СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

StT – Speech to Text, перевод речи в текстовый формат.

PHC – рекуррентная нейронная сеть.

LSTM – long short-term memory, длинная краткосрочная память.

seq2seq – sequence to sequence, модель формата “последовательность в последовательность”

WER – word error rate, процент ошибок в словах.

ELMo – Embeddings from Language Models.

BERT – Bidirectional Encoder Representations from Transformers.

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Датасет (англ. Dataset) – набор данных, использующийся для обучения модели.

Нейронная сеть – это метод в искусственном интеллекте, который учит компьютеры обрабатывать данные таким же способом, как и человеческий мозг. Это тип процесса машинного обучения, называемый глубоким обучением модели, который использует взаимосвязанные узлы или нейроны в слоистой структуре, напоминающей человеческий мозг.

Рекуррентная нейронная сеть – нейронная сеть, образующая направленную последовательность.

Нейрон – единичный узел нейронной сети имеющий свой вес.

Трансформер (англ. Transformer) – вид архитектуры глубокой нейронной сети, основанный на механизме внимания без использования рекуррентных нейронных сетей.

Энкодер (англ. Encoder) – часть архитектуры модели, которая преобразует входные данные в скрытое представление или кодированное представление.

Декодер (англ. Decoder) – часть архитектуры модели, которая преобразует скрытое представление, полученное из кодировщика, обратно в исходный формат или желаемый выходной формат.

ВВЕДЕНИЕ

Анализ и обработка аудиоданных – одна из крупнейших сфер в мире информационных технологий. Сегодня проводится множество исследований и разработок, которые могут улучшить качество коммуникации между людьми. Одно из перспективных направлений этой области – распознавание эмоций в речи на основе аудиозаписей. Анализ голоса человека позволяет понимать эмоциональный контекст разговора, оценивать окраску речи собеседника, контролировать себя. Также, разработки в сфере определения тональности речи являются крайне важными для людей с проблемами слуха. Подобного рода ограничения восприятия заставляют многих упускать важные детали в речи собеседника. Таким образом, определение эмоционального окраса речи является крайне полезной технологией для множества пользователей из разных слоев населения.

Так как мессенджеры набирают все большую популярность, пользователи интернета стали чаще пользоваться функцией аудиосообщений. Данный формат удобен и прост в обращении, однако уследить за собой при его использовании становится не так просто, зачастую люди случайно могут высказать нечто агрессивное, не осознавая это. Так же возможны недопонимания, когда один из собеседников не смог распознать шутку, или наоборот серьезность чужих слов. Все эти факторы негативно сказываются на взаимопонимании людей.

В качестве одного из путей решения этой проблемы, возможно использовать алгоритмы машинного обучения. Для точного определения эмоциональной окраски необходимо конвертировать исходный аудиофайл в текстовый формат, и затем проанализировать содержавшуюся речь на эмоции. Соответственно, должны использоваться две модели, обученные на различных датасетах. В связи с этим, корректный выбор подходящих моделей, данных для обучения и параметров обучения позволит получить наиболее точный результат. В качестве интерфейса взаимодействия будет использоваться чат-

бот в приложении Telegram, т.к. данный мессенджер является одним из самых популярных в РФ.

Цель данной работы – автоматизация процесса распознавания эмоций по содержанию голосовых сообщениях для мессенджера Telegram

Для достижения обозначенной цели, необходимо выполнить следующие задачи:

1) сравнить эффективность использования моделей для решения задач перевода голоса в текст и анализа текста на эмоции. Рассмотреть разницу между рекуррентными нейронными сетями (Deep Speech, ELMo) и трансформерными нейронными сетями (Whisper, BERT);
2) сформировать датасет, содержащий в себе короткие, эмоционально окрашенные тексты, промаркированные соответствующими эмоциями;
3) спроектировать детали технической реализации бота;
4) выполнить дообучение модели на основе сформированного датасета, подобрать подходящие параметры для корректного распознавания эмоций в транскрибациях голосовых сообщений;
5) реализовать обработку аудиодорожек моделью распознавания текста, и подготовку полученного результата для последующей работы дообученной модели;
6) разработать бота Telegram, передающего на обработку аудиодорожки голосовых сообщений, и возвращающего результат работы модели;
7) произвести тестирование и оценку точности работы бота;
8) публикация в открытых репозиториях

1 ТЕОРЕТИЧЕСКИЙ ОБЗОР

В данной главе проведен анализ предметной области распознавания эмоций в речи на основе аудиозаписей. Рассмотрена задача и подходы к ее решению, произведен выбор моделей для разработки бота и произведен обзор их аналогов.

1.1Подходы к распознаванию эмоций

Человек производит распознавание эмоций в речи собеседника не только по значению сказанного, но и по выражению лица, жестам и по большому количеству других параметров. Однако, когда дело касается распознавания эмоций программой, разработать модель учитывающую абсолютно все факторы невозможно, поэтому есть множество различных направлений, сконцентрированных на конкретных проявлениях эмоций, например по лицу или голосу [1].

В данной работе рассмотрено определение эмоций речи в аудиозаписи, что позволяет нам выявить два принципиально различных способа определения эмоций: по смысловой нагрузке сказанного и по параметрам голоса: скорости, громкости, тембру. Оба подхода имеют свои преимущества и недостатки, поэтому рассмотрим каждый из них подробнее.

1.1.1 Определение эмоций по смысловой нагрузке

Для распознавания эмоций в аудиозаписи по смысловой нагрузке необходимо перевести озвученную речь в текстовый формат. Данная задача лежит в разделе компьютерной лингвистики под названием распознавание речи – крайне перспективном и быстро развивающемся направлении. Так как разработки в данной области ведутся с 50-х годов 20 века, системы перевода речи в текст (Speech to Text) развивались вплоть до сегодняшнего дня, когда самым успешным решением для проблемы стали нейронные сети.

Несмотря на то, что первые результаты применения машинного обучения в 90-х не имели существенного роста эффективности по сравнению со старыми способами распознавания, появившиеся глубокие, а затем и рекуррентные, нейронные сети значительно повысили качество результата

обработки звука [2]. Соответственно, единственно подходящим методом для решения задачи перевода речи в текст будет являться использование обученной модели на основе машинного обучения.

Для распознавания эмоций в полученном тексте необходимо использовать вторую модель, которая уже распознает эмоции, заложенные в смысловую нагрузку слов. Распознаваний эмоций в тексте также является разделом компьютерной лингвистики, возникшим уже позже, в 2000-х годах нашего века [3]. Несмотря на относительно недавнее появление, нынешние результаты точности определения эмоций моделями показывают успешное и быстрое развитие этой области.

Однако, несмотря на высокие значения точности определения эмоций, рассмотрим сложности и другие лексические факторы, влияющие на корректность итоговых результатов работы модели, которые будут характерные данному подходу решения задачи [4]:

1) многозначность лексики – слова зачастую могут иметь несколько разных смыслов, что будет мешать модели корректно распознавать эмоциональную окраску необходимого слова;
2) контекстозависимость – оценочные слова могут использоваться не только в прямом смысле, но и являться частью фразеологизма или устойчивого выражения;
3) нереальный контекст – при использовании слов в сослагательном наклонении, или при описывании воображаемого события определение соответствующей окраски словам становится затруднительным;
4) ирония – при использовании слов в саркастическом тоне, выявить истинную эмоциональную окраску является очень сложной задачей.

Рекуррентные нейронные сети позволяют эффективно справляться с большей частью проблем, связанных с контекстом, однако даже они с трудом справляются с определением сарказма. Также стоит рассмотреть количество подходящих датасетов, позволяющих обучить модель корректному

распознаванию. В открытом доступе находится огромное количество различных данных для обучения модели. Они могут являться наборами новостей, постов из социальных сетей, переписок, рецензий на фильм, отзывов на места и многим другим [5]. Всё содержимое датасетов написано реальными людьми, что положительно скажется на точности определения эмоций пользователя.

1.1.2 Определение эмоций по параметрам голоса

Не только смысловая нагрузка позволяет человеку уловить эмоции своего собеседника, бывают даже случаи, когда люди делают выводы об эмоциональной окраске речи неосознанно, считая агрессивными или грубыми слова, сказанные без какого-либо намерения проявить агрессию.

Для понимания тональности речи, мы обращаем внимание на тембр, скорость и множество других факторов. Модели, которые определяют эмоции по параметрам голоса, работают по схожему принципу – они вычленяют определенные признаки звуковой волны, например форманты или интенсивности, и на основе их совокупности делают вывод о типе эмоции [6].

Нейронные сети, построенные на основе этого метода, могут быть довольно точными и умело определять интонации голосов. Однако, есть причины, по которым определение тональности речи может быть значительно усложнено [7]:

1) различие языков – одна из главных сложностей данного подхода, это различие системы оценивания для разных языков. Некоторые языки звучат грубее, другие мягче, речь может быть быстрее или медленнее;
2) различие культур – разница в культуре у людей разных национальностей может быть разительной и сильно влиять на их речь;
3) индивидуальность – индивидуальные особенности речи каждого человека так же могут разительно отличаться. Громкость, тембр, скорость – все это может очень сильно отличаться от человека к человеку и влиять на точность распознавания эмоций, особенные

проблемы могут вызвать люди, которые обладают более спокойным темпераментом;

4) смешанность эмоций – звучание тех или иных эмоций могут смешиваться, мешая распознаванию.

Учесть все вышеперечисленные факторы – практически невозможная задача, так как они могут варьироваться от человека к человеку с очень большим разбросом.

1.1.3 Выбор подхода решения задачи

Рассматривая каждый из описанных подходов, выбор кажется крайне неоднозначным, так как у каждого из них есть свои преимущества. Модели, основанные на анализе текста, будут лучше справляться в случаях менее экспрессивных людей, в то время как модели, которые анализируют признаки звуковых волн, будут учитывать интонации речи.

