 707–710.

106 Watanabe S., Hori T., Kim S., Hershey J. R., Hayashi T. Hybrid CTC/Attention Architecture for End-to-End Speech Recognition // IEEE Journal of Selected Topics in Signal Processing. – 2017. – Vol. 11, № 8. – P. 1240–1253.

107 Kingma D. P., Ba J. Adam: A method for stochastic optimization // ArXiv. – 2014. – Vol. abs.1412.6980. – P. 1-15.

108 Mamyrbayev O., Turdalyuly M., Mekebayev N., Mukhsina K., Keylan A., BabaAli B., Nabieva G., Duisenbayeva A., Akhmetov B. Continuous Speech Recognition of Kazakh Language // AMCSE 2018 - International Conference on Applied Mathematics, Computational Science and Systems Engineering. – Rome, Italy, 2018. – Vol. 24. – P. 1-5.

109 Mamyrbayev O., Oralbekova D., Kydyrbekova A., Turdalykyzy T., Bekarystankyzy A. End-to-End Model Based on RNN-T for Kazakh Speech Recognition // 3rd International Conference on Computer Communication and the Internet. – 2021. - P. 163-167.

110 Пат. №35886 Республика Казахстан. Система и способ распознавания агглютинативной слитной речи на основе интегрального (end-to-end) подхода /

Мамырбаев О.Ж.; заявитель и патентообладатель ИИВТ. – № 35886; опубл. 07.10.2022. – 2 с.

111 Mamyrbayev, O., Kydyrbekova, A., Alimhan, K., Oralbekova, D., Zhumazhanov, B., Nuranbayeva, B. Development of security systems using DNN and i & x-vector classifiers // Eastern-European Journal of Enterprise Technologies. – 2021. – Vol. 4 (9 (112)). – P. 32–45.

ПРИЛОЖЕНИЕ А

Фрагмент текста программы для обучения системы

Интегральное обучение сети

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import time

import tensorflow as tf
import scipy.io.wavfile as wav
import numpy as np
import utils as ut
from python_speech_features import mfcc
import os
tf.compat.v1.reset_default_graph()
tf.compat.v1.disable_eager_execution()

PATH = "E:/lstm_ctc-1/"
MOD_DIR = PATH+"save/" #save model
LOG_DIR = PATH+"log/" #log

audio_data_dir = PATH+"train1/" #training audios
test_audio_data_dir = PATH+"test/"

# Constants
PAD_VALUE = 0
SPACE_TOKEN = '<space>'
SPACE_INDEX = 0

# configs
num_features = 13
num_classes = 55 + 1 + 1

# Hyper-parameters
num_epochs = 3000
num_hidden = 64
num_layers = 1
batch_size = 1
initial_learning_rate = 1e-2
momentum = 0.95
```

```
# Loading the data
audio_filenames, script_filenames = ut.get_audio_scripts_files(audio_data_dir)
scripts_content = ut.get_script_content(script_filenames);
training_targets = dict(zip(audio_filenames,scripts_content))
print(training_targets)
num_examples = len(training_targets)
num_batches_per_epoch = int(num_examples/batch_size)
print("num_examples = {} ".format(num_examples))
print("num_batches_per_epoch = {}".format(num_batches_per_epoch))

#kazakh_alphabetic_order
def KAO(ch, flag):
    alphabet = ["а","ә","б","в","г","ғ","д","е","ё","ж","з","и","й","к","қ","л","м","н","ң","о",
                "ө","п","р","с","т","у","ұ","ү","ф","х","һ","ц","ч","ш","щ","ъ","ы","і","ь","э","ю","я",
                "0","1","2","3","4","5","6","7","8","9",
                "–","x","v"]
    if flag == 1:
        out = alphabet.index(ch)+1 #index
    elif flag == 0:
        out = alphabet[ch-1] if ch!=0 else " " #character
    return out

# Pre-Processing
def pre_processing(dic_audio_script):
    features = []
    labels = []
    seq_lengths = []
    originals = []
    max_len = 0
    for audio_file,audio_scirpt in dic_audio_script.items():
        # Process audio wav file
        fs, audio = wav.read(audio_file)
        inputs = mfcc(audio, samplerate=fs)
        train_inputs = (inputs - np.mean(inputs))/np.std(inputs)
        train_seq_len = [train_inputs.shape[0]]

        if max_len < train_inputs.shape[0]:
            max_len = train_inputs.shape[0]

