МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»

Институт (Филиал) № 8 «Компьютерные науки и прикладная математика» **Кафедра** 806

Группа М8О-406Б-19 **Направление подготовки** 01.03.02 «Прикладная математика и информатика»

Профиль Информатика

Квалификация **бакалавр**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

На тему: Определение эмоций персонажей в тексте посредством анализа диалогов с помощью машинного обучения

Автор ВКРБ Гаврилов Максим Сергеевич (_____)
(фамилия, имя, отчество полностью)

Руководитель Лемтюжникова Дарья Владимировна (_____)
(фамилия, имя, отчество полностью)

Консультант _____(_____)
(фамилия, имя, отчество полностью)

Консультант _____(_____)
(фамилия, имя, отчество полностью)

Рецензент _____(_____)
(фамилия, имя, отчество полностью)

К защите допустить

Заведующий кафедрой 806 Крылов Сергей Сергеевич (_____)
(№ каф) (фамилия, имя, отчество полностью)

___ _____ 2023г.

Москва 2023

РЕФЕРАТ

Выпускная квалификационная работа бакалавра состоит из 48 страниц, 11 рисунков, 11 таблиц, 28 использованных источников.

ОБРАБОТКА ЕСТЕТСТВЕННОГО ЯЗЫКА, РАСПОЗНАНИЕ ЭМОЦИЙ В РАЗГОВОРЕ, АНАЛИЗ ТЕКСТА, АНАЛИЗ РАЗГОВОРА, BERT, РАЗРЕШЕНИЕ КОРЕФЕРЕНЦИИ

Цель работы – Получить модель, способную определять эмотивную тональность речи персонажей в диалоге на основе модели эмоций Келлермана-Плутчика.

Для достижения поставленной цели были проведены сравнительные исследования различных методов машинного обучения применительно к задаче определения эмоций в тексте. Основное содержание работы состояло в разработке алгоритма, способного извлекать из текста информацию о действующих лицах, сопоставлять действующие лица и их высказывания, а также алгоритма способного определять эмоциональный окрас высказываний.

Основными результатами работы, полученными в процессе разработки, являются модель, способная определять эмоции персонажей в художественном тексте и программа на ее основе и алгоритм выделения персонажей и их высказываний, который может применяться самостоятельно.

Данные результаты разработки предназначены для интеграции с прочими методами анализа текста для создания системы автоматического анализа художественных текстов.

Применение результатов данной работы позволяет повысить эффективность работы в областях, связанных с необходимостью исследовать большое количество новых художественных текстов н расширить диапазон данных, доступных системам искусственного интеллекта для обучения.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 4
1 ПОСТАНОВКА ЗАДАЧИ 7
1.1 Потребности в разработке темы 7
1.2 Анализ существующих подходов, применимых к решению проблемы определения эмоций персонажей в художественном тексте 9
1.2.1 Методы предобработки текста 11
1.2.2 Методы векторизации текста 11
1.2.3 Методы классификации текста 13
1.2.4 Методы улучшения результатов классификации 16
1.2.5 Методы валидации результатов классификации 17
1.2.6 Анализ примеров решений схожих проблем 19
1.3 Обоснование цели и поставленных задач 21
2 РАЗРАБОТАННОЕ РЕШЕНИЕ 24
2.1 Алгоритм, разработанный для решения поставленных задач 24
2.2 Используемые технологии и методы 28
2.2.1 Составление и анализ наборов обучающих данных 29
2.2.2 Определение методов, используемых для извлечения слов, указывающих на действующих лиц из предложений, содержащих прямую речь 30
2.2.3 Выбор алгоритма на роль слабого ученика 32
2.2.4 Выбор метода построения ансамбля 34
2.3 Описание разработанной программы 35
3 РЕЗУЛЬТАТЫ РАБОТЫ 37
3.1 Результаты работы программы 37
3.2 Технические характеристики программы 40
ЗАКЛЮЧЕНИЕ 44
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 46

ВВЕДЕНИЕ

Актуальность темы данной работы связана потребностью в системах автоматического анализа художественных текстов в областях образования, публицистики и искусственного интеллекта. Такие системы могут значительно повысить эффективность человеческой деятельности, связанной с изучением большого количества новых текстов, облегчить использование художественных текстов для обучения систем искусственного интеллекта и, при использовании в образовательных организациях – повысить интерес молодежи к области искусственного интеллекта. Определение эмоций персонажей в тексте – один из важнейших этапов анализа художественного текста, а в полной мере исследовать эмоции персонажей можно только анализируя их высказывания в диалогах. Однако, на данный момент задача определения эмоций людей по их высказываниям решается только применимо к записям разговоров в социальных сетях и для автоматического сбора мнений в отзывах и комментариях, то есть, для уже размеченного текста. Если создать алгоритм, способный автоматически выделять диалог из текста и размечать его по принадлежности реплик персонажам, то методы, используемые для определения эмоций людей в записях разговоров в социальных сетях можно будет применить и к художественному тексту, что позволит решить задачу определения эмоций персонажей в художественных текстах и приблизится к созданию полноценной системы автоматического анализа художественных текстов.

Таким образом, выполненная работа актуальна и с научно-методической/теоретической, и с практической точек зрения.

Цель работы – Получить модель, способную определять эмотивную тональность речи персонажей в диалоге на основе модели эмоций Келлермана-Плутчика.

Для достижения поставленной цели в работе были поставлены и решены следующие задачи:

- Создать парсер диалога, способный устанавливать соответствия персонаж-реплика. Провести оценку точности его работы, выполнить валидацию результата.
- Для этого найти решение задачи разрешения кореференции в диалоге.
- Создать анализатор эмотивной тональности по Келлерману-Плутчику.
- Для этого сформировать и разметить датасет текстов или предложений и присущей им эмотивной тональности по Келлерману-Плутчику, либо найти существующий датасет такого рода.
- Сравнить различные методы машинного обучения применительно к задаче классификации текстов и предложений по их эмотивной тональности и выбрать наиболее эффективный. При сравнении выполнять кросс-валидацию.
- Изучить уже существующие решения задач исследования эмотивной тональности и разрешения кореференции, по возможности имплементировать их в собственное решение.
- Собрать из реализованных в прошлых пунктах частей модель, способную отслеживать эмотивную тональность реплик определенных персонажей в диалоге.

Работа основывалась на следующих инструментах и методах: язык программирования python и фреймворк для машинного обучения pytorch, библиотека предобученных моделей глубокого обучения transformers, библиотека простых моделей машинного обучения и метрик scikit-learn, библиотека для создания двумерных графиков matplotlib, семейство моделей

глубокого обучения BERT, метод построения ансамблей стекинг, метод векторизации WordPiece.

Основными результатами, полученными в работе, являются:

- Модель, способная определять эмотивную тональность речи персонажей в диалогах в соответствии с моделью Келлермана-Плутчика.
- Программа, созданная на основе полученной модели и способная отслеживать изменения эмоций персонажей с ходом повествования.
- Парсер диалогов, созданный в ходе выполнения работы и способный выделять из художественного текста персонажей и определять их реплики. Он может применяться отдельно от созданной программы, например, для помощи в создании обучающих наборов данных для моделей глубокого обучения.

Результаты работы предназначены для интеграции с другими системами анализа различных аспектов художественного текста для создания системы автоматического анализа художественных текстов. Также они могут быть применены самостоятельно в области образования с целью демонстрации возможности автоматического исследования взаимодействий персонажей в художественных текстах, либо с целью автоматического создания и разметки наборов данных на основе художественных текстов для применения в области глубокого обучения.

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Потребности в разработке темы

Существует потребность в автоматическом анализе художественных произведений. В любой сфере деятельности, где человеку необходимо ознакамливаться с большим количеством художественных текстов, автоматизация анализа текста приведет к значительному повышению эффективности. Особенно актуально это в ситуациях, где рассматриваются недавно написанные тексты, которые еще ни разу не были проанализированы людьми, в частности, в работе редакторов и рецензентов.

В области образования такая технология может как оптимизировать процесс изучения художественных текстов как преподавателем, так и учениками. Впрочем, ввиду того, что большая часть текстов, входящих в общеобразовательные программы, уже давно изучены и проанализированы литературоведами, актуальность такого применении технологии автоматического анализа текстов невысока. С другой стороны, возможность быстро и надежно узнавать основную информацию о тексте даст возможность исследовать большое количество менее известных произведений и позволит ввести более исследовательский подход к изучению литературы в школах. К тому же, применение этой технологии в школьном образовании может увеличить интерес молодежи к искусственному интеллекту.

Технология автоматического анализа текста также крайне актуальна в задаче создания систем искусственного интеллекта, осуществляющих взаимодействие с человеком на естественном языке. Они нуждаются в обучении на художественных текстах. Сейчас такие системы обучаются на массивах сообщений из социальных сетей, однако такие массивы содержат лишь специфические примеры межличностного взаимодействия, и не в полной мере охватывают разнообразие человеческого общения. Художественные произведения могут дополнить обучающие данные. Обучение на массивах художественных текстов уже используется при создании некоторых интеллектуальных систем, в частности, голосовой

помощник "Алиса" был предварительно обучен на массиве художественной литературы, и лишь затем – на сообщениях из социальных сетей. Такой метод обучения может быть улучшен, если понимать структуру подаваемого на обучение текста, выделять из него важные части, или фокусировать внимание на определенных аспектах взаимодействия людей. Размечать тексты вручную невозможно ввиду того, что для обучения современных интеллектуальных систем необходимо огромное количество данных. Потому и необходимо иметь систему, способную автоматически решать эти задачи.

Для решения комплексной задачи автоматического анализа художественных текстов необходимо решить ряд подзадач, в числе которых определение эмоций персонажей:

1) Определение главных и второстепенных персонажей.
2) Определение основных характеристик каждого персонажа.
3) Определение основных действий каждого персонажа.
4) Определение основного сюжета произведения.
5) Определение основных эмоций персонажей и их изменения в ходе сюжета.
6) Определение эмоций, которые возникают между персонажами.
7) Определение основных конфликтов в произведении.
8) Определение ключевых действий персонажей.
9) Определение характеров взаимодействия персонажей.
10) Определение возможных трактовок действий и слов персонажей, разрешение неоднозначности смыслов.