В среднем, результаты точности обоих подходов отличаются незначительно, в районе 5% [8], что так же не позволяет нам сделать конкретных выводов о том, какой вид модели выбрать. Однако, рассматривая доступность и качество датасетов, необходимых для обучения моделей, можно прийти к выводу, что количество и качество данных значительно выше в случае обучения модели на основе текста. Основная часть датасетов, необходимых для обучения нейронных сетей, анализирующих параметры голоса, является составленными искусственно [9], для предоставления большей вариативности эмоций, что негативно скажется на точности обучения. Также, сильно повлияет на точность определения эмоций то, что почти все данные для датасетов состоят из разных языков, эмоциональный окрас интонаций которых может не совпадать с эмоциональным окрасом русского языка, на котором планируется производить тестирование.

Таким образом, оптимальным выбором для решения задачи этой работы, является подход, основанный на анализе эмоций смысловой нагрузки сказанного.

1.2Обзор существующих моделей

Так как выбранный подход представляет собой взаимодействие двух моделей, нам необходимо найти наиболее подходящие варианты для обоих видов нейронных сетей. Точность определения эмоций будет напрямую зависеть от точности перевода речи в текст, следовательно в первую очередь рассмотрим, как он происходит. Основной принцип работы моделей StT представляет собой следующую последовательность действий:

1) подача на вход аудиофайлов с речью;
2) разделение аудиофайлов на фрагменты и преобразование их в спектрограмму;
3) обучение на основе полученных спектрограмм с расшифровкой;
4) проведение предсказаний на основе полученных данных.

Рассмотрим подробнее наиболее популярные и точные модели Whisper и DeepSpeech.

1.2.1 Модель DeepSpeech

Данная модель разработана компанией Mozilla, основной задачей модели является обработка аудиофайлов – проведение преобразования речи в текст. Архитектура модели DeepSpeech базируется на принципе рекуррентной нейронной сети (РНС) [10]. Так как процесс распознавания текста напрямую зависит от контекста, традиционные нейронные сети для данных задач не подойдут.

В отличии от обычных глубоких нейронных сетей, где существует только один вход, РНС работают по другому принципу. Такая сеть имеет два входа – настоящее и недавнее прошлое [11]. Это позволяет эффективно подстраивать РНС под различного рода изменения и дает возможность произведения прогнозов.

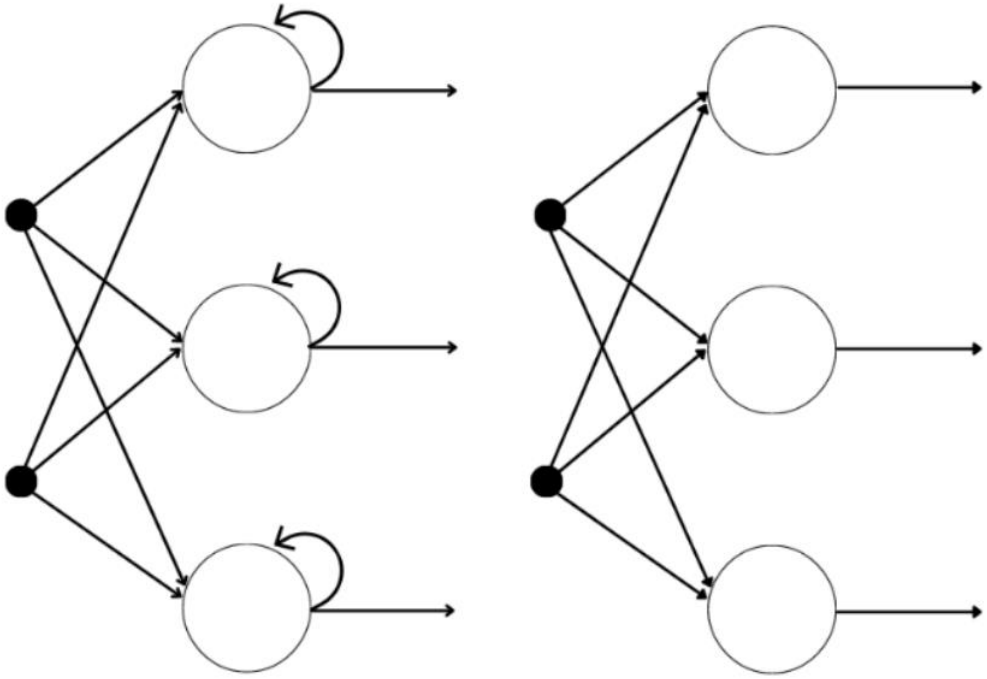

Рисунок 1 – Пример разницы в информационном потоке между рекуррентной нейронной сетью и сетью с прямой связью [11]

DeepSpeech в свою очередь является подвидом РНС – сетью long short-term memory (LSTM) [10]. Главное отличие такой сети от обычной РНС – возможность запоминать информацию на гораздо большей дистанции. Происходит это благодаря взаимодействию с состоянием ячейки – информации, передающейся по всей цепочке модулей РНС, по мере обучения различные фильтры регулируют состояние ячейки, добавляя или удаляя информацию, давая возможность получать из информацию из ранних шагов.

1.2.2 Модель Whisper

Модель Whisper была разработана компанией OpenAI. В отличии от DeepSpeech, она использует архитектуру Transformer, которая отлично работает с последовательностями данных. Таким образом, Whisper представляет собой модель sequence-to-sequence(seq2seq). Она принимает в себя последовательность данных, в частности аудиодорожку, и возвращает последовательность данных в виде текста.

Основная разница трансформерных сетей от РНС это отсутствие каких-либо рекуррентных связей. Вместо запоминания результатов предыдущих итераций, модель использует механизм внимания, моделирующий

зависимости между различными частями последовательности. Механизмов внимания может быть несколько. В случае Whisper, в энкодере, компоненте отвечающем за обработку входящей в сеть последовательности, находится механизм self attention, а в декодере, компоненте, генерирующем выходную последовательность, находятся механизмы self attention и cross attention[12]. Трансформерная модель архитектуры также позволяет производить параллельную обработку последовательности, что значительно ускоряет получения результатов.

Whisper так же корректно работает с различными языками и не требует дополнительного обучения для понимания русского. Для обучения модели было использовано около 680 тысяч часов записи на разных языках.

1.2.3 Сравнение Whisper и DeepSpeech

Для сравнения данных моделей необходимо учесть следующие факторы:

1) процент ошибок в словах word error rate (WER);
2) объем данных, на которых обучена модель;
3) распознавание разных языков.

Модель DeepSpeech после обучения на датасете размером в 2300 часов аудиозаписей, получила 16% WER [10], что является неплохим показателем. В последствии было проведено обучение на большем объеме данных, размером около 7000 часов аудиозаписей, включающем в себя записи с шумом и помехами. После проведения тестов, средний WER уменьшился до 12%. Как уже было указано, DeepSpeech был обучен только на датасетах с английским языком, поэтому его использование потребует дополнительного обучения для распознавания русского языка.

В свою очередь, модель Whisper, обученная на колоссальном количестве данных, размером в 680 тысяч часов, предоставляет широкий объем различных данных касательно своего процента WER, для различных масштабов модели. В частности, даже на Whisper Medium, WER для русского языка составляет всего 9.3% [12]. Результат на моделях Whisper большего

масштаба еще лучше, однако, время обработки увеличится непропорционально приросту точности,

1.2.4 Языковые модели

Проведя анализ языковых моделей, среди релевантных нейронных сетей выделяются, по аналогии с задачей распознавания речи, два представителя уже упомянутых архитектур. Языковая модель Embeddings from Language Models (ELMo) устроена на основе РНС, в то время как Bidirectional Encoder Representations from Transformers (BERT) использует архитектуру Transformer, как и Whisper.

Так как устройство архитектур было рассмотрено на примере моделей DeepSpeech и Whisper, перейдем к сравнению показателей точности. Рассмотрев результаты сетей на маленьких объемах данных, точность РНС-сети будет немного выше, чем точность сети трансформерной – 70.08% против 67.15% [13].

Однако, если рассмотреть точность результатов на основе большего объема данных, будет видно, что разрыв между РНС и трансформерной сетью сокращается – 92.49% против 90.79% [14]. Это логично, так как BERT и другие модели изначально создавались под использование с большими объемами данных, этому также способствует то, что модели, использующие архитектуру Transformer, могут использовать параллельную обработку, в отличии от РНС.

Основным отличием между моделями является то, что ELMO может использоваться только для генерации контекстуальных эмбеддингов, в то же время как BERT может использоваться для решения задач целиком. Для решения задачи потребуется подключать дополнительные средства распознавания эмоций.

Однако, нужно учитывать то, что у модели BERT есть огромное количество различных вариаций и размерностей, в том числе поддерживающих русский язык, и даже специализирующихся на нем. Также, существуют разновидности моделей BERT задуманные для обработки сообщений из диалогов.

1.3Результаты обзора существующих моделей

Суммируем все изученные нами особенности моделей и сделаем выводы для того, чтобы определить итоговую пару моделей для реализации приложения.

1.3.1 Результаты обзора моделей преобразование речи в текст:

Обобщим полученные нами в ходе исследования данные для модели DeepSpeech от компании Mozilla:

1) архитектура модели – LSTM, что позволяет ей учитывать ранние шаги своей работы;
2) итоговая WER составляет 12%, что является крайне низким показателем;
3) эффективно справляется с распознаванием слов в аудиозаписях с шумами на фоне;
4) не имеет возможности распознавания русского языка, для этого придется проводить сложное дообучение;

Рассмотрим результаты, полученные во время изучения принципов модели Whisper от OpenAI:

1) архитектура модели – Transformer, что дает большое преимущество во время работы с seq2seq задачами;
2) имеет огромный объем данных обучения, собранный из самых различных источников, 680 тысяч часов аудиозаписей;
3) умеет обрабатывать русский язык;
4) имеет крайне низкий WER в 9,3%.

1.3.2 Вывод по моделям StT

Таким образом, проведя сравнение моделей, а также рассмотрев их ключевые особенности, можно сделать вывод, что Whisper является наилучшим решением, для перевода аудио в текстовый формат. Решающим фактором выбора Whisper является возможность распознавания русского языка, так как дообучение модели DeepSpeech будет сложным и крайне долгим процессом.