        # Process audio scirpts to unicodes
        text = ut.delete_punct(audio_scirpt)

        targets = text.replace(' ', '  ')
        targets = targets.split(' ')
```

```
            targets = np.hstack([SPACE_TOKEN if x == " else list(x) for x in
targets])# change " " to <space>, set each character as a member of the list

            unicode_targets = np.asarray([SPACE_INDEX if x == SPACE_TOKEN
else KAO(x,1) for x in targets])# set unicodes, subtract the FIRST_INDEX to start
encode from the value of one

            # Set these in their lists
            features.append(train_inputs)
            labels.append(unicode_targets)
            seq_lengths.extend(train_seq_len)
            originals.append(text)

        return features, np.array(labels), seq_lengths, originals

    features, label_array, seq_lengths, originals = pre_processing(training_targets)

    # next batch generator
    def next_batch_gen(batch_counter):
        start = batch_counter * batch_size
        end = (batch_counter+1) * batch_size

        batchfeature = features[start:end]
        batch_label = label_array[start:end]
        batch_seq_lengths = seq_lengths[start:end]
        batch_seq_originals = originals[start:end]
        max_len = max(batch_seq_lengths)

        padded_features = np.zeros([batch_size, max_len, num_features],
dtype=np.float32)
        # padding
        for i, feature in enumerate(batchfeature):
            pad_length = max_len - len(feature)
            padded_feature = feature.tolist() + [[PAD_VALUE] * num_features] *
pad_length
            padded_features[i] = np.array(padded_feature);

        return padded_features, batch_label, batch_seq_lengths, batch_seq_originals

    # Graph
    # data placeholder
    inputs = tf.compat.v1.placeholder(tf.float32, [None, None,
num_features])#batchnum, total_samples_length, num_features
```

```
        #targets = tf.sparse_placeholder(tf.int32,[None,None])#batchnum,
samples_scripts_length
        targets = tf.compat.v1.sparse_placeholder(tf.int32,[None,None])#batchnum,
samples_scripts_length

        seq_len = tf.compat.v1.placeholder(tf.int32, [None])#

        # LSTM cell
        #cell = tf.contrib.rnn.LSTMCell(num_hidden, state_is_tuple=True)
        def lstm_cell():
            #lstm = tf.contrib.rnn.LSTMCell(num_hidden, state_is_tuple=True)
            lstm = tf.compat.v1.nn.rnn_cell.LSTMCell(num_hidden, state_is_tuple=True)
            return lstm

        # Stacking rnn cells
        #stack = tf.contrib.rnn.MultiRNNCell([lstm_cell() for _ in range(num_layers)])
        stack = tf.compat.v1.nn.rnn_cell.MultiRNNCell([lstm_cell() for _ in
range(num_layers)])

        #outputs, _ = tf.nn.dynamic_rnn(stack, inputs, seq_len, dtype=tf.float32)
        outputs, _ = tf.compat.v1.nn.dynamic_rnn(stack, inputs, seq_len,
dtype=tf.float32)

        shape = tf.shape(inputs)
        batch_s, max_timesteps = shape[0], shape[1]
        outputs = tf.reshape(outputs, [-1, num_hidden])

        W = tf.Variable(tf.random.truncated_normal([num_hidden,
                                        num_classes],
                                        stddev=0.1))
        b = tf.Variable(tf.constant(0., shape=[num_classes]))

        logits = tf.matmul(outputs, W) + b
        logits = tf.reshape(logits, [batch_s, -1, num_classes])
        logits = tf.transpose(logits, (1, 0, 2))

        with tf.name_scope('cost'):
            loss = tf.nn.ctc_loss(targets, logits, seq_len)
            #loss = tf.nn.ctc_loss(logits)
            cost = tf.reduce_mean(loss)
        tf.summary.scalar('cost', cost)

        with tf.name_scope('train'):
```

```
        #optimizer =
tf.train.MomentumOptimizer(initial_learning_rate,momentum).minimize(cost)
        #optimizer = tf.train.AdamOptimizer(initial_learning_rate).minimize(cost)
        #optimizer = tf.train.RMSPropOptimizer(initial_learning_rate,
0.9).minimize(cost)
        original_optimizer =
tf.train.AdamOptimizer(learning_rate=initial_learning_rate)
        optimizer = tf.contrib.estimator.clip_gradients_by_norm(original_optimizer,
clip_norm=9.0)
        train_op = optimizer.minimize(cost)

    decoded, log_prob = tf.nn.ctc_greedy_decoder(logits, seq_len)

    with tf.name_scope('label_error_rate'):
        ler = tf.reduce_mean(tf.edit_distance(tf.cast(decoded[0], tf.int32),
                                              targets))
    tf.summary.scalar('label_error_rate', ler)

    merged = tf.summary.merge_all()
    saver = tf.train.Saver()