В данной работе рассмотрено решение задачи определения эмоций, возникающих между персонажами, их изменения в ходе сюжета, а также эмоций, возникающих между персонажами. Для определения эмоций, возникающих между персонажами, необходимо рассматривать диалоги, так как именно в диалогах такие эмоции чаще всего проявляются. К тому же, изучение реплик персонажа – гарантия того, что выявленные эмоции принадлежат самому персонажу, а не являются частью эмоционального

настроя авторского текста. Помимо этого, нередко эмоции, выраженные в диалогах, никак не комментируются в авторской речи, что означает, что их необходимо определять, анализируя реплики, а не ища ключевые слова в речи автора.

Итак, мы убедились, что для выявления эмоций персонажей необходимо анализировать диалоги. Анализ диалогов, в свою очередь, требует решить задачи определения персонажей, выделения из текста высказываний и правильного присвоения высказываний персонажам, т.е. выявления связей "высказывание-персонаж".

1.2 Анализ существующих подходов, применимых к решению проблемы определения эмоций персонажей в художественном тексте

Для создания систем автоматического анализа художественного текста необходимо найти способ определять эмоции персонажей в тексте. Для определения эмоций персонажей в тексте, в свою очередь, необходимо исследовать диалоги.

Проблемы, которые необходимо решить:

- Извлечение диалога из текста.
- Анализ диалогов.

Для получения размеченного диалога, необходимо решить задачу выделения диалога из текста. Проблема выделения диалога из текста состоит из нескольких частей, некоторые из которых тривиальны, а некоторые требуют пристального внимания при решении. Задачи, которые необходимо решить для решения данной проблемы, это:

- отыскание предложений, содержащих диалог,
- отыскание действующих лиц,
- установление связей между действующими лицами и высказываниями в диалоге.

Основная сложность при решении этих задач заключается в необходимости разрешения кореференции между различными действующими лицами.

Задача разрешения кореференции заключается в определении, какие упоминания в тексте обозначают один и тот же объект действительности (референт) [1]. Для решения этой задачи применяются как методы, основанные на правилах, так и методы на основе глубокого обучения, впрочем, в задаче разрешения кореференции в текстах на русском языке методы на основе правил показывают лучшие результаты [1].

Задача распознания эмоций в тексте является, по своей сути, задачей классификации текста. Существует множество подходов к задаче классификации текста, по большей части их можно разделить на подходы на основе классического машинного обучения и подходы на основе языковых моделей [2] – [6].

Методы на основе классического машинного обучения имеют следующие особенности:

- требуют большого количества предобработки,
- просты вычислительно и структурно,
- рассматривают каждое слово отдельно.

Методы на основе алгоритмов глубокого обучения имеют следующие особенности:

- требуют малого количества предобработки,
- требуют больших вычислительных мощностей,
- учитывают контекст.

Оба подхода стоит рассмотреть, как потенциальные решения задачи классификации текста, и определить, какой из них даст более точные результаты.

Задача классификации текста решается в три этапа:

1) Предобработка.
2) Векторизация.
3) Классификация.

1.2.1 Методы предобработки текста

На первом этапе текст нормализуется и очищается от стоп-слов. Этот этап необходим только если применяются методы классического машинного обучения.

При использовании методов классического машинного обучения для классификации текста, необходимо проводить его нормализацию. Лемматизация – это частный случай нормализации, который заключается в приведения слова к лемме, то есть к его словарной форме. Это необходимо из-за того, что в противном случае модель будет воспринимать одно и то же слово в разных падежах, числах и родах, как разные слова, не имеющие между собой ничего общего.

Стоп-слова – это слова, которые не несут особого смысла и употребляются очень часто. Например, это числительные, местоимения и предлоги. При классификации текста частотными методами, стоп слова вносят ненужный шум и уводят фокус с действительно имеющих значение слов, а также увеличивают объем данных, не неся при этом полезной информации. Поэтому, при использовании методов классического машинного обучения для классификации текстов, стоп-слова необходимо убирать.

1.2.2 Методы векторизации текста

На втором этапе выполняется векторизация. Большинство моделей машинного обучения могут работать только с числами (тензорами). По этой причине, для того чтобы применять такие модели к тексту, его необходимо предварительно векторизовать, то есть представить в виде вектора (последовательности чисел). Существует множество различных способов векторизации [7]. Вот некоторые из них:

One hot encoding

Каждому слову из словаря в N слов соответствует вектор длины N, с единицей на позиции n, где n - номер слова в словаре, и нулями на всех остальных позициях. Это самый простой метод векторизации и, как следствие, имеет массу недостатков, в числе которых очень большой размер векторов для

больших словарей, разреженность вектора и полная потеря семантической близости слов при векторизации.

Bag of words

Каждому предложению ставится в соответствие вектор длины N, где N - длина словаря. Значение на позиции n в векторе равно тому, сколько раз слово с номером n встречалось в предложении. Данный метод позволяет добиться одинаковой длины векторов для предложений разной длины, а также решает проблему разреженности векторов. Впрочем, проблема потери семантической близости слов остается неразрешенной.

Bag of n-grams

Этот метод, вместо того, чтобы подсчитывать частоты отдельных слов, ведет подсчет частот групп из n токенов. Такие группы называются n-граммами. Каждому предложению ставится в соответствие вектор длины N, где N – число n-грам в словаре. «Bag of words» можно считать частным случаем метода bag of n-grams, где n = 1. Этот метод позволяет учитывать семантическую близость слов, однако он страдает от высокой разреженности векторов.

Word to vec [8]

Каждому слову ставится в соответствие вектор размерности n. Вектора строятся с учетом значений слов, т.е. чем ближе значения слов, тем меньше угол между соответствующими векторами. Этот метод основан на нейронных сетях. Он принимает на вход текстовый корпус и формирует векторные представления на основе контекстной близости слов, то есть, чем чаще слова встречаются близко друг к другу в тексте, тем более близкими по смыслу они считаются.

Этот метод использует continuous bag of words и skip-gram для обучения. continuous bag of words – это метод предсказания слова по его контексту с помощью нейронных сетей. skip-gram – это метод предсказания слов контекста по одному слову.

1.2.3 Методы классификации текста

На третьем этапе выполняется классификация векторизованного текста. Для этого могут применяться различные модели машинного обучения, от простых частотных алгоритмов, до моделей глубокого обучения. Рассмотрим наиболее подходящие для решения этой задачи модели.

Наивный байесовский классификатор [9].

Это простой вероятностный классификатор, основанный на наивном предположении о независимости слов в классифицируемом тексте. Он применяет теорему Байеса для определения вероятности, с которой текст T относится к классу C. Теорема Байеса записывается так:

$$p(C|x) = \frac{p(C)p(x|C)}{p(x)}, \quad (1)$$

где $p(C|x)$ – вероятность того, что предложение, в котором есть слово x, относится к классу C,

$p(C)$ – вероятность того, что случайное предложение будет относится к классу C,

$p(x|C)$ – вероятность того, что в предложении класса C встретится слово x,

$p(x)$ – вероятность того, что случайное слово будет словом x.

Ввиду того, что знаменатель не зависит от C, его можно не учитывать. Если сложить вероятности $p(C|x)$ для всех слов x в предложении, то получится вероятность того, что предложение относится к классу C. Вероятности $p(C)$ и $p(x|C)$ рассчитываются при обучении на выборке с размеченными классами, как частота, с которой встречается класс C и частота, с которой в предложениях класса C встречается слово x.

Этот алгоритм обучается крайне быстро и не требует большого количества обучающих данных, но не является достаточно точным, ввиду того, что слова в предложении не являются независимыми признаками. Тем не

менее его можно рассматривать, как потенциальный алгоритм для решения поставленной задачи.

Наивный словарный метод.

Для определения эмоций в тексте можно использовать следующий метод: для каждого слова в словаре определяется, какие эмоции это слово выражает. Эта информация записывается в бинарном векторе размерностью N, где N – число эмоций, которые необходимо распознавать. При классификации предложения, вектора, соответствующие словам в нем, складываются и позиция с наибольшим значением в результирующем векторе считается номером эмоции, наиболее выраженной в предложении.

Этот подход, по своей сути, является предобученным байесовским классификатором с дополнительным предположением о равновероятности классов, а значит, он еще более примитивен. Его стоит применять только если в наличии имеется готовый размеченный словарь, а набор данных недостаточно объемен для качественного обучения других алгоритмов.

Рекуррентные нейронные сети [10].

Это нейронные сети, предназначенные для обработки последовательных данных, в том числе, естественного языка. Они имеют собственную память (внутреннее состояние), в которой хранится информация об обработанных ранее входных данных. При обработке естественного языка рекуррентная нейронная сеть принимает на вход по одному слову за раз и хранит информацию о полученных словах во внутреннем состоянии.

Основным недостатком рекуррентных нейронных сетей является невозможность параллелизировать их исполнение, что привело к почти полному их вытеснению из области обработки естественного языка моделями на основе трансформеров.

Трансформер [11] – [13].

Это модель глубокого обучения, которая использует механизм внутреннего внимания (self-attention). Слой внутреннего внимания позволяет

определить, насколько нужно учитывать другие слова предложения (последовательности) при кодировании рассматриваемого слова. На вход трансформеру-энкодеру поступает последовательность векторизованных слов, после чего для каждого из них вычисляется вектор внутреннего внимания, с учетом которого слова кодируются дальше. Результатом работы такой модели является некоторое векторное представление предложения, содержащее информацию о контексте его элементов.

Основным преимуществом трансформеров по сравнению рекуррентными нейронными сетями является их высокая эффективность при параллельном выполнении. Трансформеры используют только слои внутреннего внимания и прямого распространения. После прохождения слоя внутреннего внимания каждое слово обрабатывается независимо от других.

BERT [14],[15] (bidirectional encoder representation transformers) – это семейство языковых моделей. Они основаны на технологии трансформеров, а именно, используют двунаправленные трансформеры-энкодеры.

Языковая модель – это распределение вероятностей над последовательностями слов. Получая на вход последовательность слов, языковая модель должна вычислить вероятность, с которой такая последовательность встретится в тексте, созданном человеком.

Модели семейства BERT предобучаются на двух типах задач – предсказание слов по контексту (моделирование языка) и определение того, является ли вторая фраза логичным предложением первой.

Предобученная таким образом модель может быть использована для решения широкого спектра задач обработки естественного языка, благодаря тому, что обученные энкодеры могут создавать компактные представления текстов, содержащие актуальную информацию об использованных в них словах, их взаимосвязи и смыслах. Для такого использования модели BERT, необходимо создать поверх ее выхода дополнительный линейный слой и обучить его, не изменяя весов самой сети. Предобученный bert будет выделять

из текста смысловую информацию, а линейный слой – используя ее, решать поставленную задачу.