1.3.3 Результаты обзора языковых моделей:

Рассмотрев особенности модели ELMo можно сделать следующие выводы:

1) архитектура модели – LSTM, что позволит ей учитывать контекст даже очень длинных предложений;
2) имеет высокую точность на небольших объемах данных;
3) является лишь способом эмбеддинга предложений, потребуется дальнейшая обработка

BERT:

1) архитектура модели – Transformer, что позволит решить задачу эмоционального анализа текста целиком, за счет дообучения;
2) отлично работает на больших объемах данных, показывает высокие значения точности;
3) имеет большое разнообразие размерностей и специализированных версий, что положительно скажется на точности работы, а также на скорости обучения

1.3.4 Вывод по языковым моделям

В результате рассмотрения языковых моделей, был сделан вывод, что BERT лучше подходит для решения задачи, так как модель имеет возможность использоваться не только в качестве эмбеддера, но и как полноценный анализатор эмоциональной окраски за счет возможности проведения fine tuning.

1.3.5 Заключение

В результате проведенного анализа моделей, было принято решение использовать BERT и Whisper для решения данной задачи. Модели с архитектурой Transformer позволяют быстро и эффективно справляться с задачами обработки естественного языка. Модель Whisper не нуждается в дообучении, так как полностью готова для решения задачи перевода голоса в текст, однако есть необходимость выбрать подходящую размерность, для дальнейшего встраивания в приложение. Модель BERT, в свою очередь,

нуждается в дообучении, проведенном при помощи процесса fine tuning. Также, необходимо рассмотреть и выбрать разновидности и размерности модели, для того чтобы результаты дообучения позволяли эффективно определять эмоциональную окраску текста и имели высокую точность.

1.4Обзор разновидностей BERT

Так как BERT оказалась крайне эффективной моделью, она была адаптирована и улучшена для различных задач и языков. Существуют версии модели, оптимизированные под конкретные случаи, что позволяет использовать её в широком спектре приложений. Например, существуют специализированные версии BERT для русского языка, или направленные на анализ разговорной речи. Рассмотрим несколько наиболее подходящих для нашего приложения версий данной модели:

1) bert-base-multilingual – стандартная версия BERT, обученная на 102 языках, включая русский. Состоит из 12 слоев, содержит в себе более 110 миллионов параметров;
2) ai-forever/RuBert-base – модель BERT размерности base, созданная специализированно для русского языка. Она обучена на данных с абсолютно разных источников, в том числе разговоров: русскоязычные статьи википедии, данные новостных агентств (Лента, Газета и др.), книги, стихотворения, интернет-переписки, субтитры к различным медиа. Состоит из 12 слоев и имеет 178 миллионов параметров [15];
3) ruBert-Tiny2 – модель BERT размерности tiny на русском языке, является уменьшенной и дистиллированной версией bert-base-multilingual. Обучен на основе корпуса данных переводчика Яндекс, OPUS-100 и Tatoeba, суммарно около 2.5 миллионов коротких текстов. Главное различие – сокращение размера эмбеддинга с 768 до 312, а число слоев уменьшено с 12 до 3. В результате, размер модели составляет всего 45MB, а время обработки одного предложения достигает 6мс. Это вторая версия данной модели, в которой словарь

увеличен до 83828 токенов, а также увеличен максимальный размер последовательностей;

4) ruBert-base-conversational – русскоязычная модель BERT размерности base, обучена на данных субтитров, социальных сетей, и частям Taiga corpus. Имеет 180 миллионов параметров и 12 слоев.

На основе изученных данных, можно сделать вывод, что наиболее подходящими для решения задачи являются модели Ai-forever/RuBert-base и RuBert-Tiny2. Модель Ai-forever/RuBert-base обладает высокой точностью из-за большого объема обучающих данных. В то же время, RuBert-Tiny2, будучи уменьшенной и оптимизированной версией, предлагает компромисс между производительностью и скоростью обработки, что особенно полезно с учетом специфики использования модели в боте Telegram.

Мультиязычная версия BERT не сможет превзойти по качеству анализа текста специализированную версию для русского языка, это обусловлено более глубоким и специализированным обучением на разнообразных русскоязычных источниках данных, что позволяет моделям лучше понимать контекст и нюансы языка.

RuBert-base-conversational является почти полным аналогом Ai-forever/RuBert-base, однако последний был обучен на данных из больших источников. Так как нет однозначного ответа, насколько эффективнее обучение на более широком объеме данных.

Таким образом, в результате проведенного анализа можно утверждать, что для эффективного решения поставленной задачи, специализированные модели, такие как Ai-forever/RuBert-base и RuBert-Tiny2, являются наиболее подходящими, но RuBert-base-conversational также может оказаться эффективным

1.5Выбор палитры для обучения

Для облегчения анализа тональности текста, необходима составленная палитра эмоций человека, элементы которой будут достаточно сильно отличаться друг от друга, чтобы избежать спорных ситуаций, а также иметь

явные слова-маркеры, обозначающие конкретную эмоцию, для возможности более точного распознавания [16].

Одной из самых ранних работ на эту тему является книга Уильяма Джеймса “Принципы психологии”, вышедшая в 1890 году. Теория эмоций Джеймса на тот момент была крайне новаторской, он утверждал, что эмоции являются не более чем физиологическими реакциями на внешние стимулы, а также был уверен, что они предшествуют когнитивной оценке ситуации. Помимо этого, им были выдвинутые четыре основные эмоции, такие как радость, печаль, гнев и страх. Данный набор является самыми базовыми ощущениями, они покрывают большой спектр человеческих чувств, а также являются обособленными и отличаемыми друг от друга. Радость и печаль являются противоположными, с явными текстовыми обозначениями эмоции, что крайне благоприятно будет влиять на их определение. Гнев и страх в свою очередь являются крайне специфичными, и по своей природе более яркими относительно других эмоций. Соответственно, их определение является более простой задачей, и увеличит точность при анализе тональности. Данные четыре эмоции в последствии будут использоваться и в других теориях, соответственно их можно принять за основу палитры для решения задачи.

Несмотря на то, что выявленные Джеймсом основные эмоции были бы крайне удачным выбором для определения тональности текста, они упускают многие другие чувства, которые может испытывать человек. В конце XX века, профессор Пол Экман, американский психолог, выдвинул свою теорию эмоций, которая в том числе ссылается на теорию Джеймса, и дополняет ее. Он утверждает, что базовые эмоции являются универсальными среди людей любых культур и народов. В основе его теории лежат 6 различных эмоций, таких как основные злость, радость, страх, печаль, а также дополнительные отвращение и удивление. Однако, помимо основной четверки, упомянутой еще Уильямом Джеймсом, отвращение и удивление являются гораздо более комплексными эмоциями. Крайне сложно выделить слова или словосочетания, напрямую передающее отвращение. Удивление, в свою

очередь, является проблематичным из-за схожести с радостью. Также, в контексте оно может быть похоже на испуг, т.к. в основе обеих эмоций лежат разрушенные ожидания или внезапность, что значительно затруднит классификацию предложений. Данные эмоции будет проблематично определять используя исключительно текст, так основной способ их выражения напрямую связан с мимикой.

Рассмотрим другую работу Экмана, в которой он выделяет список из 15 основных эмоций. Так же, как и с презрением и удивлением, многие чувства из списка с трудом поддаются идентификации при помощи текста, например гордость или смущение. Они либо частично пересекаются с основными эмоциями, либо имеют большие сложности в выявлении напрямую передающих чувства слов (слов-идентификаторов). Но также, в списке присутствуют такие эмоции как веселье и стыд, которые соответствует всем требованиям, необходимым для использования в решении задачи. Они проявляются достаточно ярко, для того чтобы их было удобно классифицировать, а также сильно отличаются от других эмоций, которые мы уже добавили в палитру.

Таблица 1 - Выделенные эмоции и примеры слов-идентификаторов

Эмоция	Радость	Грусть	Злость	Страх	Веселье	Стыд	Отвращение	Удивление
1	2	3	4	5	6	7	8	9
Слова для идентификации в тексте	Круто, классно, радость, приятно, нравится, лучший, весело, невероятно, удивительно, вау	Печально, грустно, тяжесть, расстроиться, жалкость, сложность, гадость	Злиться, ругательства, раздражать, ярость, рассердиться, возмутиться, ненавидеть	Бояться, страшно, боязнь, испугаться, жутко, стремно, тревожиться, внезапно, неожиданно	Смешно, угар, весело, смеяться, забавно, ржать	Прости, виноват, жалкость, неловкость, стеснятьсяя, стыдитьсяя	Гадость, ненавидеть, раздражать, противность	Невероятность, удивительно, внезапность, неожиданность

Продолжение таблицы. 1

1	2	3	4	5	6	7	8	9
Количество	8	7	7	8	6	6	4	4
Количество совпадений	3	1	2	2	1	1	3	4

Рассмотрев все упомянутые эмоции и их слова-идентификаторы, представленные в таблице 1, можно прийти к выводу, что эмоции отвращения и удивления не подходят для решения задачи, из-за высокого процента совпадений слов идентификаторов и другими эмоциями. Веселье и стыд были добавлена в палитру, т.к. они легче всего отличаются от выбранных нами раннее чувств.

Таким образом, после изучения трудов Экмана и Джеймса и сравнительного анализа полученных эмоций, мы можем прийти к выводу, что необходимая нам палитра эмоций состоит из злости, радости, страха, печали, веселья и стыда. Однако, учитывая специфику задачи, а именно необходимость анализа текстовых данных, нужно принимать во внимание то, что часть текста не несет в себе какой-либо эмоциональной окраски, и является, например, сухой констатацией какого-либо факта. Чтобы повысить точность определения тональности текста, необходимо добавить в палитру нулевую эмоциональную окраску. Назовем ее нейтральностью.

В результате исследования была проведена идентификация ключевых эмоций человека для решения задачи определения тональности текста, в результате чего получили палитру из 7 эмоциональных состояний: нейтральность, радость, печаль, злость, страх, веселье, стыд.

2 ПРОЕКТИРОВАНИЕ

Во втором разделе данной работы рассмотрено проектирование бота для веб-приложения Telegram, анализирующего эмоции в полученном голосовом сообщении.