    # Session
    with tf.Session() as sess:
        tf.global_variables_initializer().run()
        summary_writer = tf.summary.FileWriter(LOG_DIR,
graph=tf.get_default_graph())

        for curr_epoch in range(num_epochs):
            train_cost = train_ler = 0
            start = time.time()

            for batch_counter in range(num_batches_per_epoch):
                batch_feature, batch_label, batch_seq_lengths, batch_originals =
next_batch_gen(batch_counter)
                sparse_labels = ut.sparse_tuple_from(batch_label)
                #print(batch_label)
                #print(str(curr_epoch)+" "+str(batch_counter))
                #print(batch_originals)
                #print(np.shape(batch_label))

                feed = {inputs: batch_feature,
                        targets: sparse_labels,
                        seq_len: batch_seq_lengths}
```

```
            #batch_cost, _, summary = sess.run([cost, optimizer, merged], feed)
            batch_cost, _, summary = sess.run([cost, train_op, merged], feed)


            train_cost += batch_cost*batch_size #batchcost
            train_ler += sess.run(ler, feed_dict=feed) * batch_size
            summary_writer.add_summary(summary,curr_epoch)
        train_cost /= num_examples
        train_ler /= num_examples

        if train_ler < 0.1:
            break
        if curr_epoch%10 ==0:
            saver.save(sess, MOD_DIR, global_step=curr_epoch)