1.2.4 Методы улучшения результатов классификации

Для улучшения точности классификации могут быть использованы ансамбли. Для снижения переобучения можно применить аугментацию и регуляризацию. В качестве основного метода получения качественных моделей глубокого обучения можно использовать transfer learning.

Ансамбль [16] – это метод машинного обучения, в котором результат применения нескольких слабых моделей, решающих одну и ту же задачу, но обученных на разных наборах данных, объединяют для получения более точного результата, чем результат работы каждой из этих моделей по отдельности. Слабыми моделями называют обычные алгоритмы машинного обучения – составные части ансамбля, которые объединяются для получения более точного результата. Существует несколько видов ансамблей. Применительно к задаче классификации текста, будут рассмотрены стекинг и бэгинг.

Стекинг – это тип ансамблей, в которых могут использоваться разнородные слабые алгоритмы. Результат работы всех слабых алгоритмов поедается на вход мета-алгоритму, который вырабатывает финальный прогноз.

Бэгинг – это тип ансамблей, в которых могут использоваться только однотипные слабые алгоритмы. После получения результата, применяется голосования – если большинство моделей предсказали данный класс, то он и считается финальным прогнозом. Также может применяться мягкое голосование, тогда каждая модель помимо предсказания предоставляет некое число – уверенность в этом предсказании, которое учитывается при определении финального прогноза.

Для обучения слабых алгоритмов, необходимо разбивать обучающую выборку на K подвыборок, где K – число алгоритмов в ансамбле. Для этого используется алгоритм бутстрап. Этот алгоритм заключается в извлечении из

выборки размером N некого числа M объектов с возвращением, т.е. вероятность извлечь объект x всегда равна $\frac{1}{N}$. Сформированные таким образом подвыборки размера M используются для обучения слабых алгоритмов.

Аугментация [17] – это создание дополнительных данных для обучения на основе существующих. Это крайне эффективных инструмент для повышения точности обучения, который активно применяется в задачах классификации изображений, однако, в задаче классификации текста, ее применение невозможно ввиду того, что создание новых осмысленных текстов на основе имеющихся – крайне нетривиальная задача.

Регуляризация – это метод ограничения сложности модели для предотвращения переобучения. В данной работе рассматривается применение регуляризации путем исключения (dropout)

Dropout [18] – это метод предотвращения взаимоадаптации нейронов на этапе обучения путем отключения заданного процента отдельных нейронов. При использовании данного метода любой нейрон ети может быть выключен на этапе обучения с заданной вероятностью p. Использование метода dropout существенно снижает переобучение и позволяет улучшить точность классификации новых объектов.

Transfer Learning [19] – это метод ускорения и улучшения качества обучения, основанный на применение нейронной сети, обученной для решения определенной задачи для решения схожей задачи. Это возможно благодаря тому, что для решения схожих задач нейронная сеть должна выделять из данных схожие признаки и особенности. Если она уже обучена это делать, то достаточно лишь переобучить верхние слои сети, чтобы применить ее для решения другой задачи. Это особенно полезно, когда обучающих данных недостаточно, чтобы качественно обучить сеть с нуля.

1.2.5 Методы валидации результатов классификации

Для валидации результатов работы рассматриваемых алгоритмов можно использовать следующие метрики:

Оценка accuracy [20] – это отношение правильно классифицированных элементов тестового набора данных к общему количеству элементов тестового набора данных. Эта метрика хороша только тогда, когда в тестовом наборе данных равное количество элементов каждого класса.

Оценка top-n-accuracy [20] – это версия оценки accuracy, при расчете которой элемент считается правильно классифицированным, если его истинный класс попал в n наиболее вероятных классов, к которым принадлежит данных элемент согласно предсказанию модели. Эта метрика полезна в случаях, когда модель не может выполнять предсказания с высокой точностью, в том числе в задачах классификации текста.

Оценки precision, recall и F1 [20],[21] – это метрики, которые позволяют преодолеть недостаток оценки accuracy. Они считаются по отдельности для каждого класса, и определяются так:

$$precision = \frac{TP}{TP + FP}, \tag{2}$$

$$recall = \frac{TP}{TP + FN}, \tag{3}$$

где TP – Это число элементов, относящихся к данному классу, которые были классифицированы, как относящиеся к данному классу,

FP – Это число элементов, не относящихся к данному классу, которые были классифицированы, как относящиеся к данному классу,

FN – Это число элементов, относящихся к данному классу, которые были классифицированы, как не относящиеся к данному классу.

Таким образом, precision демонстрирует способность алгоритма отличать данный класс от остальных, а recall – демонстрирует способность алгоритма обнаруживать данный класс вообще.

Оценка F1 – это способ объединить precision и recall в одной метрики, в частности, это среднее гармоническое этих двух величин.

Матрица ошибок [20] – это способ визуально отобразить результат классификации набора данных алгоритмом. Каждому столбцу матрицы соответствует класс, который определяет алгоритм, а каждой строке – класс, к которому действительно относится объект. Если объект относится к классу N, а модель определила его, как объект класса M, то значение матрицы ошибок в ячейке M строки N инкрементируется на единицу.

Также при оценивании метрик того или иного алгоритма машинного обучения необходимо проводить кросс-валидацию. *Кросс-валидация* [22] – это метод оценивания обобщающей способности модели. Она подразумевает разбиение набора данных на тренировочный и валидационный. Модель обучается на тренировочном наборе данных, а затем ее точность проверяется на валидационном. Таким образом проверяется поведение модели на новых данных. Процесс разбиение-обучение-проверка проводится несколько раз, чтобы получить среднюю точность модели на незнакомых данных, при этом разбиение каждый раз выполняется по-разному. Различные методы кросс-валидации используют различные способы разбиения.

1.2.6 Анализ примеров решений схожих проблем

В работе [4] рассматривается решение задачи распознания эмоций применимо к социальным роботам. Одна из задач авторов работы – создание модели, которая позволит роботу определять эмоции в речи человека. Для этого используется ансамбль типа стекинг из трансформеров с многослойными классифицирующими головами, содержащими по два скрытых слоя в 500 и 300 нейронов. В качестве метаалгоритма применяется многослойный персептрон со скрытым слоем в 10 нейронов. В качестве модели эмоций используется модифицированная модель Келлермана-Плутчика на 11 эмоций и с отсутствием эмоций, как отдельным классом. Такая модель достигла точности определения эмоций 0.54. Результаты этой работы нельзя использовать для решения поставленной задачи ввиду того, что в ней рассматривалась классификация уже размеченных высказываний человека. Представленная в работе архитектура может быть применена для создания

классификатора эмоций, однако не может самостоятельно решить проблему определения эмоций персонажей художественного текста. Также стоит учитывать, что модель, созданную авторами этой работы, не получится в том же виде применить для определения эмоций в предложениях на русском языке, ввиду того, что использованные ими трансформеры были предобучены на корпусе английских текстов.

В работе [3] рассматриваются методы определения эмоций в размеченном диалоге между двумя людьми. Основной упор делается на моделях на основе рекуррентных нейронных сетей, в том числе классификаторах, запоминающих историю диалога. Отмечается, что такие модели плохо обрабатывают изменение эмоций в ходе диалога. Очевидно, что такие методы будет затруднительно применить к диалогу, заключенному в художественный текст, даже если он будет предварительно оттуда извлечен.

Можно заключить, что существующие решения опираются на уже размеченный диалог. К тому же, чаще всего они используют не только текст, но и записи голоса. Ни того, ни другого нет в художественных текстах, так что необходимо разработать свое решение. Имея размеченный диалог, можно использовать архитектуры из приведенных статей для решения задачи определения эмоций. При этом, использовать сами готовые решения невозможно, ввиду того, что они обучены для классификации англоязычных предложений.

В работе [23] описывается метод разрешения кореференции между именами собственными, относящимися к персоналиям. Представленный алгоритм работает в два этапа: вначале выполняется извлечение упоминаний, а затем – аггломеративная кластеризация.

Готовые решения могут не идеально подходить для решения задачи разрешения корефернции между персонажами, произносящими реплики в диалогах, ввиду специфичности референтов, которые представляют из себя исключительно сущности, способные вести диалог, в то время, как большая часть алгоритмов разрешения кореференции предназначена для решения этой

задачи в общем виде, то есть между всеми потенциальными референтами в тексте.

1.3 Обоснование цели и поставленных задач

Для того, чтобы решать задачу определения эмоций в тексте, необходимо решить, какие эмоции нужно определять. Для этого следует выбрать модель эмоций, которая ляжет в основу программы. В качестве такой модели была выбрана модель эмоций Келлермана-Плутчика, ввиду того, что для решения поставленной задачи были предоставлены два набора данных, размеченных в соответствии с этой моделью.

Модель эмоций Келлермана-Плутчика [24] – это пространственная модель эмоций, то есть эмоции в ней расположены в двумерном пространстве с типом эмоции и ее выраженностью в качестве координат. Графическое представление модели изображено на рисунке 1. Она содержит 8 биполярных первичных эмоций: радость, печаль, гнев, страх, доверие, отвращение, удивление, настороженность. С учетом различных степеней выраженности, она определяет 24 эмоции, однако небольшой размер набора данных не позволяет проводить классификацию по всем 24 эмоциям, так что в данной работе будут рассматриваться 8 первичных эмоций по Келлерману-Плутчику. Это следующие эмоции:

- радость (восторг, радость, безмятежность),
- восхищение (восхищение, доверие, принятие),
- злость (гнев, злость, досада),
- грусть (горе, грусть, печаль),
- удивление (изумление, удивление, возбуждение),
- отвращение (отвращение, неудовольствие, скука),
- ожидание (настороженность, ожидание, интерес),
- страх (ужас, страх, тревога).

Таким образом, цель работы – получить модель, способную определять эмотивную тональность речи персонажей в диалоге на основе модели эмоций Келлермана-Плутчика.