Для корректной реализации бота, необходимо определить функциональные требования к приложению, основные компоненты, их взаимосвязи, а также происходящие в нем процессы. Это включает в себя разработку схемы архитектуры, определение ролей и задач каждого компонента, а также построение последовательности операций, необходимых для достижения функциональности. Для этого необходимо построить диаграммы, описывающие работу будущего приложения, и определить итоговую архитектуру.

2.1Функциональные требования

Реализация выбранного бота должна выполнять следующие функциональные требования:

1) бот должен при запуске командой /start отправлять приветственное сообщение;
2) бот должен в реальном времени обрабатывать и определять тип приходящего сообщения;
3) в случае неправильного типа, бот должен уведомлять пользователя об ошибке;
4) бот должен обрабатывать входящее голосовое сообщение;
5) обработка должна переводить голосовое сообщение в текстовый формат;
6) текст должен анализироваться на эмоциональную окраску;
7) результат работы бота должен отправляться пользователю.

Выполнение всех указанных функциональных требований позволит решать обозначенную задачу.

2.2Диаграмма компонентов UML

Диаграмма компонентов в нотации UML представлена на рисунке 2.

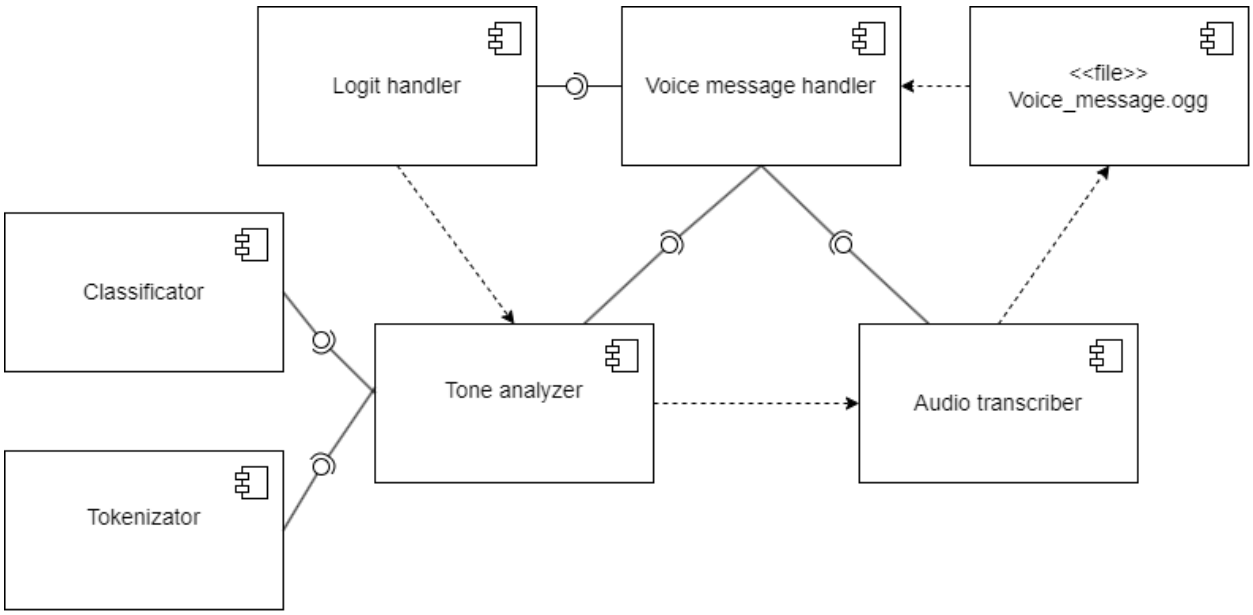

Рисунок 2 – Архитектура бота в диаграмме UML

Основными компонентами архитектуры являются:

1) voice message handler – компонент, отвечающий за взаимодействие с данными пользователя. Основной задачей компонента является обработка приходящих сообщений. Достигается это при помощи взаимодействия с библиотекой Telebot, которая предоставляет простой интерфейс взаимодействия с Telegram Bot API. Функционал библиотеки позволяет обрабатывать сообщения любого типа, настраивать шаблоны реагирования на эти сообщения. При инициализации чата, должно отправляться стартовое сообщение. Все сообщения от пользователя, кроме голосовых, должны вызывать ошибку, отправляемую пользователю. В случае получения голосового сообщения, компонент должен определять имя необходимого файла и загружать его, а затем вызывать работу следующего компонента;
2) audio transcriber – компонент, отвечающий за обработку аудиофайла. Попадающая в данный компонент аудиодорожка должна проходить обработку моделью Whisper. В процессе обработки, аудиофайл переводится в текстовый формат и подается в следующий компонент;

3) tone analyzer – компонент, производящий анализ тональности. Для обработки текста необходимо произвести токенизацию. Для этого текст поступает в компонент Tokenizator, в котором текст разбивается на токены. Далее, токены необходимо закодировать их порядковыми индексами в словаре, добавив PAD-токены и другие специальные токены. Затем, полученные индексы и список attention mask, указывающий, является ли данный токен PAD-токеном, должны быть конвертированы в тензоры и поданы в компонент Classificator. В нем происходит классификация входящего текста, загруженная модель выдает предсказание о том, какую эмоциональную окраску несет текст. Полученное предсказание передается в следующий компонент;
4) logit handler – компонент, отвечающий за обработку получившихся предсказаний, и отправляющий пользователю результат работы бота. Приходящие в него тензоры переводятся в формат массивов, сравниваются между собой, и по результатам сравнения пользователю отправляется результаты работы компонентов audio transcriber и tone analyzer.

2.3Проектирование процессов IDEF0

Для описания всех процессов, обозначенных в компонентах нашей архитектуры, была построена диаграмма в нотации IDEF0.

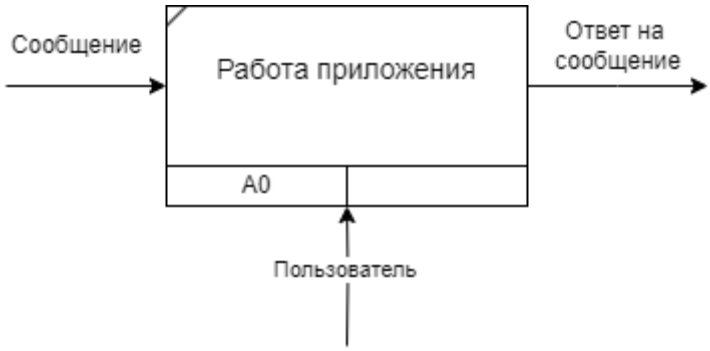

Рисунок 3 – Основная схема

На рисунке 3 изображена основная схема работы бота. Главный принцип работы – получение сообщения, обработка и отправка ответа.

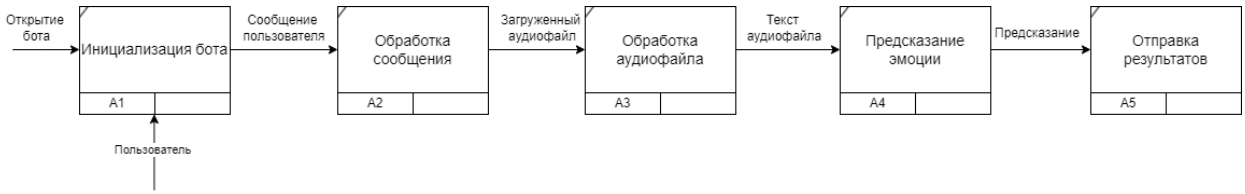

Рисунок 4 – Декомпозиция A0

Декомпозиция A0 на рисунке 4 изображает основные процессы, происходящие в работающем приложении. Ими являются следующие этапы работы:

1) инициализация бота – процесс запуска диалога с ботом, посредством введения команды /start;
2) обработка сообщения – процесс получения необходимой информации из сообщения пользователя;
3) обработка аудиофайла – процесс транскрибации аудиофайла при помощи модели Whisper, для дальнейшего анализа;
4) определение эмоции – процесс анализирования текста на эмоциональную окраска при помощи обученной модели BERT;
5) отправка результатов – процесс обработки предсказания и формирования сообщения для пользователя.

Рисунок 5 – Декомпозиция A1

Порядок проведения инициализации бота Telegram описан в декомпозиции A1, на рисунке 5. Этот этап происходит во время первого взаимодействия пользователя с ботом. Основные шаги– определение команды /start, и ожидание сообщений пользователя.

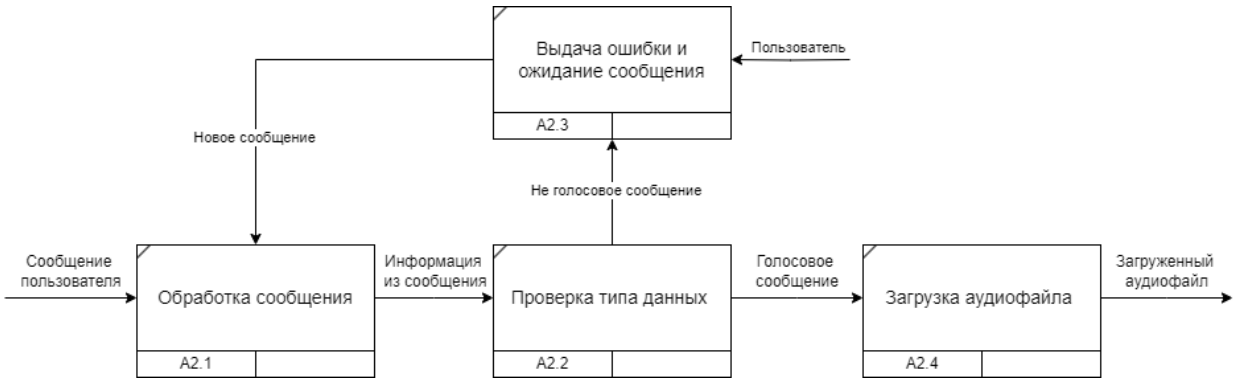

Рисунок 6 – Декомпозиция A2

Процесс обработки сообщения рассмотрен в декомпозиции A2, на рисунке 6. В данном процессе бот получает необходимую информацию из сообщения пользователя, проверяет тип данных, и либо выдает пользователю ошибку, либо загружает аудиофайл.