        log = "Epoch {}/{}, train_cost = {:.3f}, train_ler = {:.3f}, time = {:.3f}"
        print(log.format(curr_epoch+1, num_epochs, train_cost, train_ler,
time.time() - start))
    print("Training processes is Done!")
```

Процесс обучения сети

{'E:/lstm_ctc-1/train1/1/5ed8a0aa76264.wav': 'жұмекен астына от салып еттің көбігін алып күйб еңдеп жүр', 'E:/lstm_ctc-1/train1/1/5ed8a0b5e9ce1.wav': 'мұның амалы ұстап беру деп алексей к үлді', 'E:/lstm_ctc-1/train1/1/5ed8a0c5026f0.wav': 'мұның амалы ұстап беру деп алексей күлді', ' E:/lstm_ctc-1/train1/1/5ed8a1a5304b7.wav': 'көкесінің қалауымен күрес секциясына жазылған', ' E:/lstm_ctc-1/train1/1/5ed8a1a54abb8.wav': 'енді міне олар біз көрмеген алыс сапарға аттанып к етті біле білсек бақыт дегеніміз байлық пен мансап емес', 'E:/lstm_ctc-1/train1/1/5ed8a1af23a62 .wav': 'оу мазақ қылғанда төрт аяқты мақұлықты өстіп те мазақ қыла ма екен а', 'E:/lstm_ctc-1/ train1/1/5ed8a1b634bed.wav': 'әнжаннан шам әшім заманбек сынды ұлдар тарайды', 'E:/lstm_ct c-1/train1/1/5ed8a1b808a48.wav': 'өзің махаббат мәселесінен жұрдай екенсің ғой білмей жүрсе м', 'E:/lstm_ctc-1/train1/1/5ed8a1b83deb9.wav': 'өз құдіретіне өзі сенген жерде маңдай тер білек күші ақыл ойымен құдіретіне құдірет қоса білген', 'E:/lstm_ctc-1/train1/1/5ed8a1c02545d.wav': ' қолдан келіп тұрғанда қонышымнан бассам айып па', 'E:/lstm_ctc-1/train1/1/5ed8a1c0f3ea2.wa v': 'өз құдіретіне өзі сенген жерде маңдай тер білек күші ақыл ойымен құдіретіне құдірет қоса білген', 'E:/lstm_ctc-1/train1/1/5ed8a1c403a9a.wav': 'иіссезбес жоқ мылқау бәрін ішіп қойыпты', 'E:/lstm_ctc-1/train1/1/5ed8a1c87d2b1.wav': 'көкейкесті романымның қазақшасы қырық мың да намен шықты', 'E:/lstm_ctc-1/train1/1/5ed8a1c882f62.wav': 'бұрынғы өткен қазақ ақын жыраула рының басым бөлігінде бұл тақырып міндетті түрде қозғалатын', 'E:/lstm_ctc-1/train1/1/5ed8a1 cc54ab9.wav': 'бір күні сырбаймен көрші тұратын айтжан байқожаев деген артист әлде бір себ еппен жұмысқа бармай қалады', 'E:/lstm_ctc-1/train1/1/5ed8a1d517e52.wav': 'интенсивті модер н ғылымның салаларға бөлініп дамуына жол ашқанымен қарапайым халықтың санасының то қырауына алып келді', 'E:/lstm_ctc-1/train1/1/5ed8a1dd17a88.wav': 'қамақтан шыққан қыран бір ден көкке самғамайды', 'E:/lstm_ctc-1/train1/1/5ed8a1e1a30f2.wav': 'кім көрінген жеке өміріңе қол сұғып көңіл күйіңді күйзелтіп арыңды саудаға салып көзқарасыңа қысым адамдық күш ж ігеріңе зорлық зомбылық жасай алады', 'E:/lstm_ctc-1/train1/1/5ed8a1e42b719.wav': 'осыны да

күйеу дейді ғой', 'E:/lstm_ctc-1/train1/1/5ed8a1e5a78dc.wav': 'елші құрчидің жауабын тізбектеп құсбегіге мәлімдеді'}
num_examples = 20
num_batches_per_epoch = 20
WARNING:tensorflow:Entity <bound method MultiRNNCell.call of <tensorflow.python.ops.rnn_cell_impl.MultiRNNCell object at 0x0000019F6BD52108>> could not be transformed and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bound method MultiRNNCell.call of <tensorflow.python.ops.rnn_cell_impl.MultiRNNCell object at 0x0000019F6BD52108>>: AttributeError: module 'gast' has no attribute 'Index'
WARNING: Entity <bound method MultiRNNCell.call of <tensorflow.python.ops.rnn_cell_impl.MultiRNNCell object at 0x0000019F6BD52108>> could not be transformed and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bound method MultiRNNCell.call of <tensorflow.python.ops.rnn_cell_impl.MultiRNNCell object at 0x0000019F6BD52108>>: AttributeError: module 'gast' has no attribute 'Index'
WARNING:tensorflow:Entity <bound method LSTMCell.call of <tensorflow.python.ops.rnn_cell_impl.LSTMCell object at 0x0000019F612EE1C8>> could not be transformed and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bound method LSTMCell.call of <tensorflow.python.ops.rnn_cell_impl.LSTMCell object at 0x0000019F612EE1C8>>: AttributeError: module 'gast' has no attribute 'Index'