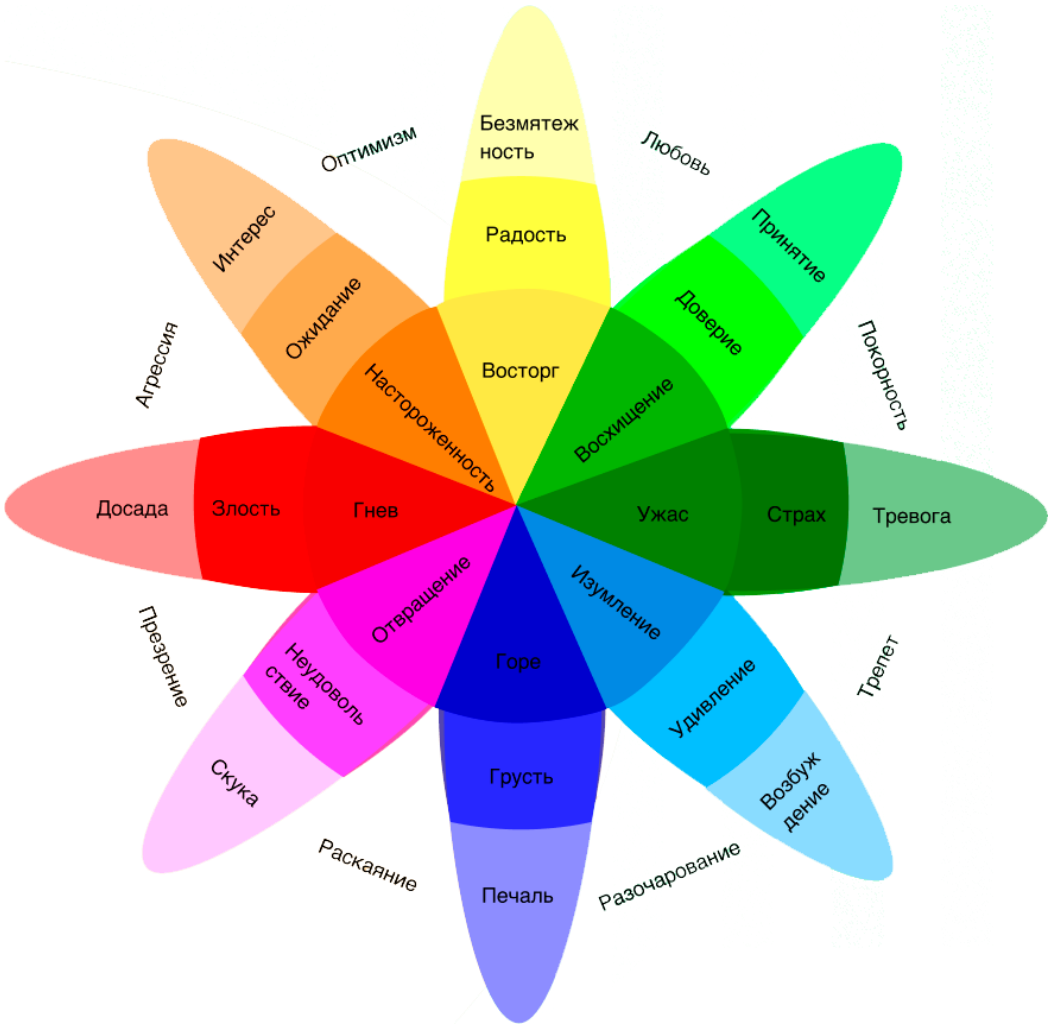

Рисунок 1 – Графическое представление модели эмоций Келлермана-Плутчика

Для достижения поставленной цели необходимо определить, какой подход к классификации текста наиболее применим для решения задачи определения эмоций, то есть, надо провести сравнение точности различных методов машинного обучения применительно к задаче определения эмоций. При сравнении необходимо использовать кросс-валидацию. После этого на основе наиболее применимого подхода необходимо создать модель, способную определять эмоции в отдельных предложениях.

Для сравнения различных методов машинного обучения, необходимо использовать размеченный в соответствии с моделью эмоций Келлермана-Плутчика набор данных. Если имеющегося набора данных будет недостаточно, необходимо самостоятельно его расширить.

Для того чтобы применять полеченную модель, способную определять эмоции в отдельных предложениях к диалогам персонажей в художественных текстах, необходимо создать парсер диалогов, способный устанавливать соответствия персонаж-реплика. Необходимо оценку точности его работы и выполнить валидацию результатов.

Для создания такого парсера необходимо найти решения задачи разрешения кореференции в диалоге.

При этом необходимо проверить применимость уже существующих методов определения эмоций к решению задачи определения эмотивной тональности речи персонажей в диалоге и, по возможности, использовать из.

Наконец, необходимо собрать из парсера диалогов и готовой модели, определяющей эмоции в отдельных предложениях, полную модель, способную отслеживать эмотивную тональность реплик определенных персонажей в диалоге.

2 РАЗРАБОТАННОЕ РЕШЕНИЕ

2.1 Алгоритм, разработанный для решения поставленных задач

Для решения поставленной задачи был разработан следующий алгоритм:

Сначала при помощи парсера диалогов из текста извлекаются реплики персонажей. Определяются действующие лица (персонажи) и принадлежность реплик тому или иному персонажу.

Затем список реплик подается на вход анализатору эмоций. Устанавливается эмотивная тональность каждой реплики.

Полученный список персонажей, соответствующих им реплик и соответствующих репликам эмоций используется для определения эмоций персонажей в рассматриваемом художественном тексте. Диаграмма, визуализирующая данный алгоритм представлена на рисунке 2.

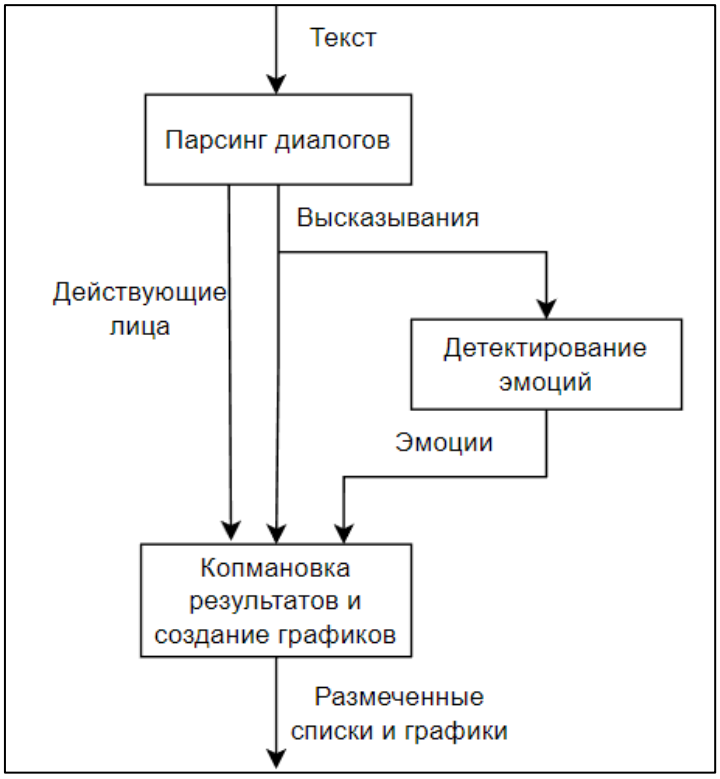

Рисунок 2 – Алгоритм работы программы

Парсер диалогов работает следующим образом. На вход поступает текст.

1) Из текста выделяются все предложения, содержащие прямую речь (реплики).
2) Из авторской части каждого такого предложения выделяются слова, указывающие на действующих лиц.
3) Каждому слову, указывающему на действующее лицо, ставится в соответствие объект класса референт.
4) Каждой реплике ставится в соответствие референт, произнесший ее.
5) Производится проверка синонимичности референтов. Синонимичные референты сливаются в один.

На первом этапе алгоритма каждому предложению, содержащему прямую речь, ставится в соответствие объект класса высказывание. Созданные объекты помещаются в список предложений, содержащих прямую речь.

Каждый объект класса высказывание хранит в себе список слов, которыми в авторской речи данного высказывания обозначалось действующее лицо, произнесшее это высказывание и ссылку на референт, который произнес это высказывание. Список слов, указывающих на произнесшее высказывание действующее лицо определяется на втором этапе алгоритма.

Действующее лицо, произнесшее прямую речь чаще всего является подлежщим в авторской речи, что следует за ней. Чтобы найти подлежащее в предложении нужно использовать синтаксический парсер. Найденные слова помещаются в список слов, указывающих на произнесшее высказывание действующее лицо, а также в общий список слов, указывающих на действующих лиц.

Полученный на втором этапе алгоритма общий список используется для построения списка референтов на третьем этапе. Объект класса референт ставится в соответствие слову, указывающему на действующее лицо только если оно встречается в списке два и более раз.

Каждый объект класса референт хранит в себе список слов, которыми его обозначали в тексте (имена референта) и список высказываний, которые

ему принадлежат. Объект класса референт создается на основе одного слова и сразу после создания имеет в списке имен лишь слово, на основе которого он был создан.

Стоит отметить, что в одном высказывании авторская речь может содержать несколько разных слов, обозначающих действующие лица, при том что все эти слова будут относиться к одному и тому же действующему лицу.

После создания объектов класса референт начинается итеративный процесс установления связей между высказываниями и референтами. При рассмотрении высказывания список извлеченных из него слов, указывающих на действующих лиц, сопоставляется со списками имен референтов. Если найдено пересечение со списком одного из референтов, то считается, что высказывание было произнесено данным референтом. Слова, указывающие на действующее лицо в рассматриваемом высказывании, добавляются к списку имен референта.

В случае, если в списке слов, указывающих на действующих лиц, выделенных из рассматриваемого высказывания, есть местоимения, в тексте выполняется поиск соответствующих им имен, посредством обратного движения по тексту до достижения имени, входящего в общий список слов, указывающих на действующих лиц, и находящегося в именительном падеже.

В случае, если рассматриваемое высказывание не имеет авторской речи, выполняется проверка предыдущего предложения в тексте. Если оно оканчивается двоеточием, то оно содержит авторскую речь, относящуюся к данному выказыванию.

В случае, если из авторской речи рассматриваемого высказывания не удалось выделить слова, указывающие на действующих лиц, подразумевается, что высказывания принадлежат двум персонажам, ведущим диалог, и действующие лица чередуются.

После завершения процесса установления связей между высказываниями и референтами выполняется слияние синонимичных референтов. Референты считаются синонимичными, если между списками их

имен имеются пересечения. Это значит, что они описывают одну и ту же сущность. В таком случае их списки имен и списки высказываний объединяются и остается лишь один объект класса референт.

Результатом работы парсера диалогов является список референтов и список высказываний. Между референтами и высказываниями должна быть установлена связь «один-ко-многим». Каждый референт представляет собой персонажа, произносившего реплики в тексте. Диаграмма, визуализирующая основные этапы работы парсера диалогов представлена на рисунке 3.