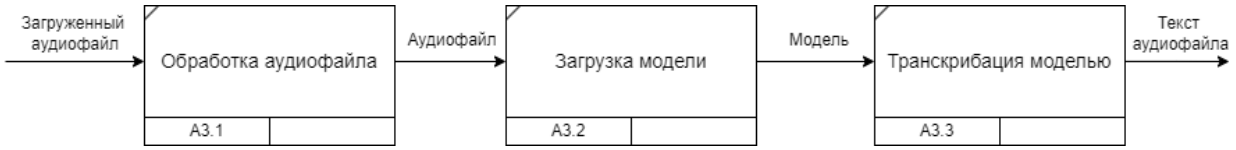

Рисунок 7 – Декомпозиция A3

В декомпозиции A3, на рисунке 7, проиллюстрирован процесс транскрибации аудиосообщения. Загружается модель Whisper, которая переводит аудиофайл в спектрограмму, и на основе этой спектрограммы предсказывает текст, который в дальнейшем будет отправлен на определение эмоции.

Рисунок 8 – Декомпозиция A4

Декомпозиция A4, на рисунке 8, представляет основной этапы работы дообученной модели BERT, токенизацию и классификацию. В процессе токенизации текст переводится в порядковые индексы распознанных токенов. В процессе классификации, токены и их лейблы проходят обработку моделью и на выходе возвращает тензор с предсказанием. В результате работы данного процесса дальше отправляется полученное предсказание.

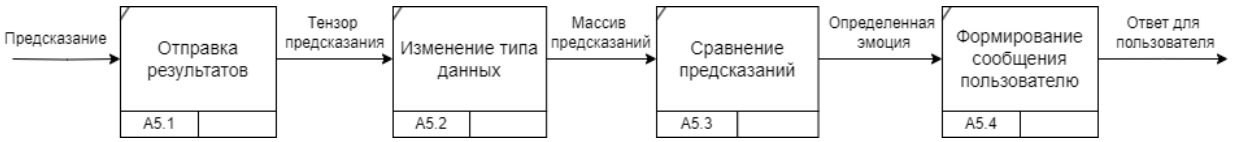

Рисунок 9 – Декомпозиция A5

Итоговое формирование ответа рассмотрено на рисунке 9, в декомпозиции A5. Для реализации удобного сравнения предсказаний, полученный тензор конвертируется в массив, затем элементы массива сравниваются между собой и по результатам сравнения выбирается наиболее высокое значение эмоции. Затем, пользователю отправляется ответ с полученной эмоцией и результатом транскрибации.

2.4Результаты проектирования

В результате проведенного проектирования, нами были сформулированы функциональные требования, а также выстроена архитектура приложения. Бот состоит из 4 основных компонентов, которые включают в себя Voice message handler, Audio transcriber, Tone analyzer и Logit handler. Процесс проектирования включал в себя описание архитектуры системы, построение UML диаграмм компонентов, а также моделирование процессов в нотации IDEF0.

Все процессы, происходящие во время работы бота, были описаны и визуализированы в диаграмме IDEF0, что помогло выделить основные функциональные блоки, их взаимосвязь друг с другом, а также последовательность исполнения. На основе составленных диаграмм и описаний можно производить реализацию приложения.

3 ФОРМИРОВАНИЕ ДАТАСЕТА

Формирование датасета это один из важнейших шагов для успешного обучения модели, так как качество и структура набора данных будет влиять на точность работы модели.

В данной главе будут рассмотрены принципы формирования датасетов, описан процесс составления, редактирования и подготовки данных для обучения. Также, необходимо рассмотреть существующие наборы данных, которые подойдут для решения задачи, сравнить их и выбрать лучший, для дальнейшей обработки.

3.1Свойства качественного набора данных

При формировании датасета для обучения модели нейронной сети необходимо обращать внимание на следующие свойства, которые позволят качественно выделить подходящий для задачи набор данных:

1) соответствие поставленной задаче – так как задача в своей основе будет требовать обработки коротких, ярко окрашенных эмоциями, сообщений, для нее необходимы соответствующие данные для обучения;
2) репрезентативность – данные должны представлять все виды рассматриваемых в задаче эмоций. Таким образом, можно избежать ситуации, когда некоторые эмоции определяются гораздо хуже других;
3) качество маркировки – маркированные эмоции должны соответствовать содержаниям сообщений. Некорректная маркировка может привести к искажению результатов обучения;
4) качество текста – данные должны быть проверены на ошибки и пропущенные маркировки;
5) оличество текста – количество текста должно быть достаточным для обучения модели;

Итоговый датасет для обучения модели должен как можно точнее соответствовать описанным принципам. В случае их соблюдения, итоговая точность определения эмоциональной окраски текста будет расти.

3.2 Анализ существующих датасетов

В открытом доступе находится множество наборов данных, которые содержат в себе различную информацию в любых выбранных объемах. Для обучения модели среди них необходимо найти наиболее подходящий датасет. Количество подходящих датасетов также увеличивает популярность задачи классификации текста в машинном обучении. Среди всех наборов данных, можно выделить 3 подхода к маркированию: бинарное, тернарное и по отдельным эмоциям.

1) бинарное маркирование – датасет содержит в себе данные, обозначенные только 2 эмоциями: позитивно, негативно. Работа с таким датасетом не подходит под палитру из 7 эмоций, выделенных раннее, однако может использоваться для дополнения основного датасета новым данными;
2) тернарное маркирование – аналогично бинарному, однако в маркировку добавляется нейтральность. Его можно также использовать для дополнения датасета, в случае дополнения нейтральными данными повторная маркировка не потребуется. Наиболее частый вид датасетов;
3) маркирование по эмоциям – самый сложный вид маркировки данных, так как в тексте может быть выделено любое количество эмоций. Главной проблемой датасетов этого вида является неточность. Также, они являются довольно редкими, из-за тяжелого процесса составления.

Рассмотрим несколько открытых источников, таких как Hugging Face и Kaggle, на предмет наличия соответствующего запросам набора данных.

Таблица 2 – Датасеты из открытых источников

Название датасета	Краткое описание	Количество данных, строки
1	2	3
Russian Emotional Speech Dialogs – Kaggle	Датасет переведенных в текст записей разговоров, промаркированных 7 эмоциями на русском языке	1112
Russian_twitter_sentiment - Kaggle	Датасет бинарной маркировки, содержит позитивные и негативные сообщения из соцсети Twitter на русском языке	274890
Deysi/sentences-and-emotions - Hugging Face	Датасет предложений с маркировкой на 7 различных эмоций, на английском языке	3714
blinoff/kinopoisk - Hugging Face	Датасет рецензий фильмов сервиса Кинопоиск, тернарная маркировка	36,591
Djacon/ru-izard-emotions - Hugging Face	Датасет переведенных на русский язык сообщений Reddit, маркировка на 10 различных эмоций	24,891
Romjiik/Russian_bank_reviews - Hugging Face	Датасет содержащий в себе отзывы к банкам и их оценку, на русском языке	12,392
fhamborg/news_sentiment_newsmtsc - Hugging Face	Датасет новостей на английском языке, тернарная маркировка эмоциональной окраски	110000
MonoHime/ru_sentiment_dataset - Hugging Face	Датасет сообщений с тернарной маркировкой, на русском языке	210,989

Рассмотрим перечисленные открытые датасеты, представленные в таблице 2, и определим среди них наиболее подходящий для дальнейшей обработки:

Russian Emotional Speech Dialogs – датасет содержит в себе небольшие, эмоционально окрашенные сообщения, которые являются транскрибацией живых диалогов людей. Маркировка состоит из 7 значений: злость, отвращение, страх, энтузиазм, радость, нейтрально, грусть. Эмоции не полностью совпадают с той палитрой, что была составлена для решения поставленной задачи, однако часть данных могла бы быть полезна для дальнейшего использования в связи с тем, что данные из набора являются живой человеческой речью. Общий объем данных в 1112 строк является недостаточным для качественного процесса обучения.

Russian_twitter_sentiment – огромный датасет из эмоционально окрашенных сообщений пользователей социальной сети Twitter на русском языке. Данные поделены на позитивные и негативные. Бинарная классификация не позволяет использовать оцененные сообщения для определения необходимых эмоций. Однако, объем данных в 274890 сообщений позволяет использовать данный датасет для отбора текстовых сообщений в другой набор данных.

Deysi/sentences-and-emotions - Hugging Face – датасет из 3714 сообщений на английском языке, поделенных на 7 эмоций: нейтральность, злость, грусть, радость, отвращение, удивление и страх. Эмоции не полностью совпадают с выбранной палитрой, а распределение данных, где 84% датасета занимают нейтральные и радостные сообщения. Эти факторы негативно скажутся на точности определения эмоциональной окраски текста.

blinoff/kinopoisk – набор данных, содержащий в себе большое количество рецензий на фильмы с платформы Кинопоиск. Предоставляет как тернарную оценку текста, так и десятибалльную, которую при необходимости можно конвертировать тернарную оценку с градацией. Рецензии представляют собой тексты большого объема, оценивающие фильм. Объем

данных, содержащихся в данном датасете огромен из-за 36,591 строк и большого количества слов в одной строке текста. Несмотря на это, использование текста из данного датасета является нецелесообразным. Из-за специфики слов, описывающих качества фильма, их значение будет искажено для текстов, содержащих в себе разговоры людей.

Djacon/ru-izard-emotions – датасет, содержащий в себе комментарии с платформы Reddit из разных категорий, переведенных на русский язык с большой точностью, благодаря модели DeepL. Данные промаркированы следующими эмоциями: нейтральность, радость, грусть, гнев, интерес, удивление, отвращение, страх, вина, стыд. Разнообразие выделенных в датасете эмоций почти целиком покрывает выбранную нами палитру, что делает этот набор данных удачным для использования в обучении. Также, то, что тексты сообщений собирались из разных тем форума, позитивно повлияет на точность оценки эмоциональной окраски, т.к. затрагивает множество различных тем, на которые люди ведут диалоги. Однако, из-за того, что марикровка изначально производилась на английском языке, эмоции английского сообщения могут не соответствовать эмоциям перевода этого сообщения на русский. Кроме того, несмотря на точность перевода моделью Deep L, в текстах датасета наблюдается некоторое количество артефактов, непереведенных слов или длинных последовательностей символов, которые будут снижать качество обучения модели.