C:\Users\Dina\anaconda3\envs\tf\lib\site-packages\ipykernel_launcher.py:94: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray
WARNING: Entity <bound method LSTMCell.call of <tensorflow.python.ops.rnn_cell_impl.LSTMCell object at 0x0000019F612EE1C8>> could not be transformed and will be executed as-is. Please report this to the AutgoGraph team. When filing the bug, set the verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full output. Cause: converting <bound method LSTMCell.call of <tensorflow.python.ops.rnn_cell_impl.LSTMCell object at 0x0000019F612EE1C8>>: AttributeError: module 'gast' has no attribute 'Index'
Epoch 1/3000, train_cost = 822.948, train_ler = 0.967, time = 4.353
Epoch 2/3000, train_cost = 631.519, train_ler = 0.968, time = 3.671
Epoch 3/3000, train_cost = 291.615, train_ler = 0.992, time = 3.632
Epoch 4/3000, train_cost = 226.145, train_ler = 0.999, time = 3.566
Epoch 5/3000, train_cost = 222.027, train_ler = 0.999, time = 3.604
Epoch 6/3000, train_cost = 220.234, train_ler = 0.999, time = 3.666
Epoch 7/3000, train_cost = 218.114, train_ler = 1.000, time = 3.581
Epoch 8/3000, train_cost = 216.000, train_ler = 1.000, time = 3.664
Epoch 9/3000, train_cost = 213.237, train_ler = 1.000, time = 3.682
Epoch 10/3000, train_cost = 211.642, train_ler = 0.999, time = 3.671
Epoch 11/3000, train_cost = 210.655, train_ler = 0.998, time = 4.042
Epoch 12/3000, train_cost = 207.825, train_ler = 1.000, time = 3.638
Epoch 13/3000, train_cost = 205.480, train_ler = 0.996, time = 3.658
Epoch 14/3000, train_cost = 201.686, train_ler = 0.996, time = 3.697
Epoch 15/3000, train_cost = 198.342, train_ler = 0.995, time = 3.701
Epoch 16/3000, train_cost = 194.004, train_ler = 0.992, time = 3.756
Epoch 17/3000, train_cost = 192.727, train_ler = 0.989, time = 3.705

Epoch 18/3000, train_cost = 193.089, train_ler = 0.986, time = 3.708
Epoch 19/3000, train_cost = 199.638, train_ler = 0.983, time = 3.600
Epoch 20/3000, train_cost = 183.035, train_ler = 0.980, time = 3.647
Epoch 21/3000, train_cost = 178.188, train_ler = 0.980, time = 4.068
Epoch 22/3000, train_cost = 176.081, train_ler = 0.979, time = 3.738
Epoch 23/3000, train_cost = 171.418, train_ler = 0.975, time = 3.684
Epoch 24/3000, train_cost = 167.628, train_ler = 0.971, time = 3.634
Epoch 25/3000, train_cost = 164.819, train_ler = 0.971, time = 3.703
Epoch 26/3000, train_cost = 160.324, train_ler = 0.963, time = 3.670
Epoch 27/3000, train_cost = 158.799, train_ler = 0.956, time = 3.628
Epoch 28/3000, train_cost = 156.451, train_ler = 0.956, time = 3.662
Epoch 29/3000, train_cost = 154.006, train_ler = 0.952, time = 3.656
Epoch 30/3000, train_cost = 152.035, train_ler = 0.946, time = 3.667
Epoch 31/3000, train_cost = 148.432, train_ler = 0.949, time = 4.322
Epoch 32/3000, train_cost = 146.372, train_ler = 0.935, time = 3.637
Epoch 33/3000, train_cost = 142.492, train_ler = 0.927, time = 3.677
Epoch 34/3000, train_cost = 141.198, train_ler = 0.914, time = 3.676
Epoch 35/3000, train_cost = 139.766, train_ler = 0.914, time = 3.676
Epoch 36/3000, train_cost = 134.409, train_ler = 0.898, time = 3.670
Epoch 37/3000, train_cost = 131.414, train_ler = 0.864, time = 3.669
Epoch 38/3000, train_cost = 129.496, train_ler = 0.830, time = 3.709
Epoch 39/3000, train_cost = 127.051, train_ler = 0.832, time = 3.722
Epoch 40/3000, train_cost = 123.667, train_ler = 0.781, time = 3.690
Epoch 41/3000, train_cost = 118.855, train_ler = 0.774, time = 4.027
Epoch 42/3000, train_cost = 117.974, train_ler = 0.759, time = 3.661
Epoch 43/3000, train_cost = 113.192, train_ler = 0.725, time = 3.624