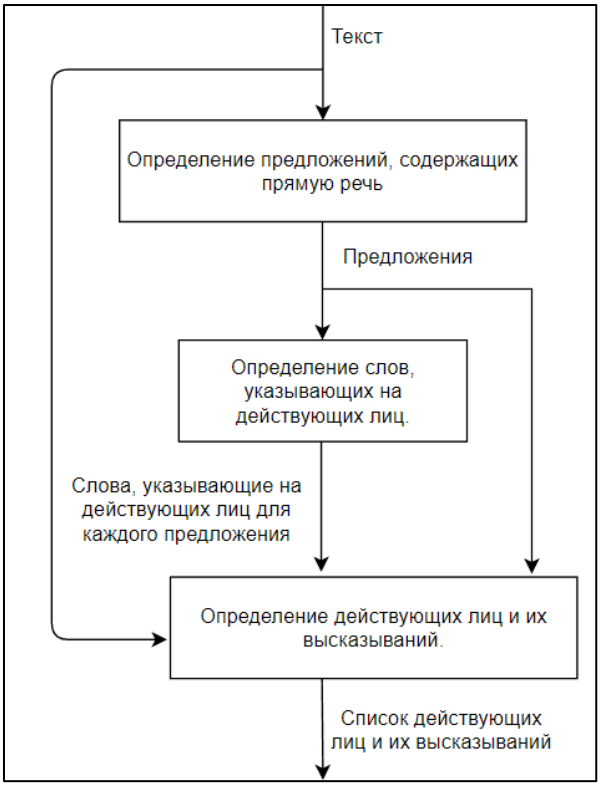

Рисунок 3 – Алгоритм работы парсера диалогов

Полученный список высказываний нужно обработать анализатором эмоций, чтобы получить эмотивную тональность реплик и, соответственно, эмоции персонажей, что их произносят.

Анализатор эмоций работает следующим образом. На вход поступает реплика персонажа и авторская речь. Проводится ее анализ несколькими слабыми алгоритмами. Это могут быть как простые алгоритмы машинного

обучения, так и классификаторы, построенные с использованием глубокого обучения. Затем на основе их предсказаний происходит формирование финального результата.

Финальный результат может формироваться как путем стекинга, так и путем бегинга, выбор типа ансамбля зависит от результатов сравнения данных методов. Выбор слабых алгоритмов также должен быть продиктован результатами сравнения различных методов машинного обучения применимо к задаче определения эмоций.

Итоговый программный продукт работает следующим образом: на вход поступает текст. Парсер диалогов производит извлечение высказываний и действующих лиц из текста. Затем, анализатор эмоций устанавливает эмотивную тональность каждого высказывания. На выход подается список высказываний, для каждого из которых известны выраженные в нем эмоции и список действующих лиц, для каждого из которых известно, какие высказывания ему принадлежат. Полученная информация отображается визуально, в виде графиков зависимости выражаемых эмоций от времени для каждого из персонажей, либо сохраняется в файл.

2.2 Используемые технологии и методы

Для решения задач в области искусственного интеллекта и машинного обучения лучше всего подходит язык программирования python. Платформа Jupyter была выбрана из-за своей интерактивности, что позволяет в полной мере пользоваться преимуществами интерпретируемости языка python. Для реализации методов машинного обучения был выбран фреймворк pytorch, ввиду того, что на нем легче работать с GPU, чем при использовании аналогов. Для классификации предложений по эмотивной тональности будет применяться ансамбль. Тип моделей, используемых в качестве слабых учеников и метод ансамблирования должен быть выбран по результатам экспериментов.

2.2.1 Составление и анализ наборов обучающих данных

Для обучения классификаторов текста необходимо найти или создать обучающий набор данных. Из готовых наборов данных наиболее подходящим для решения поставленной задачи является датасет cedr [25], содержащий 9410 предложений, размеченных на 6 классов, включающих 5 эмоций: удовольствие, грусть, печаль, удивление, страх, злость и отсутствие эмоций, как отдельный класс. Количества предложений каждого из классов в датасете cedr представлены в таблице 1. Сам по себе он недостаточен для обучения классификатора, определяющего эмоции в соответствии с моделью Келлермана-Плутчика, ввиду отсутствия в нем трех из восьми эмоций модели, однако, он может быть использован для предобучения классификатора.

Также путем слияния трех малых датасетов, был составлен набор данных, размеченный в соответствии с моделью Келлермана-Плутчика, содержащий всего 712 предложений из первых глав романа «Мастер и Маргарита». Столь малое количество примеров, а также низкое качество разметки и наличие в наборе как высказываний прямой речи, так и обычных предложений, делают его едва ли пригодным для обучения классификатора.

По этой причине был составлен новый набор данных, содержащий 900 предложений из первых 10 глав романа «Мастер и Маргарита», содержащий прямую речь, и размеченных на восемь базовых эмоций по Келлерману-Плутчику с отсутствием эмоций, как девятый класс. Количества предложений каждого из классов в датасете cedr представлены в таблице 2.

Таблица 1 – Количество элементов каждого из классов в датасете cedr

	Удовольствие	Печаль	Страх	Удивление	Злость	Нет эмоций
Количество предложений	1569	1417	589	607	411	3043

Таблица 2 – Количество элементов каждого из классов в датасете, составленном из 10 первых глав романа «Мастер и Маргарита»

	Радость	Восхищение	Злость	Грусть	Удивление	Ожидание	Отвращение	Страх	Нет эмоций
Количество предложений	83	45	114	46	73	88	46	70	308

Для валидации результатов работы парсера диалогов были составлены датасеты на основе первых двух глав романа «Мастер и Маргарита». В каждом предложении, содержащем прямую речь, были установлены слова, указывающие на действующих лиц, был сформирован список действующих лиц, и для каждого из них были установлены слова, которыми автор их называет. Метрики на датасете на основе первой главы (датасет 1) использовались для настройки парсера, метрики на датасете на основе второй главы (датасет 2) использовались только для валидации результатов.

2.2.2 Определение методов, используемых для извлечения слов, указывающих на действующих лиц из предложений, содержащих прямую речь

На втором этапе работы парсера диалогов происходит извлечение слов, указывающих на действующих лиц из предложений, содержащих прямую речь. Сравнивались следующие методы извлечения слов, указывающих на действующих лиц из авторской речи:

1) Определение подлежащего с помощью синтаксического парсинга (основной метод).
2) Определение имен собственных, находящихся в именительном падеже.
3) Определение имен с подчиненным глаголом, находящихся в именительном падеже.
4) Определение любых имен, находящихся в именительном падеже.

Для синтаксического парсинга использовалась бибилиотека Natasha, для определения частей речи – rupostager.

Точность каждого из методов по-отдельности на текстах первой и второй главы романа «Мастер и Маргарита» представлена в таблице 3. Точность комбинаций из этих методов на тех же текстах представлена в таблице 4. При использовании комбинаций методов, они применялись к текстам в той последовательности, в которой представлены в списке выше. Если один из методов находил слово, указывающее на действующее лицо, то следующие методы не применялись.

Таблица 3 – Точности каждого из методов извлечения слов, указывающих на действующих лиц из авторской речи

Используемый метод	Точность	
	Датасет 1	Датасет 2
Подлежащее по синтаксическому парсеру	0.715	0.767
Имя собственное	0.482	0.630
Имя с подчиненным глаголом.	0.919	0.848
Любое имя	0.88	0.832

Таблица 4 – Точности комбинаций из методов извлечения слов, указывающих на действующих лиц из авторской речи

Используемая комбинация	Точность	
	Датасет 1	Датасет 2
1, 3	0.944	0.894
3, 4	0.936	0.851
1, 3, 4	0.962	0.903
1, 2, 3, 4	0.974	0.921

Было установлено, что наилучшую точность показывает комбинация из всех четырех методов, и среди представленных методов нету лишних, т.е. таких, которые можно было бы исключить без потери точности.

2.2.3 Выбор алгоритма на роль слабого ученика

В качестве кандидатов на роль слабого алгоритма в ансамбле, осуществляющем определение эмоций выступали:

- Наивный байесовский классификатор.
- Наивный словарный классификатор.
- Классификатор на основе модели DeepPavlov RuBERT-base-cased.
- Классификатор на основе модели DeepPavlov RuBERT-base-cased, дополнительно предобученной на датасете cedr.
- Классификатор на основе модели cointegrated rubert-tiny 2, дополнительно предобученной на датасете cedr.

Сравнение данных алгоритмов выполнялось с использованием набора данных из 900 размеченных в соответствии с моделью Келлермана-Плутчика предложений (8 базовых эмоций и отсутствие эмоций, как отдельный класс). Набор данных 7 раз разбивался случайным образом на обучающую и тестовую выборки с отношением количества предложений в обучающей выборке к количеству предложений в тестовой 4 к 1.

Модель DeepPavlov RuBERT-base-cased, предобученная на корпусе русских текстов была взята из библиотеки transformers [26],[27].

Модель cointegrated rubert-tiny2, предобученная на датасете cedr была взята из библиотеки transformers. Ее точность на данном датасете составляла 0.895. [28]

Модель DeepPavlov RuBERT-base-cased была предобучена на датасете cedr самостоятельно тем же методом, что был применен для предобучения модели cointegrated rubert-tiny2. За основу была взята эта модель, предобученная на корпусе русских текстов. Была достигнута точность 0.92.

В качестве наивного байесовского классификатора использовался MultinominalNB из библиотеки scikit-learn. Наивный словарный классификатор обучался на размеченном словаре из 14246 слов.

Классификатор на основе модели DeepPavlov RuBERT-base-cased обучался в течении 70 эпох с использованием функции потерь cross entropy loss с весами классов рассчитанными с помощью метода compute class weights библиотеки scikit-learn и оптимизатора adadelta из библиотеки pytorch optim со стандартными параметрами.

Модель cointegrated rubert-tiny2, предобученная на датасете cedr была взята из библиотеки transformers. Она была обучена в течении 40 эпох и достигла точности 0.89 на этом датасете.

Модель DeepPavlov RuBERT-base-case, была предобучена на датасете cedr самостоятельно, с использованием сервиса google colab. Обучение было выполнено в 10 эпох и была достигнута точность 0.92.

Классификаторы на основе этих моделей обучались в течении 50 эпох эпох с функцией потерь cross entropy loss с весами классов рассчитанными с помощью метода compute class weights библиотеки scikit-learn и оптимизатора adadelta из библиотеки pytorch optim со стандартными параметрами.