Romjiik/Russian_bank_reviews – датасет, содержащий в себе рецензии пользователей на различные банки с сайта агрегатора, на русском языке. Состоит из 12,392 строк, в которых содержится тексты рецензий, а также пятибалльные метрики оценки банка. Данные из этого набора можно было бы конвертировать в тернарную оценку, однако специфика рецензий на банки не позволит такой конвертации быть точной. К тому же, распределение оценок будет неподходящим, из-за того, что крайне негативных и крайне позитивных оценок будет гораздо больше средних. Также, крайне негативные

комментарии будут проявлять гораздо больше злости, чем крайне позитивные – радости.

fhamborg/news_sentiment_newsmtsc – датасет состоящий из новостей на английском языке, а также их эмоциональной окраски в тернарной классификации: хорошая, плохая, нейтральная. Несмотря на огромный объем данных в 110 тысяч строк, использование данных из данного датасета является нецелесообразным, ввиду сильного отличия формата новости от формата сообщения, анализируемого в поставленной задаче, даже если произвести качественный перевод с английского языка.

MonoHime/ru_sentiment_dataset – данный датасет является компиляцией из множества других наборов данных, с тернарной классификацией. В нем можно найти сообщения с различных форумов, таких как Пикабу, рецензии, новости на русском языке и данные из тонального словаря. Из-за такой огромной разницы в сферах, откуда были взяты тексты для датасета, в нем можно встретить, как и подходящие для решения поставленной задачи короткие, эмоционально окрашенные сообщения, так и огромные новостные статьи, которые будут мешать точности обучения. Множество данных не будет возможно использовать из-за того, что они описывают специфичные объекты, например машины. Таким образом, данный датасет может служить неплохим источником текста для дополнения, но использование как основы для формирования своего набора данных является нецелесообразным.

3.3Итоги анализа готовых наборов данных

В результате проведенной оценки открытых датасетов, можно сделать вывод, что возможность обучения на уже готовом наборе данных отсутствует. Для качественного обучения модели, необходимо взять уже существующий датасет и доработать его под поставленную задачу. Наиболее удачным выбором для основы нашего датасета будет являться набор данных с маркировкой по эмоциям, чтобы соответствовать выделенной палитре. Кроме того, контекст задачи предполагает работу с короткими, эмоционально

окрашенными сообщениями, что тоже следует учитывать при выборе датасета.

Под поставленные условия подходят три датасета: Russian Emotional Speech Dialogs, Deysi/sentences-and-emotions, Djacon/ru-izard-emotions. Однако, первые два набора содержат в себе недостаточное количество строк с данными для того, чтобы формировать основную часть необходимого нам датасета. В то же время третий датасет наиболее точно соответствует нашей палитре эмоций: нейтральность, радость, печаль, злость, страх, веселье, стыд. В нем присутствуют маркировки со всеми этими эмоциями, кроме веселья, которое в наборе ассоциировано с радостью. В связи с этим, было принято решение выбрать в качестве основы для доработки датасет Djacon/ru-izard-emotions.

Упомянутые ранее два датасета, а также наборы данных Russian_twitter_sentiment и MonoHime/ru_sentiment_dataset, несмотря на свою бинарность и тернарность соответственно, отлично подойдут как источники данных для дополнения датасета, в случае возникновения такой необходимости.

3.4Доработка выбранного датасета

Следующим этапом для составления собственного датасета было взятие выбранного основного набора данных и произведение повторной обработки данных. Из Djacon/ru-izard-emotions были вытащены 15 тысяч сообщений для дальнейшей маркировки и корректировки.

Для удобства процесса обработки данных, лишние значения предыдущей маркировки были удалены, а затем, сообщения были отсортированы по оставшимся необходимым эмоциям. Таким образом, повторный анализ текста на эмоции происходил быстрее.

Каждому сообщению было присвоено новое значение, соответствующее палитре, составленной в пункте 1.5 данной работы, каждое из которых означает ту или иную эмоцию. Значения присваивались в диапазоне от 0 до 6, где 0 – нейтральность, 1 – радость, 2 – печаль, 3 – злость, 4 – страх, 5 – веселье,

6 – стыд. Помимо обработки, происходила корректировка данных, так как в датасете обнаруживались следующие виды ошибок:

1) неполный перевод – из-за того, что данные изначально были на английском языке, и несмотря на общую точность работы модели DeepL, некоторое количество записей остались непереведенными, либо переведенными лишь частично. Подобные сообщения переводились вручную;
2) длинная последовательность из символов – так как датасет составлен на основе сообщений форума, написанных живыми людьми, в него случайно могли попасть сообщения, не несущие смысловую нагрузку, либо частично заполненные не текстом, пример такого изображен на рисунке 10. Если сообщение целиком состояло из символов, оно удалялось, если частично – символы удалялись;
3) нечитаемые сообщения – из-за особенностей перевода и формирования датасета, имена, фамилии и большая часть названий заменялись специальными конструкциями. Иногда из-за них сообщение становилось нечитаемым. В таких случаях, удалялись либо конструкции, либо сами записи;
4) неопределяемая эмоция – эмоциональную окраску некоторых сообщений было невозможно определить из-за наличия в одном тексте абсолютно противоположных по значению и передающим эмоциям слов. Такие сообщения либо маркировались нейтральностью, либо удалялись;
5) наличие хэштегов и ссылок – в сообщениях форумов, зачастую пользователи приводят хэштеги или ссылки. Такие элементы в тексте будут мешать обучению, так как не несут смысловой нагрузки, написаны на английском языке, а также используют символы. Если сообщение целиком состояло из них, данные удалялись, в случае частичного наличия удалялись подобные элементы.

Reddit I'm blind, what's the bottom right?
Я так давно !?!!"!!!!"!!!"!!!"!!
[NAME] такой [NAME], [NAME]
Гнев доброта слабость
[ИМЯ] все было ужасно. я предполагаю, что они все еще там. #teamlemoncreams

Рисунок 10 – Пример типичных ошибок в датасете

3.5Результаты формирования датасета

В результате обзора и обработки существующих наборов данных, был сформирован собственный маркированный датасет, специализированный под решение задачи определения эмоциональной окраски текста для бота Telegram. После полного редактирования данных датасета Djacon/ru-izard-emotions, из выборки в 15000 данных осталось 12600 записей. Была проведена коррекция и полная маркировка текста, удалены ошибки и переведены английские фразы. Процесс повторной обработки данных позволил существенно улучшить качество датасета. Получившийся набор данных является окончательным, и будет использован в дальнейшем для обучения модели BERT.

4 РЕАЛИЗАЦИЯ

На основе проведенного анализа предметной области, выполненного проектирования и сформированного датасета, в данной главе будет представлена реализация чат-бота Telegram для распознавания эмоций в голосовых сообщениях. Основными задачами являются:

1) проведение дообучения моделей Ai-forever/RuBert-base, RuBert-Tiny2, RuBert-base-conversational;
2) сравнение их результатов после обучения на одном и том же датасете;
3) определение размерности модели Whisper;
4) разработка бота для Telegram, использующего наиболее эффективные модели;

4.1Процесс обучения моделей

Основной идеей для реализации обучения является процесс fine-tuning для модели BERT. Языковая модель на выходе создает векторные представления для слов, и даже целых фраз. Добавив сверху над такой языковой моделью небольшой блок из пары дополнительных слоев нейронов, можно дообучить эту нейронную сеть на любые задачи.

Для того, чтобы дообучить модель в кратчайшие сроки, было принято решение использовать Google Colab. Это позволит использовать большие вычислительные мощности, так как Google дает возможность подключить их собственный высокопроизводительный GPU Tesla T4. Подключить его можно при помощи библиотеки PyTorch.

После импортирования всех зависимостей и настройки используемых ресурсов, необходимо загрузить датасет. Для этого используется библиотека Pandas, которая представляет собой одно из удобнейших средств для взаимодействия с датасетами в Python. Для дальнейшей обработки, необходимо данные из набора превратить в токены.

4.1.1 Токенизация

Каждая модель BERT имеет свой собственный токенизатор, так как каждая у каждой модели есть словарь с использующимися в нем индексами.

Чтобы избежать ошибок с неверным определением индексов, необходимо инициализировать токенайзер выбранной модели. Далее, для конвертации текста в токены, нам необходимо достать все данные из датасета в два списка, массив текстов для обучения и массив маркировок. Список текстов как раз и должен пройти токенизацию и вернуться в виде тензоров библиотеки PyTorch.

Во время обработки нам необходимо, чтобы токенизатор добавил специальные токены:

1) токен [SEP] – обозначает конец предложения;
2) токен [CLS] – обозначает начало предложения;
3) токен [PAD] – обозначает токен паддинга.

Паддинг является обязательным условием для дальнейшей обработки моделью BERT. Из-за разницы в длине текстов, токенизатор должен их уравнять по длине до какого-то конкретного значения, обозначаемого max length. Чтобы определить, до какого значения необходимо уравнять все остальные предложения, узнаем максимальную длину текста в нашем датасете. Далее, необходимо взять ближайшие степени двойки, степень больше и степень меньше, и выбрать из них ближайшую, однако приоритетно выбирать большую, чтобы не потерять важную информацию. Однако, чем больше будет длина сообщений, тем больше времени займет обработка моделью. В случае с нашим датасетом, самым оптимальным значением max length будет являться 128. Выбрав его, мы не потеряем никаких данных, а разница в скорости обучения будет незначительной.

Само уравнивание происходит при помощи [PAD]-токенов. Такие токены имеют индекс 0, и в последствии будут игнорироваться при помощи Attention Mask – специального списка, который поможет определить какой из токенов в предложении является [PAD]-токеном и должен быть пропущен, что представлено на рисунке 11.