Epoch 44/3000, train_cost = 111.097, train_ler = 0.691, time = 3.665
Epoch 45/3000, train_cost = 108.333, train_ler = 0.682, time = 3.703
Epoch 46/3000, train_cost = 104.821, train_ler = 0.640, time = 3.727
Epoch 47/3000, train_cost = 101.977, train_ler = 0.630, time = 3.699
Epoch 48/3000, train_cost = 99.499, train_ler = 0.620, time = 3.706
Epoch 49/3000, train_cost = 99.535, train_ler = 0.571, time = 3.732
Epoch 50/3000, train_cost = 93.987, train_ler = 0.549, time = 3.638
WARNING:tensorflow:From C:\Users\Dina\anaconda3\envs\tf\lib\site-packages\tensorflow\python\training\saver.py:960: remove_checkpoint (from tensorflow.python.training.checkpoint_management) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to delete files with this prefix.
Epoch 51/3000, train_cost = 95.074, train_ler = 0.545, time = 4.189
Epoch 52/3000, train_cost = 90.157, train_ler = 0.512, time = 3.818
Epoch 53/3000, train_cost = 87.776, train_ler = 0.493, time = 3.682
Epoch 54/3000, train_cost = 85.042, train_ler = 0.468, time = 3.693
Epoch 55/3000, train_cost = 81.015, train_ler = 0.451, time = 3.693
Epoch 56/3000, train_cost = 79.735, train_ler = 0.436, time = 3.718
Epoch 57/3000, train_cost = 78.884, train_ler = 0.419, time = 3.727
Epoch 58/3000, train_cost = 76.507, train_ler = 0.390, time = 3.687
Epoch 59/3000, train_cost = 76.651, train_ler = 0.371, time = 3.849
Epoch 60/3000, train_cost = 72.654, train_ler = 0.353, time = 3.803
Epoch 61/3000, train_cost = 73.564, train_ler = 0.355, time = 4.180

Epoch 62/3000, train_cost = 74.534, train_ler = 0.350, time = 3.800
Epoch 63/3000, train_cost = 72.689, train_ler = 0.329, time = 3.715
Epoch 64/3000, train_cost = 66.422, train_ler = 0.299, time = 3.736
Epoch 65/3000, train_cost = 67.987, train_ler = 0.335, time = 3.844
Epoch 66/3000, train_cost = 68.119, train_ler = 0.298, time = 3.841
Epoch 67/3000, train_cost = 66.209, train_ler = 0.299, time = 3.816
Epoch 68/3000, train_cost = 59.068, train_ler = 0.271, time = 3.723
Epoch 69/3000, train_cost = 57.937, train_ler = 0.245, time = 3.719
Epoch 70/3000, train_cost = 55.497, train_ler = 0.248, time = 3.800
Epoch 71/3000, train_cost = 53.631, train_ler = 0.246, time = 4.159
Epoch 72/3000, train_cost = 52.401, train_ler = 0.240, time = 3.777
Epoch 73/3000, train_cost = 52.621, train_ler = 0.230, time = 3.820
Epoch 74/3000, train_cost = 53.435, train_ler = 0.230, time = 3.818
Epoch 75/3000, train_cost = 54.175, train_ler = 0.219, time = 3.759
Epoch 76/3000, train_cost = 51.804, train_ler = 0.211, time = 3.781
Epoch 77/3000, train_cost = 49.928, train_ler = 0.213, time = 3.791
Epoch 78/3000, train_cost = 53.770, train_ler = 0.199, time = 3.787
Epoch 79/3000, train_cost = 48.561, train_ler = 0.192, time = 3.755
Epoch 80/3000, train_cost = 45.307, train_ler = 0.188, time = 3.779
Epoch 81/3000, train_cost = 44.321, train_ler = 0.187, time = 4.173
Epoch 82/3000, train_cost = 47.456, train_ler = 0.186, time = 3.728
Epoch 83/3000, train_cost = 46.411, train_ler = 0.176, time = 3.791
Epoch 84/3000, train_cost = 44.186, train_ler = 0.176, time = 3.769
Epoch 85/3000, train_cost = 39.883, train_ler = 0.163, time = 3.665
Epoch 86/3000, train_cost = 39.386, train_ler = 0.157, time = 3.685
Epoch 87/3000, train_cost = 38.994, train_ler = 0.159, time = 3.664
Epoch 88/3000, train_cost = 36.401, train_ler = 0.137, time = 3.744
Epoch 89/3000, train_cost = 36.969, train_ler = 0.140, time = 3.781
Epoch 90/3000, train_cost = 33.587, train_ler = 0.137, time = 3.774
Epoch 91/3000, train_cost = 32.838, train_ler = 0.122, time = 4.243
Epoch 92/3000, train_cost = 33.878, train_ler = 0.125, time = 3.753
Training processes is Done!

ПРИЛОЖЕНИЕ Б

Фрагмент текста программы для тестирования системы