В качестве метрик использовались оценка F1, accuracy и Top-3 Accuracy на тестовой выборке. Для сравнения использовалось средние значение этих метрик по результатам 7 экспериментов. Результаты сравнения алгоритмов представлены в таблице 5.

Таблица 5 – Результаты сравнения алгоритмов-претендентов на роль слабого ученика в классифицирующем ансамбле

Метрика	Naive bayes	Naive dict	DP BERT	DP BERT cedr	Cointegrated BERT cedr
F1	0.11	0.16	0.233	0.331	0.29
Accuracy	0.15	0.18	0.318	0.375	0.323
Top-3 accuracy			0.578	0.635	0.569

По результатам сравнения принято решение использовать в качестве слабого ученика классификатор на основе модели DeepPavlov RuBERT, предобученной на датасете cedr. Ее превосходство в точности над cointegrated rubert-tiny2, предобученной на том же датасете, объясняется большим числом параметров, а над DeepPavlov RuBERT, не предобученным на датасете cedr, тем, что предобучение на большом количестве данных, схожих с данными в решаемой задаче, позволило модели лучше научиться выделять признаки, необходимые для определения эмоций.

2.2.4 Выбор метода построения ансамбля

Для определения оптимального метода построения ансамбля, было проведено сравнение метода стекинга и бэгинга с мягким и жестким голосованием при использовании ансамблей из четырех слабых классификаторов. Также проводилось сравнение их результатов при использовании различных методов разбиения на подвыборки. В частности, рассматривались следующие методы: бутстрап, бэгинг и случайные перестановки. Обучение проходило на том же наборе данных. В качестве мета-алгоритма в стекинге выступал линейный слой, принимавший на вход 27 параметров (объединенный выход слабых классификаторов). При обучении слабых учеников использовалась функции потерь cross entropy loss с весами классов, рассчитанными с помощью метода compute class weights библиотеки scikit-learn, и оптимизатор adadelta из библиотеки pytorch optim со стандартными параметрами. При обучении мета-алгоритма использовались те же функция потерь и оптимизатор. Обучение слабых учеников проходило в течении 50 эпох, обучение мета-алгоритма – в течении 200. Для валидации результата использовались те же метрики, что и при сравнении слабых учеников. Результаты сравнения представлены в таблицах 6-8.

Таблица 6 – Результаты сравнения методов построения классифицирующего ансамбля. Метод разбиения – бутстрап

Метрика	Стекинг	Жесткое голосование	Мягкое голосование
F1	0.36	0.31	0.32
Accuracy	0.41	0.38	0.39
Top-3 accuracy	0.6		

Таблица 7 – Результаты сравнения методов построения классифицирующего ансамбля. Метод разбиения – бэгинг

Метрика	Стекинг	Жесткое голосование	Мягкое голосование
F1	0.36	0.32	0.31
Accuracy	0.4	0.39	0.38
Top-3 accuracy	0.56		

Таблица 8 – Результаты сравнения методов построения классифицирующего ансамбля. Метод разбиения – случайные перестановки

Метрика	Стекинг	Бегинг с жестким голосованием	Бегинг с мягким голосованием
F1	0.33	0.33	0.34
Accuracy	0.39	0.39	0.38
Top-3 accuracy	0.63		

По результатам сравнения принято решение использовать в качестве метода разбиения на подвыборки бутстрап, а в качестве метода построения ансамбля стекинг.

2.3 Описание разработанной программы

Полученная программа состоит из двух основных частей – парсер диалогов и анализатор эмоций. Парсер диалогов в виде класса text_parser, с основными методами extract_utterances_and_tokenise, extract_actor_words_ from_utterances, construct_actor_objects. При создании класса парсера ему дается текст, который будет необъодимо обработать. Метод extract_utterances_and_tokenise токенизирует текст, извлекает из него предложения, содержащие диалог и помещает их в отдельный список.

Метод extract_actor_words_from_utterances извлекает из предложений, содержащих прямую речь слова, указывающие на действующие лица. Для каждого предложения составляется список таких слов.

Метод construct_actor_objects создает объекты класса персонаж и устанавливает связи персонаж-высказывания. При этом выполняется разрешение кореферентных связей между словами, указывающими на действующих лиц.

Анализатор эмоций реализован в виде класса emotion_detector. Этот класс представляет из себя собственную реализацию ансамбля типа стекинг. Использовать готовую реализацию стекинга невозможно, ввиду того, что готовые реализации требуют, чтобы каждый слабый ученик был полностью отдельной моделью, а в качестве слабых учеников используются классификаторы на основе модели BERT. Веса предобученной модели BERT занимают около гигабайта памяти, из-за чего невозможно использовать большое количество классификаторов. Однако, благодаря тому, что сама модель BERT не изменяется в ходе обучения, для экономии памяти можно хранить лишь одну такую модель и несколько классифицирующих голов к ней. Смена одной классифицирующей головы на другую в таком случае аналогично переключению между слабыми учениками. Класс emotion_detector реализует в себе такой подход к построению ансамбля из классификаторов на основе модели BERT. При поступлении на вход текста, он выполняет его классификацию каждой из классифицирующих голов и результат отправляется на вход мета-алгоритму, который определяет эмоцию, которая выражена в тексте.

Сама программа при запуске запрашивает путь к файлу, из которого необходимо считать текст. Затем к тексту применяется парсер диалогов, а к полученному списку высказываний – анализатор эмоций. На основе конечного результата генерируются графики изменения эмоций каждого персонажа в ходе повествования. Также, полученный список размеченных по персонажам и эмоциям высказываний можно сохранить в файл формата xlsx.

3 РЕЗУЛЬТАТЫ РАБОТЫ

3.1 Результаты работы программы

Готовая программа выполняется из командной строки и имеет следующие ключи запуска:

- textfile – путь к файлу с анализируемым текстом
- resfile – путь к файлу, в который будет записана таблица с результатом
- plot – ключевое слово, за которым следует имя персонажа и последовательность из названий эмоций, графики для которых надо построить
- weights – путь к файлу с весами предобученной модели, если нужно применить веса, отличные от стандартных

Пример таблицы, генерируемой программой, представлен на рисунке 4. В столбце один указывается номер предложения, содержащего высказывание в тексте, в столбце два и три – текст предложения и наиболее вероятная эмоция, выраженная в нем. В столбце четыре – наиболее часто встречающееся имя персонажа, что его произнес. В следующих девяти столбцах помещаются вероятности, с которой, согласно предсказанию модели, соответствующие эмоции представлены в данном высказывании.

Примеры созданных при работе программы графиков изменения выраженности эмоций в речи персонажей с ходом повествования представлены на рисунках 5 – 8. По оси x указывается номер предложения в тексте, во оси y – вероятность того, что в данном предложении высказывание персонажа содержало рассматриваемую эмоцию.

Эти графики составлены по первой главе романа «Мастер и Маргарита». По ним видно, что иностранец в течении повествования скорее испытывает удивление, а Берлиоз – радость или восторг. По двум последним графикам можно проследить, как Берлиоз испытывает страх ближе к концу главы после вспышки злости в речи иностранца в середине.

	A	B	C	D	E	F	G	H	I
1	4	– Дайте нар	нет эмоций	берлиоз	1.4247506e-0	1.217995e-06	2.7771039e-0	7.2011954e-0	2.0687244e-0
2	5	– Нарзану не	злость	женщина	0.000863816	0.001064185	0.9313898	0.002276112	0.051858265
3	6	– Пиво есть	ожидание	бездомный	0.003404973	0.000344620	0.000104388	0.000619018	0.01814858
4	7	– Пиво прив	нет эмоций	женщина	7.2922717e-0	7.5770953e-0	1.985681e-06	4.0531825e-0	1.1907151e-0
5	8	– А что есть	ожидание	берлиоз	3.991707e-07	2.3821242e-0	2.5377224e-0	6.680884e-06	0.001295207
6	9	– Абрикосов	нет эмоций	женщина	8.191425e-07	8.5886575e-0	2.209254e-06	4.7138994e-0	1.4305116e-0
7	10	– Ну, давайт	радость	берлиоз	0.9864103	0.000458337	0.000468560	0.002219610	0.000336849
8	17	– Фу ты чер	страх	редактор	0.000263829	9.116892e-05	0.000552669	0.000217525	0.000137261
9	22	– Нет ни одн	нет эмоций	берлиоз	8.0773736e-0	8.605686e-07	2.2293964e-0	4.7180006e-0	1.4386903e-0
10	33	– Ты, Иван,	удивление	берлиоз	0.000127185	0.000220348	0.004969532	0.000271927	0.99084455
11	36	– Извините	удивление		1.8640836e-0	3.9700877e-0	2.9995217e-0	3.2546275e-0	0.96725804
12	41	– Разрешите	нет эмоций	иностранец	0.36044335	0.000550544	0.000130730	0.001251873	9.659151e-05
13	42	– Если я не	удивление	иностранец	1.2059995e-0	3.5793673e-0	2.2331622e-0	1.3214756e-0	0.58388597
14	43	– Нет, вы не	радость	берлиоз	0.9990132	3.7180314e-0	6.583879e-06	0.000177199	6.326427e-06
15	44	– Ах, как ин	удивление	иностранец	1.5638087e-0	4.276424e-05	2.986006e-05	2.1800737e-0	0.88493794
16	46	– А вы согл	нет эмоций	неизвестный	3.0878457e-0	3.0587767e-0	6.2781182e-0	2.1993756e-0	8.581882e-06
17	47	– На все сто	радость		0.99977905	1.5160015e-0	2.6136177e-0	8.437486e-05	2.7466021e-0
18	48	– Изумитель	удивление	бездомный	0.000544700	0.000423242	0.000342494	0.000591112	0.9627348
19	49	– Да, мы не	радость	берлиоз	0.89861935	0.003144345	0.001643397	0.006675275	0.001662723
20	51	– Вы – атеи	удивление	иностранец	2.0344338e-0	3.1792217e-0	2.7773678e-0	8.1523154e-0	0.9972589
21	52	– Да, мы – а	радость	берлиоз	0.99789363	9.631288e-05	4.1028994e-0	0.000518967	3.644346e-05
22	53	– Ох, какая	удивление	иностранец	0.012803487	0.003046781	0.002093744	0.003600264	0.91534823
23	54	– В нашей с	ожидание	берлиоз	2.0568365e-0	4.292772e-05	3.561596e-05	7.260794e-05	0.004147940
24	56	– Позвольте	радость	иностранец	0.9997756	1.5325066e-0	2.641778e-06	8.554094e-05	2.7764813e-0
25	57	– За что это	удивление	бездомный	1.0702015e-0	3.1990923e-0	1.688004e-05	1.3734781e-0	0.6095328
26	58	– За очень в	радость		0.9997781	1.5205625e-0	2.6215487e-0	8.46024e-05	2.7539961e-0
27	61	– Но, позвол	страх		0.003961994	0.001764466	0.007862822	0.002090079	0.024203265
28	62	– Увы! – с с	грусть	берлиоз	0.000210185	0.000195880	0.000391566	0.8824379	0.000415890
29	63	– Браво! – в	удивление	иностранец	2.0768091e-0	3.2519012e-0	2.8373435e-0	8.1478815e-0	0.9971119
30	64	– Доказател	злость	редактор	0.000519665	0.000315151	0.99140877	0.000387912	0.003919970
31	66	– Взять бы э	удивление	иван	3.0903288e-0	4.494237e-05	6.9613154e-0	7.983242e-05	0.9957646
32	67	– Иван! – ск	радость	берлиоз	0.9997756	1.5330987e-0	2.6464127e-0	8.5186155e-0	2.7776944e-0
33	69	– Именно, и	страх	иностранец	0.000740205	0.000232415	0.016428286	0.000923074	0.000273002
34	71	– Но, – прод	удивление		1.9285964e-0	3.7784073e-0	2.9061817e-0	4.1896896e-0	0.98336154
35	72	– А жаль! –	грусть		9.1541835e-0	9.161372e-05	0.000170912	0.939519	0.000179783
36	73	– И мне жал	страх	неизвестный	0.000660436	0.000296296	0.000834976	0.12690657	0.000267317
37	74	– Сам челов	ожидание	бездомный	9.462676e-06	2.3311071e-0	9.625047e-06	2.0509928e-0	0.32171014
38	75	– Виноват,	злость	иностранец	0.001349662	0.001880695	0.5307122	0.002434892	0.4386402
39	77	– Вы хотите	ожидание	иностранец	3.6988524e-0	2.164476e-06	2.2595596e-0	6.1326846e-0	0.001352953
40	78	– А у вас ра	ожидание	поэт	7.570634e-06	2.8654826e-0	1.5145173e-0	8.529417e-05	0.052446302
41	79	– Какие пред	нет эмоций	иностранец	2.3530425e-0	3.25025e-05	5.1152736e-0	3.422488e-05	0.000274597
42	80	– Ну, «Нашу	злость	бездомный	0.000130996	0.000122457	0.9964324	0.000161734	0.001570511
43	82	– «Наша мар	нет эмоций	иностранец	1.2085753e-0	1.0839448e-0	2.5586075e-0	6.2185075e-0	1.7921918e-0
44	88	– Да, челове	грусть		9.252409e-05	9.1178976e-0	0.000170838	0.9376127	0.000176919
45	90	– Ну, здесь	страх	берлиоз	0.000264632	9.13922e-05	0.000553248	0.000218014	0.000138207