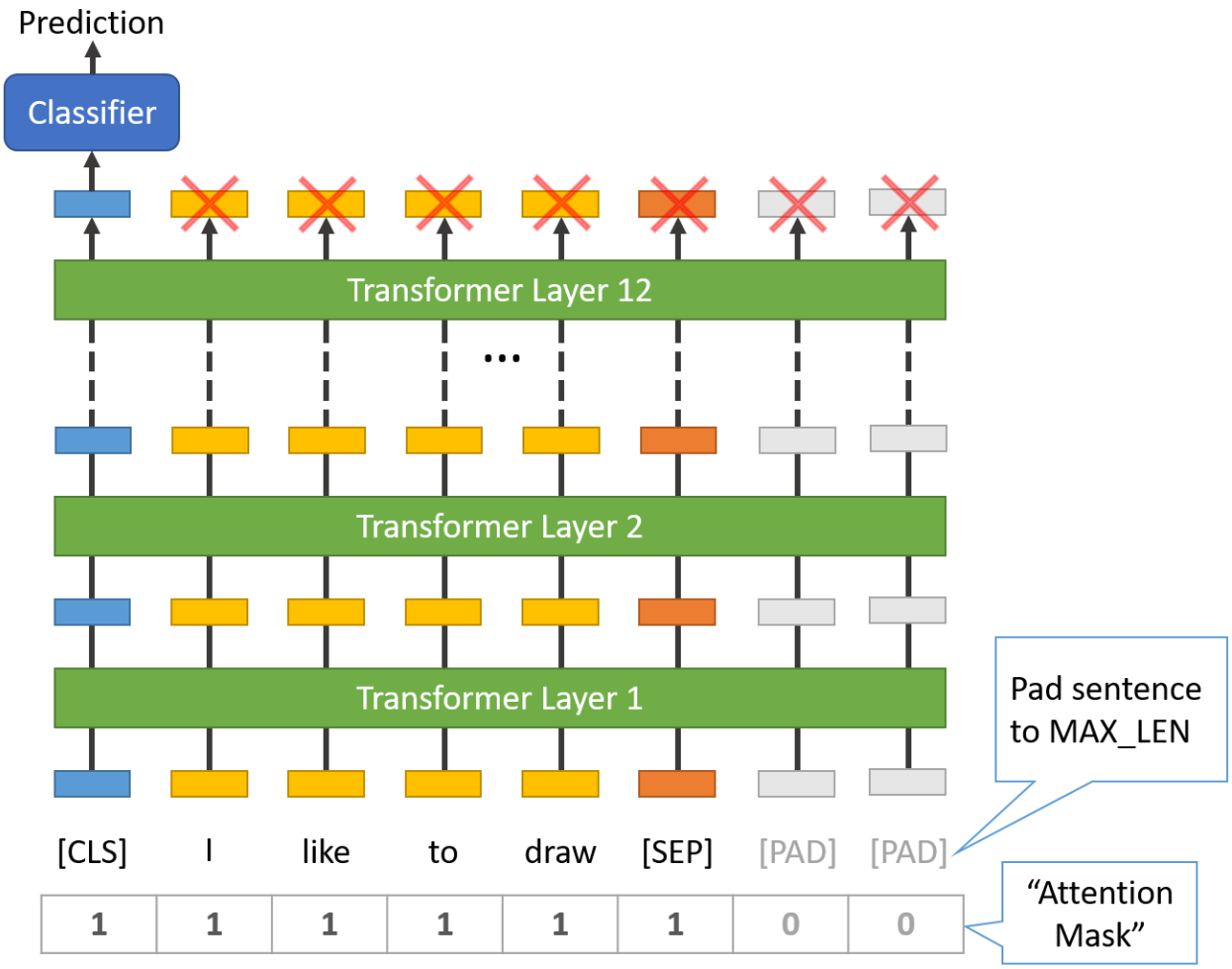

Рисунок 11 – Токены, паддинг и Attention Mask

Все описанные процессы выполняются функцией tokenizer.encode_plus, с параметрами: add_special_tokens = true, max_length = 128, pad_to_max_length = True, return_attention_mask = True, return_tensors = 'pt'.

По итогам обработки, мы получим два массивы – массив тензоров с идентификаторами токенов, и массив тензоров Attention Mask. Полученные массивы, а также массив маркировок, необходимо конвертировать в тензоры.

4.1.2 Настройка данных, загрузка модели и параметров обучения

Тензоры, полученные после проведения токенизации необходимо превратить в датасет тензоров из библиотеки Torch, после чего полученный набор данных должен быть разделен на train-часть, на которой модель будет обучаться, и на validation-часть, которая будет задействована во время валидации. Разделение происходит в соотношении 9 к 1, при помощи функции random_split.

Полученные части датасетов необходимо обернуть в специальный класс DataLoader, который будет последовательно загружать данные в модель в батчах, чей размер необходимо установить параметром batch_size. Данный параметр создателями модели BERT рекомендуется устанавливать в значении 16 или 32, однако для небольших объемов данных, подойдут и меньшие значения. Batch_size является одним из основных инструментов настройки обучения наших моделей.

Для начала обучения, модель необходимо загрузить. Происходит это при помощи класса BertForSequenceClassification из библиотеки Transformers, который добавляет к обычной модели классификационный слой, при загрузке также необходимо указать количество маркировок. При использовании этого класса, необходимо указать, какую конкретно модель мы будем использовать в качестве основы для обучения. Для быстроты обработки, также необходимо указать, что PyTorch должен запускать ее, используя GPU.

Одним из важнейших элементов в обучении также является optimizer, алгоритм необходимый для изменения параметров обучения, в нашем случае данный параметр learning rate, который напрямую влияет на скорость обучения, что при правильном выставлении увеличит точность работы получившейся модели. В документации BERT, оптимальным значением является 2e-5. Learning rate является еще одним важным параметром проводимого обучения. В случае нашей программы, в качестве инструмента выставления параметра будет использоваться метод AdamW.

Для того, чтобы скорость обучение не была изначально крайне высокой, необходим компонент scheduler, который разделит программу на определенное количество шагов, определяемое количеством батчей и количеством эпох, циклов проведения обучения, и будет постепенно изменять значение learning rate в процессе обучения. Таким образом, здесь вводится третий основной гиперпараметр – количество эпох.

4.1.3 Обучение модели

После выбора гиперпараметров модели и настройки параметров seed, начинается процесс петли обучения. Она делится на две части: тренировку и валидацию. Тренировка выставляется при помощи model.train() и представляет собой цикл следующих шагов:

1) данные вытаскиваются из загрузчика;
2) обнуляются градиенты предыдущего шага;
3) данные из одного батча отправляются в модель;
4) ошибки по результатам исполнения модели проходят в обратную сторону и на основе этого корректируются веса;
5) корректируются параметров модели при помощи optimizer;
6) корректируется параметр learning rate при помощи scheduler;
7) обновление статистики обучения.

После тренировки, происходит процесс валидации. Суть его в том, чтобы проверить, какие результаты имеет модель за эпоху обучения, происходит сбор статистики и подсчет точности. Один цикл валидации состоит из следующих шагов:

1) данные вытаскиваются из загрузчика;
2) данные из одного батча отправляются в модель;
3) пересчет потерь и статистики.

Эти два этапа повторяются каждую эпоху. В результате, мы получаем обученную модель, а также собранную статистику по каждой эпохе, что позволяет отслеживать эффективность гиперпараметров и рисовать графики, необходимые для определения эффективности обучения.

4.2 Сравнение обученных моделей

Для того, чтобы выделить наиболее эффективную модель необходимо провести обучение на подходящих для конкретной нейронной сети гиперпараметрах, и сравнить итоговый показатель accuracy на специально составленном тестовом датасете. Было проведено множество попыток обучения, и по 4 лучшим показателям был сформирован результат сравнения.

В результате подбора параметров были достигнуты следующие значения параметра accuracy, представленные в таблице 3.

Таблица 3 – Результаты подбора гиперпараметров

Модель	Гиперпараметры	Accuracy
RuBert-Base	batch size = 16, learning rate = 2e-5, epoch = 2	0,72
	batch size = 16, learning rate = 2e-5, epoch = 3	0,71
	batch size = 8, learning rate = 2e-5, epoch = 2	0,75
	batch size = 8, learning rate = 1.5e-5, epoch = 2	0,69
RuBert-tiny2	batch size = 8, learning rate = 2,5e-5, epoch = 2	0,66
	batch size = 8, learning rate = 2e-5, epoch = 2	0,68
	batch size = 8, learning rate = 1.5e-5, epoch = 2	0,66
	batch size = 4, learning rate = 2e-5, epoch = 4	0,7
RuBert-base-conversational	batch size = 16, learning rate = 2e-5, epoch = 2	0,69
	batch size = 16, learning rate = 2e-5, epoch = 2	0,69
	batch size = 8, learning rate = 1e-5, epoch = 2	0,7
	batch size = 8, learning rate = 3e-5, epoch = 2	0,71

Наилучший результат продемонстрировала модель RuBert-Base, параметры batch size = 8, learning rate = 2e-5, epoch = 2. Ее точность составила 0.75.

4.3Определение размерности Whisper

Для того, чтобы определить, какой размер модели использовать в работе бота, необходимо провести анализ моделей Whisper 4 размеров: tiny, base, small и medium. Запишем аудиозапись с чтением на 30 секунд, и проведем транскрибацию, для того чтобы узнать, насколько качественно произойдет перевод аудиофайла в текст, и сколько времени это займет. Результаты работы модели представлен в таблице 4.