```
session  = tf.compat.v1.Session()
merged = tf.summary.merge_all()
saver = tf.train.Saver()

# Session
with session:
   ckpt = tf.train.get_checkpoint_state(MOD_DIR)
   if ckpt and ckpt.model_checkpoint_path:
      saver.restore(session, ckpt.model_checkpoint_path)
   test_ler = 0
   for i, f in enumerate(features):
      feed = {inputs: [f],
            seq_len: [len(f)]}

      result = session.run(decoded[0], feed_dict=feed)
      str_decoded = ''.join([KAO(x,0) for x in np.asarray(result[1])])
      str_decoded = str_decoded.replace('0', ' ')
      print("audio file: " + audio_filenames[i])
      print('Decoded:\n%s' % str_decoded+"\n")
```

Результаты полученных данных

audio file: E:/lstm_ctc-1/train1/1/5ed8a0aa76264.wav
Decoded:
жұмеке астына салып етің көбігін алып күйбеңдеп жүр

audio file: E:/lstm_ctc-1/train1/1/5ed8a0b5e9ce1.wav
Decoded:
мұның амалы ұста беру деп алексей күлді

audio file: E:/lstm_ctc-1/train1/1/5ed8a1a5304b7.wav
Decoded:
көкесінің қалауымен күрес секциясына жазылған

audio file: E:/lstm_ctc-1/train1/1/5ed8a1a54abb8.wav
Decoded:
енді міне олар бір көрмеген лыс сапарға аттанып кетті біле білсек қыт дегеніміз байлық пен масап емес

audio file: E:/lstm_ctc-1/train1/1/5ed8a1af23a62.wav

Decoded:
оу мазақ қылған таяқты мақұлықты өстіп те маза қыла ма екен а

audio file: E:/lstm_ctc-1/train1/1/5ed8a1b634bed.wav
Decoded:
әнжа нан шам әшім заманбек ынды ұлдар тарайды

audio file: E:/lstm_ctc-1/train1/1/5ed8a1b808a48.wav
Decoded:
өзің махабат мәселесінен жұрдай екенсің ғой білмей жүрсем

audio file: E:/lstm_ctc-1/train1/1/5ed8a1b83deb9.wav
Decoded:
өз құдіретіне өзі сенген жерде маңдай тер білек күші ақыл ойымен құдіретіне құдірет қоса білген

audio file: E:/lstm_ctc-1/train1/1/5ed8a1c02545d.wav
Decoded:
қолдан келі ұғынды қонышын байсап

audio file: E:/lstm_ctc-1/train1/1/5ed8a1c403a9a.wav
Decoded:
Иісе бес жоқ мылқау бәрін ішіп қойыпты

audio file: E:/lstm_ctc-1/train1/1/5ed8a1c87d2b1.wav
Decoded:
көкейкес романымның қазақшасы қырық мың данамен шық

audio file: E:/lstm_ctc-1/train1/1/5ed8a1c882f62.wav
Decoded:
бұрын өткен қазақ ақын жырауларының басым бөлігінде бұл тақырып міндетті түрде қозғалады

audio file: E:/lstm_ctc-1/train1/1/5ed8a1cc54ab9.wav
Decoded:
бірі сырмен көрші ұртын дегенде әлде бір ебеп пе жұмысқа бармай қалды

audio file: E:/lstm_ctc-1/train1/1/5ed8a1d517e52.wav
Decoded:
интенсиві модерн ғылымның саларға бөлініп дамуына жол ашқанымен қарапайым халықтың санасының оқырауындап келді

audio file: E:/lstm_ctc-1/train1/1/5ed8a1dd17a88.wav
Decoded:

қамақтан шыққан қыран бірден көке сағайды

audio file: E:/lstm_ctc-1/train1/1/5ed8a1e42b719.wav
Decoded:
осыны да күйеу дейді ғой

audio file: E:/lstm_ctc-1/train1/1/5ed8a1e5a78dc.wav
Decoded:
елші курчидің жауабын тізбгенде құсбелгіге мәлімдеді

ПРИЛОЖЕНИЕ В

Свидетельства государственной регистрации прав на объект авторского права

1) Авторское свидетельство "Система автоматического распознавания казахской речи на основе интегральной архитектуры" № 15501 2021 25 февраль.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК

2021 жылғы «25» ақпан № 15501

Автордың (лардың) жөні, аты, әкесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе):
МАМЫРБАЕВ ОРКЕН ЖУМАЖАНОВИЧ, ОРАЛБЕКОВА ЛИНА ОРЫМБАЕВНА, КЫЛЫРБЕКОВА АЙЗАТ СИЯЗБЕКОВНА, ЖҰМАЖАНОВ БАҒАШАР ЖҰМАЖАНҰЛЫ, ТҰРДАЛЫҚЫЗЫ ТОЛҒАНАЙ

Авторлық құқық объектісі: ЭЕМ-ге арналған бағдарлама

Объектінің атауы: Система автоматического распознавания казахской речи на основе интегральной архитектуры

Объектіні жасаған күні: 18.11.2020

Құжат түпнұсқасының http://www.kazpatent.kz/ru сайтының "Авторлық құқық" бөлімінде тексеруге болады https://copyright.kazpatent.kz

Подлинность документа возможно проверить на сайте kazpatent.kz в разделе «Авторское право» https://copyright.kazpatent.kz

ЭЦҚ қол қойылды Оспанов Е.К.

2) Авторское свидетельство "Система идентификации и аутентификации через речевые технологии" № 23323 от 4 февраля 2022.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК

2022 жылғы «4» ақпан № 23323

[illegible]:
ОРАЛБЕКОВА ДИНА ОРЫМБАЕВНА, [illegible]

Авторлық құқық объектісі: ЭЕМ-ге арналған бағдарлама

Объектінің атауы: Система идентификации и аутентификации через речевые технологии

Объектінің жасалған күні: 02.02.2022

[illegible]

ЭЦҚ қол қойылды

Е. Куантыров

3) Авторское свидетельство "Система автоматического распознавания казахской слитной речи на основе модели с механизмом внимания" № 24178 от 5 марта 2022.

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК

2022 жылғы «5» наурыз № 24178