Рисунок 4 – Пример таблицы, генерируемой программой

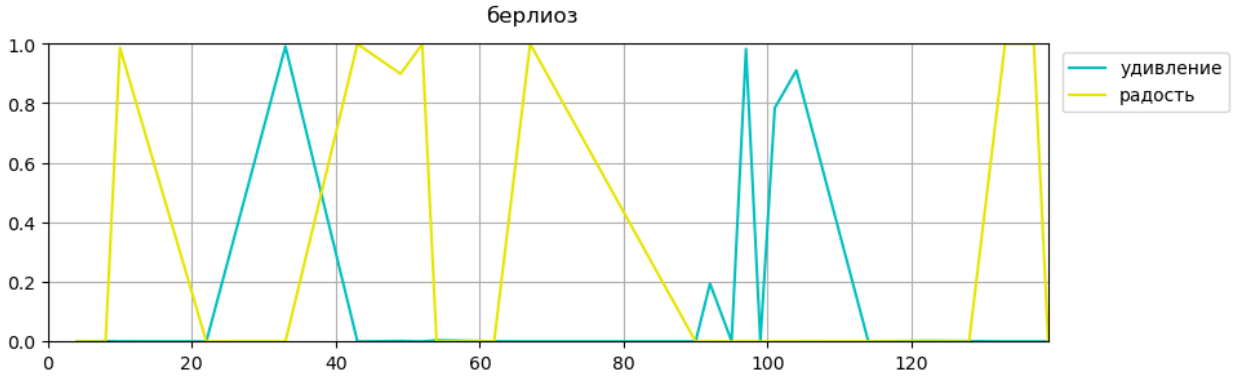

Рисунок 5 – Пример графика изменения выраженности эмоций в речи с ходом повествования

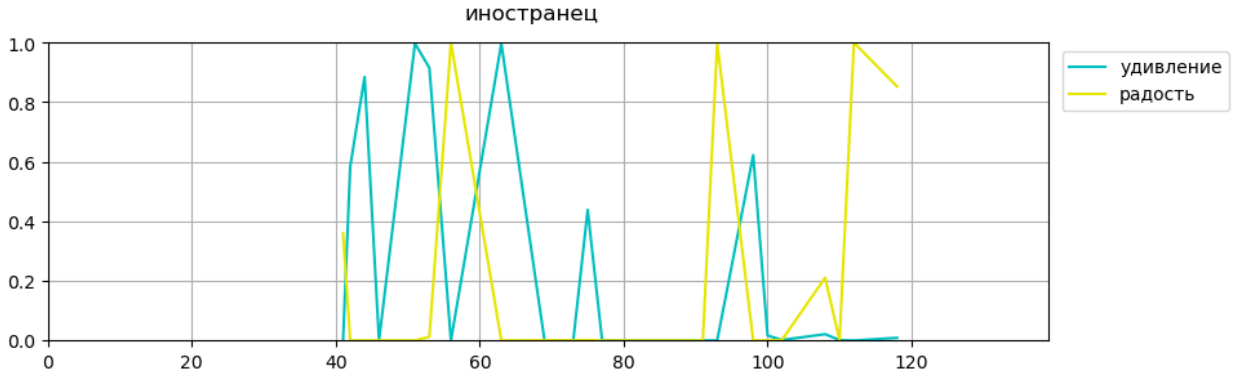

Рисунок 6 – Пример графика изменения выраженности эмоций в речи с ходом повествования

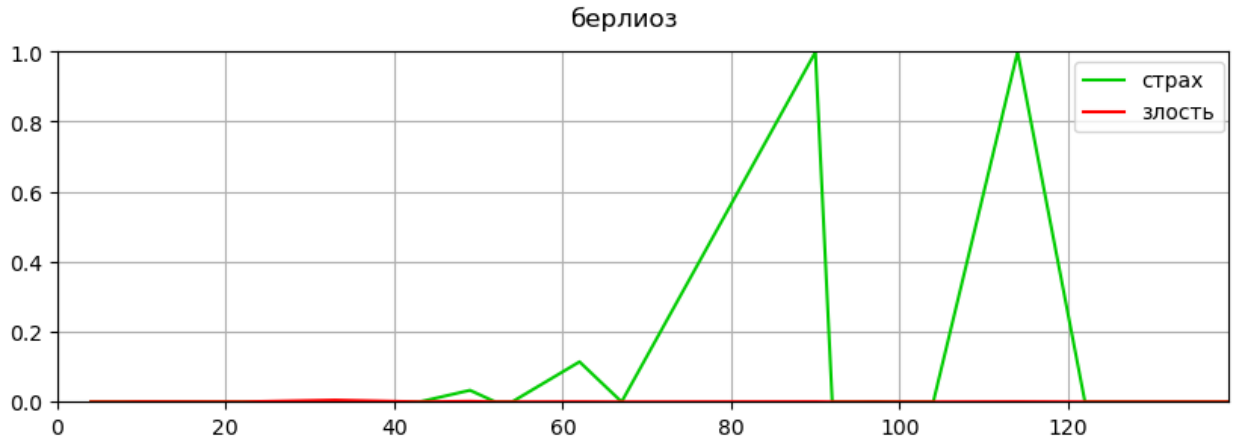

Рисунок 7 – Пример графика изменения выраженности эмоций в речи с ходом повествования

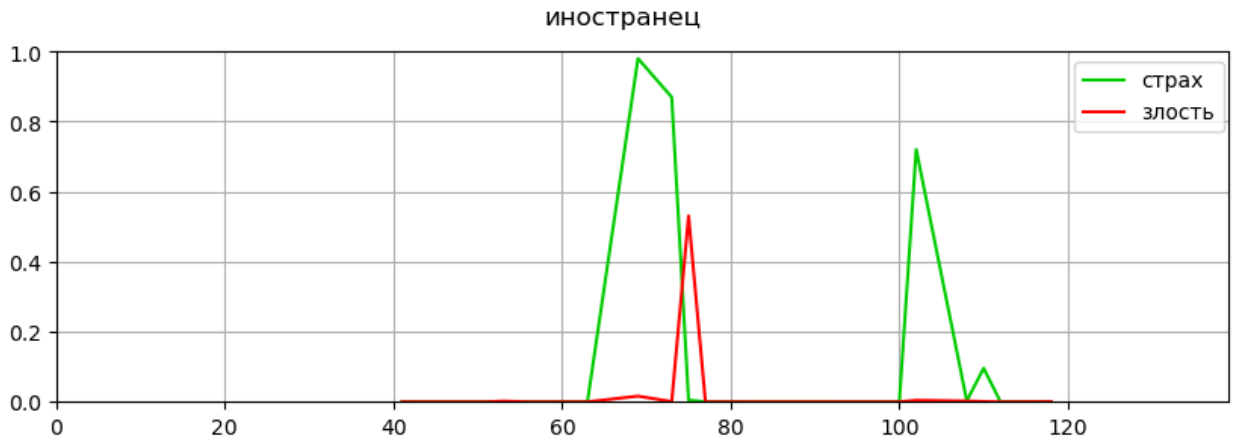

Рисунок 8 – Пример графика изменения выраженности эмоций в речи с ходом повествования

3.2 Технические характеристики программы

Для анализа качества работы парсера диалогов было создано два набора данных, представлявших из себя художественные тексты с вручную определенными персонажами, их высказываниями и словами, которые использовались для указания на этих персонажей в авторской речи при их высказываниях. Метрики на первом наборе данных (назовем его текстом А) использовались для ручной настройки алгоритмических методов, лежащих в основе парсера диалогов. Второй набор данных (назовем его текстом Б) использовался только для валидации результатов работы готовой модели. Он был исследован парсером только после завершения работы над ним, с целью проверить его способность работать над любым текстом, а не только над текстом, на котором был настроен.