Таблица 4 – Результат работы модели Whisper в 4 размерах

Распознанный текст	Время обучения	Модель
1	2	3
Да, здесь в этом лесу было этот дуб, с которым мы были согласны поддумал князь Андрей. Докде он поддумал опять князь Андрей, глядя на левую сторону дороге и сам того не знаю, не узнавая, вы любовался этим дубом, которого он искал. Старый дуб весь приображен, раскими в 6-я шатром сочный темный зелень ималел, чуть колыхаясь в лучах вечернего солнца. Никаря в их пальцев, не болячек, ни старого не доверия и горя. [17]	6 секунд	Tiny

Продолжение таблицы.4

1	2	3
Да, здесь в этом лесу был этот дуб, с которым мы были согласны, поддумал князь Андрей. Докде он подумал опять князь Андрей, глядя налевую сторону дороги, и сам того не знаю, не узнаваивали, бывал с этим дубом, которого он искал. Старый дуб, весь преображенный, раскимившийся шатром сочный, тёмный зелень, млел, чуть колыхаясь в лучах вечернего солнца, ни корявых пальцев, ни болячек, ни старого недоверие, и горя. [17]	9 секунд	Base
Да, здесь, в этом лесу, был этот дуб, с которым мы были согласны, подумал князь Андрей. Да где он, подумал опять князь Андрей, глядя на левую сторону дороги и сам того не зная, не узнава его, либовал с этим дубом, которого он искал. Старый дуб, весь преображенный, раскинувшийся шатром сочный, темный зелень млел, чуть колыхаясь в лучах вечернего солнца. Ни корявых пальцев, ни болячек, ни старого недоверия Егоря. [17]	24 секунды	Small
— Да, здесь, в этом лесу, был этот дуб, с которым мы были согласны, — подумал князь Андрей. — Да где он? — подумал опять князь Андрей, глядя на левую сторону дороги, и сам того не зная, не узнавая его, любовался тем дубом, которого он искал. Старый дуб, весь преображенный, раскинувшийся шатром сочной темной зелени, млел, чуть колыхаясь в лучах вечернего солнца, ни корявых пальцев, ни болячек, ни старого недоверия и горя.[17]	98 секунд	Medium

Рассмотрев результаты, можно сделать вывод, что результат модели Medium, хоть и крайне точен, занимает слишком много времени. Также, модель tiny, несмотря на крайне небольшое время обработки аудиофайла, предоставляет слишком неточный результат. Множество слов теряет свой смысл, появляются орфографические ошибки, и несколько слов начинают сливаться в одно. В модели Base же все еще находится множество ошибок и несколько слившихся воедино слов, но его время обработки составляет всего 9 секунд, что довольно быстро. Small в свою очередь, занимает 24 секунды, но количество ошибок значительно сокращено. Оба варианта работы модели подходят для реализации бота, однако, так как текст необходим для обработки моделью BERT, было решено допустить замедление работы программы в угоду точности интерпретации эмоций.

4.4Разработка бота

Первым шагом в разработке бота является создание бота и получение его API_KEY через бот BotFather. Данный бот разработан самими создателями Telegram, и позволяет создать свой бот в пару действий. Далее, необходимо

использовать библиотеку Telebot, которая при указании ключа API позволяет отлавливать все типы данных и обрабатывать те, которые необходимы.

Для начала, необходимо обрабатывать инициализирующую бота команду /start. Для этого создадим bot.message_handler, реагирующий на соответствующую команду. При ее получении, бот высылает приветственное сообщение, описывающее то, какие возможности он имеет.

Далее, необходимо прописать шаблон действия бота, в случае если пользователь отправляет что-либо кроме голосового сообщения. Для этого нам нужен bot.message_handler, реагирующий на все типы данных, кроме типа message. В этом случае, пользователю возвращается ошибка, с просьбой присылать данные необходимого формата.

Для обработки основного типа сообщений необходимо написать несколько функций, которые будут обрабатывать полученный файл голосового сообщения. Функция recognize_text является простым запуском транскрибации модели whisper-small, в качестве входных данных получает путь к файлу, на выход подает текст аудиофайла.

Затем, необходима функция recognize_emote, которая будет запускать процесс обработки текста. Инициализируется обученная нами модель, а также соответствующий токенайзер. Текст конвертируется в токены и attention mask, и они переводятся в формат тензоров и подаются на вход модели. Результаты работы модели представляют собой вероятности эмоций.

Последняя необходимая функция logit_handler, которая обработает полученные результат анализа тона, и выведет эмоцию с наибольшей вероятностью.

Полная обработка входящего голосового сообщения производится посредством bot.message_handler, который вытащит из полученной информации название файла, загрузит его с серверов Telegram, а затем направит в описанные функции по порядку. Получившийся результат работы последней функции logit_handler, а также результат транскрибации, будут отправлены пользователю.

ЗАКЛЮЧЕНИЕ

В результате проведенной работы была достигнута поставленная цель и получены следующие результаты в описанных задачах:

1) в рамках выполненной работы был проведен анализ предметной области, в ходе которого была исследована эффективность различных моделей для решения поставленных задач;
2) был сформирован собственный датасет на основе данных из датасета ru-izard-emotions, подходящий для обучения модели;
3) в ходе проектирования были описаны функциональные требования, построены диаграмма компонентов и диаграмма процессов;
4) на основе собственного датасета было проведено дообучение, в результате которого модель научилась распознавать эмоции в транскрибациях голосовых сообщений;
5) было реализовано использование модели распознавания текста для обработки аудиодорожек.;
6) на основе спроектированной архитектуры был реализован бот Telegram, обрабатывающий голосовые сообщения, и отправляющий результат транскрибации и распознавания эмоций;
7) Готовое приложение было опубликовано в открытом репозитории.

По результатам итогового тестирования работоспособности бота был получен удовлетворительный результат, однако для его улучшения существуют следующие возможности:

1) поиск или составление датасетов больших размерностей, в связи с позитивным влиянием больших объемов данных на сети с архитектурой Transformer;
2) добавление дополнительного алгоритма машинного обучения, анализирующего эмоциональную окраску параметров голоса, и изучение разницы полученных результатов разных алгоритмов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Астахов Д.А., Катаев А.В. Использование современных алгоритмов машинного обучения для задачи распознавания эмоций // Cloud of science. 2018. №4. URL: (дата обращения: 18.02.2024).
2. Тампель И.Б., Карпов А.А. АВТОМАТИЧЕСКОЕ РАСПОЗНАВАНИЕ РЕЧИ. Учебное пособие. – СПб: Университет ИТМО, 2017. – 152 с.
3. Котельников Е. В., Окулов С. М. Обзор подходов для автоматического распознавания эмоций в текстах // Научные итоги года: достижения, проекты, гипотезы. 2012. №2. URL: https://cyberleninka.ru/article/n/obzor-podhodov-dlya-avtomaticheskogo-raspoznavaniya-emotsiy-v-tekstah (дата обращения: 18.02.2024).
4. Н. В. Лукашевич, “Автоматический анализ тональности текстов: проблемы и методы” // Интеллектуальные системы. Теория и приложения, 26:1 (2022), 50–61 URL: https://www.mathnet.ru/rus/ista333 (дата обращения: 18.02.2024).
5. S. Kusal, S. Patil, J. Choudrie, K. Kotecha, D. Vora, and I. Pappas, ‘‘A review on text-based emotion detection—Techniques, applications, datasets, and future directions,’’ 2022 URL: https://doi.org/10.48550/arXiv.2205.03235 (дата обращения 18.02.2024).
6. Полякова А.С., Сидоров М.Ю., Семенкин Е.С. Комбинирование подходов кластеризации и классификации для задачи распознавания эмоций по речи // Сибирский аэрокосмический журнал. 2016. №2. URL: https://cyberleninka.ru/article/n/kombinirovanie-podhodov-klasterizatsii-i-klassifikatsii-dlya-zadachi-raspoznavaniya-emotsiy-po-rechi (дата обращения: 18.02.2024).
7. Anagnostopoulos, CN., Iliou, T. (2010). Towards Emotion Recognition from Speech: Definition, Problems and the Materials of Research. In: Wallace, M., Anagnostopoulos, I.E., Mylonas, P., Bielikova, M. (eds)

Semantics in Adaptive and Personalized Services. Studies in Computational Intelligence, vol 279. Springer, Berlin, Heidelberg. URL: https://doi.org/10.1007/978-3-642-11684-1_8 (дата обращения: 18.02.2024)

8. B. T. Atmaja, K. Shirai and M. Akagi, "Speech Emotion Recognition Using Speech Feature and Word Embedding," 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 2019, pp. 519-523 URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9023098&isnumber=9023008 (дата обращения: 18.02.2024)
9. Sailunaz, K., Dhaliwal, M., Rokne, J. et al. Emotion detection from text and speech: a survey. Soc. Netw. Anal. Min. 8, 28 (2018). URL: https://doi.org/10.1007/s13278-018-0505-2 (дата обращения: 18.02.2024).
10. Hannun, Awni, et al. "Deep speech: Scaling up end-to-end speech recognition." (2014). URL: https://doi.org/10.48550/arXiv.1412.5567 (дата обращения: 18.02.2024).
11. Алёшин Н.А. Рекуррентные нейронные сети. \\ InWorld science: problems and innovations 2021 (pp. 10-12). URL: https://naukaip.ru/wp-content/uploads/2021/04/MK-1050.pdf\ (дата обращения: 18.02.2024).
12. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C. and Sutskever, I., 2023, July. Robust speech recognition via large-scale weak supervision. \\ International Conference on Machine Learning (pp. 28492-28518). URL: https://doi.org/10.48550/arXiv.2212.04356 (дата обращения: 18.02.2024).
13. Ezen-Can, A. (2020). A Comparison of LSTM and BERT for Small Corpus. URL: https://doi.org/10.48550/arXiv.2009.05451 (дата обращения: 18.02.2024).
14. Шевченко, Г. О., and А. Ю. Филиппович. "Анализ тональности текстов на русском языке." Современные наука и образование:

достижения и перспективы развития: сборник материалов XXX-ой международной очно-заочной научно-практической конференции, в 4 т., Том 3, 7 июня, 2023–Москва: Издательство НИЦ «Империя», 2023.–153с.. 2023. URL: https://empirya.ru/f/sbornik_ns-30_tom_3.pdf#page=36 (дата обращения: 18.02.2024).

15. Zmitrovich, D., Abramov, A., Kalmykov, A., Tikhonova, M., Taktasheva, E., Astafurov, D., ... & Fenogenova, A. (2023). A family of pretrained transformer language models for Russian. URL: https://doi.org/10.48550/arXiv.2309.10931 (дата обращения: 22.05. 2024)
16. Студенческая наука: актуальные вопросы, достижения и инновации : Сборник статей XIV Международной научно-практической конференции, Пенза, 17 мая 2024 года. – Пенза: Международный центр научного сотрудничества "Наука и Просвещение", 2024. – 268 с. – С. 29-31. URL: https://naukaip.ru/wp-content/uploads/2024/05/MK-2025.pdf (дата обращения: 22.05.2024)
17. Толстой, Лев. Война и мир. Том 1. Москва: Эксмо, 2020. – 448с.