Автордың (лардың) жеке, аты, әкесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе):
МАМЫРБАЕВ ОРКЕН ЖУМАЖАНОВИЧ, ОРАЛБЕКОВА ЛИНА ОРЫМБАЕВНА, ӘЛІМХАН КИЛАН, КЫЛЫРБЕКОВА АЙЗАТ СИЯЗБЕКОВНА, ЖҰМАЖАНОВ БАҒАШАР ЖҰМАЖАНҰЛЫ, ТҰРДАЛЫҚЫЗЫ ТОЛҒАНАЙ

Авторлық құқық объектісі: ЭЕМ-ге арналған бағдарлама

Объектінің атауы: Система автоматического распознавания казахской слитной речи на основе модели с механизмом внимания

Объектіні жасаған күні: 01.03.2022

Құжат тексерілуін http://www.kazpatent.kz/ru сайтының "Авторлық құқық" бөлімінде тексеруге болады https://copyright.kazpatent.kz

Подлинность документа возможно проверить на сайте kazpatent.kz в разделе «Авторское право» https://copyright.kazpatent.kz

ЭЦҚ қол қойылды

А.Естаев

ПРИЛОЖЕНИЕ Г

Акт внедрения результатов диссертационного исследования

Speech Processing Solutions GmbH
Gutheil-Schoder-Gasse 8–12
1100 Vienna, Austria
Tel.: +43 1 60529
Fax: +43 1 60529

SPEECH PROCESSING SOLUTIONS

Implementation Act

On the implementation of the results of the project AP08855743 "Development of an end-to-end automatic speech recognition system for agglutinative languages" of grant funding of the Ministry of Education and Science of the Republic of Kazakhstan for 2020-2022.

The expert commission of the company «Philips» consists of:

Commission Chairman: Michael Wois

Members of the Commission: Jacob CJ, information systems specialist

drew up this act stating that the results of the project "Development of an end-to-end automatic speech recognition system for agglutinative languages", headed by Alimhan Keylan, Ph.D., Professor, chief research scientist, and executors Mamyrbayev O.Zh., Ph.D., Oralbekova D.O., junior research scientist, and other obtained during the performance of work at the Institute of Information and Computational Technologies" CS MES RK, state registration number 0120RK00344, grant funding of the MES RK for 2020-2022, were accepted for implementation in the Philips company.

The full results of this project are presented in the reports for 2020-2021.

As part of these works, the following works of the project are of interest to our company: the speech corpus of the Kazakh language and language models for recognition of Kazakh speech and the system of automatic recognition of Kazakh continuous speech. These products are necessary to automate some functions of services and work of state and commercial organizations in the Republic of Kazakhstan.

The Commission believes that the inclusion of a special system of automatic recognition of Kazakh speech developed within the framework of the project into the software product will contribute to the promotion of products in companies both in the Republic of Kazakhstan and the implementation of international interaction of the company in the global information space.

The development of this area within our company is of interest for solving the problems of effective recognition of Kazakh speech, which is necessary in the conditions of application of the company's products in the Republic of Kazakhstan.

Commission Chairman Michael Wois

Commission members Jacob CJ

Singature:

Yours faithfully,

Speech Processing Solutions GmbH

BY: Michael Wois

TITLE: Director – Business Dev

Date: 24/Marh/2022

ПРИЛОЖЕНИЕ Д

Патент на изобретение

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ РЕСПУБЛИКА КАЗАХСТАН

REPUBLIC OF KAZAKHSTAN

ПАТЕНТ
PATENT

№ 35886

ӨНЕРТАБЫСҚА / НА ИЗОБРЕТЕНИЕ / FOR INVENTION

(21) 2021/0312.1

(22) 21.05.2021

(45) 07.10.2022

(54) Агглютинативті үздіксіз сөйлеуді интегралды (end-to-end) тану тәсілі және оны жүзеге асыратын жүйе
Система и способ распознавания агглютинативной слитной речи на основе интегрального (end-to-end) подхода
System and method for recognition of agglutinative continuous speech based on end-to-end approach

(73) Мамырбаев Оркен Жумажанович (KZ)
Mamyrbayev Orken Zhumazhanovich (KZ)

(72) Мамырбаев Оркен Жумажанович (KZ) Mamyrbayev Orken Zhumazhanovich (KZ)
Оралбекова Дина Орымбаевна (KZ) Oralbekova Dina Orymbaevna (KZ)
Кыдырбекова Айзат Сиязбековна (KZ) Kydyrbekova Aizat Siyazbekovna (KZ)
Жұмажанов Бағашар Жұмажанұлы (KZ) Zhumazhanov Bagashar Zhumazhanuly (KZ)
Тұрдалықызы Толғанай (KZ) Turdalykyzy Tolganay (KZ)

ЭЦҚ қол қойылды Е. Оспанов
Подписано ЭЦП Е. Оспанов
Signed with EDS Y. Ospanov

«Ұлттық зияткерлік меншік институты» РМК директоры
Директор РГП «Национальный институт интеллектуальной собственности»
Director of the «National Institute of Intellectual Property» RSE