Для анализа качества работы первого этапа парсера диалогов использовалась метрика точности – отношение количества предложений, в которых слова, указывающие на действующих лиц, были правильно определены к количеству обрабатываемых предложений. Так как в предложении может содержаться несколько слов, указывающих на действующих лиц, в случае, если списки выявленных парсером слов, указывающих на действующих лиц и слов, действительно указывающих на действующих лиц в рассматриваемом предложении, совпадали, к сумме правильно обработанных предложений добавлялось дробное число – отношение количества правильно определенных слов, указывающих на действующих лиц в данном предложении к отношению количества слов, выделенных парсером, как указывающих на действующих лиц. Результаты оценки точности работы парсера диалогов применимо к задаче определения слов, указывающих на действующих лиц в авторской речи представлены в таблице 9.

Таблица 9 – Результаты сравнения точности работы парсера диалогов при использовании разных конфигураций методов выделения слов, указывающих на действующих лиц

	Базовый метод	Вспомогательный метод 1	Вспомогательный метод 2	Вспомогательный метод 3	Все методы вместе
Точность на тексте А	0.714	0.482	0.919	0.88	0.974
Точность на тексте Б	0.767	0.630	0.848	0.832	0.921

Из результатов тестирования можно заключить, что парсер диалогов стабильно показывает точность выше 0.9 при решении задачи выделения слов, указывающих на действующих лиц даже на новых текстах.

Для анализа качества работы второго этапа парсера диалогов использовалась метрика достаточной точности и полной точности. Достаточная точность оценивает способность алгоритма приписать предложение правильному референту. Полная точность оценивает способность алгоритма собрать все объекты, обозначающие один референт в один объект. Результаты оценки точности работы парсера диалогов применимо к задаче разрешения кореференции между словами, указывающими на действующих лиц представлены в таблице 10.

Таблица 10 – Результаты оценки точности разрешения кореференции парсером диалогов.

	Текст А	Текст Б
Достаточная точность	0.847	0.775
Полная точность	0.286	0.321

Из результатов тестирования можно заключить, что парсер диалогов хорошо решает задачу приписывания предложений правильному референту, однако испытывает трудности с объединением всех объектов класса референт, обозначающих одного персонажа в один объект.

Для анализа качества работы модели, осуществляющей определение эмоций в предложениях, использовались метрики accuracy и матрица ошибок.

Также выполнялось построение кривых обучения. Также выполнялась кросс-валидация случайными разбиениями. Классификатор обучался на наборе из 455 предложений, содержащих прямую речь размеченных в соответствии с моделью эмоций Келлермана-Плутчика. Использовались восемь базовых эмоций и отсутствие эмоций, как девятый класс. Предложения были извлечены из первых десяти глав романа «Мастер и Маргарита» с помощью парсера диалогов. Кривые обучения слабых учеников и мета-алгоритма, а также матрица ошибок тестовой выборке представлены на рисунках 9, 10 и 11. Резкий взлет точности на первых эпохах и следующая за ним стабилизация точности обусловлены тем, что модель, лежащая в основе классификатора была заранее оптимизирована для определения эмоций посредством обучения на датасете cedr. Благодаря этому классифицирующие персептроны очень быстро вышли на максимум точности. Сильные колебания метрик от эпохи к эпохе обусловлено тем, что при обучении остаются включенными dropout-слои модели BERT. Это необходимо для повышения точности обучения и снижения переобучения. В таблице 11 представлены метрики в разбиении по классам.

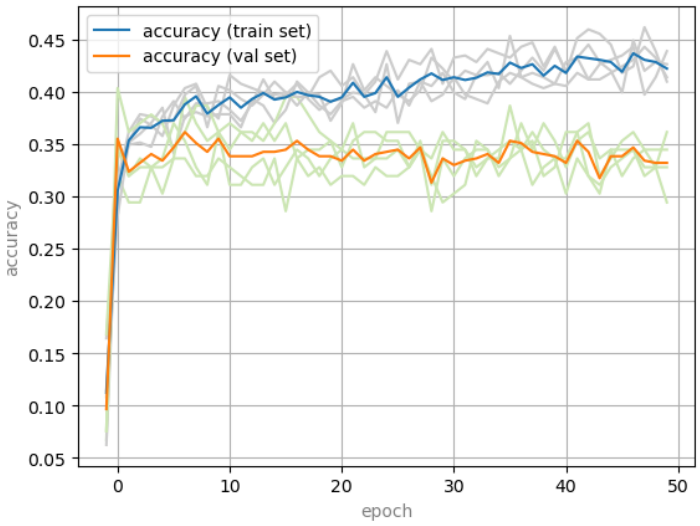

Рисунок 9 – Кривые зависимости средней точности на обучающей и тестовой выборках от числа пройденных эпох для слабых учеников ансамбля

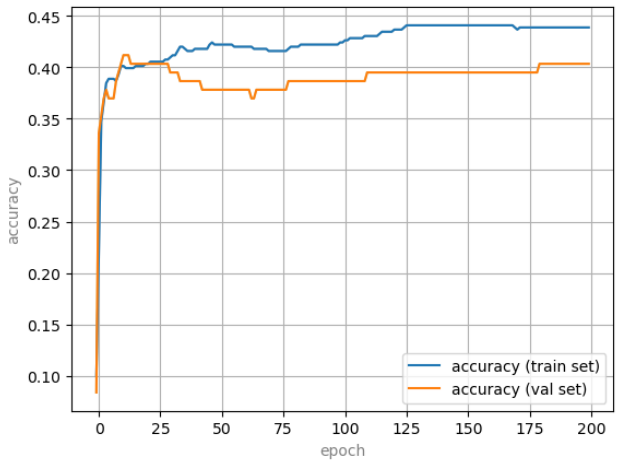

Рисунок 10 – Кривые зависимости средней точности на обучающей и тестовой выборках от числа пройденных эпох для мета-алгоритма

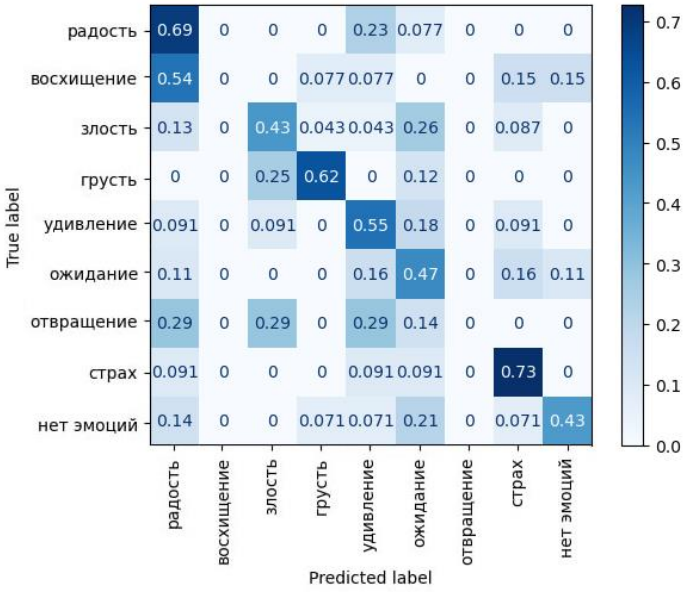

Рисунок 11 – матрица ошибок на тестовой выборке

Таблица 11 – Метрики на тестовой выборке в разбиении по классам

	Радость	Восхищение	Злость	Грусть	Удивление	Ожидание	Отвращение	Страх	Нет эмоций
Precision	0.33	0.00	0.67	0.62	0.33	0.38	0.00	0.49	0.60
Recall	0.69	0.00	0.43	0.62	0.55	0.49	0.00	0.73	0.43
F1	0.45	0.00	0.53	0.62	0.51	0.42	0.00	0.57	0.50
Train support	67	32	57	38	62	61	39	59	66

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы бакалавра создана модель на основе нейронных сетей глубокого обучения, способная определять эмоции персонажей в художественном тексте посредством анализа диалогов. Для этого проведено сравнение различных алгоритмов машинного обучения применимо к задаче определения эмоций в тексте и выбран алгоритм, показавший наилучшие результаты, коим оказался ансамбль типа стекинг из классификаторов на основе модели BERT. Также был создан парсер диалогов, при помощи алгоритмической системы правил осуществляющий извлечение из текста действующих лиц и их реплик.

Разработанный парсер диалогов показывает точность определения слов, указывающих на действующих лиц в предложениях, содержащих прямую рчеь 0.92 и точность установления связей между персонажами и высказываниями 0.77. Он может быть использован и отдельно от анализатора эмоций, например, для составления наборов данных из высказываний прямой речи. В частности, он был применен таким образом при составлении обучающей выборки для классификатора эмоций, что значительно облегчило ее формирование и разметку.

Классификатор эмоций показывает точность определения эмоций 0.41. Низкая точность определения эмоций связана, в первую очередь, с недостатком данных для обучения, а также с низким качеством некоторых обучающих наборов данных. То, что точность финальной версии модели, несмотря на малый объем обучающих данных, значительно превосходит точность случайного угадывания, а также то, что с ростом размера применяемого для обучения набора данных при прочих равных наблюдается рост точности классификации при валидации, доказывает, что задача принципиально решаема, и, при наличии большего объема данных можно достичь более качественного результата.

По вышеописанным причинам, точность определения эмоций не достигла точности, показываемой лучшими моделями, решающими эту

задачу, впрочем, ввиду того, что такие модели решают задачу для размеченного англоязычного текста, можно считать, что у полученного решения нет прямых аналогов, решающих поставленную задачу лучше него.

Можно сделать вывод, что при дальнейшей разработке данной темы необходимо сделать фокус на сборе и разметке данных для обучения.

Также можно рассмотреть отказ от использования модели Келлермана-Плутчика в пользу модели эмоций, используемой в датасете cedr. Так как в ходе выполнения работы было установлено, что классификаторы на основе модели BERT могут обучиться на датасете cedr до точности 0.9+ на тестовой выборке, можно считать, что такая замена приведет к значительному повышению качества классификации, однако будет потеряна полнота и расширяемость модели эмоций. При этом, все еще будет необходимо использовать датасет на основе художественных текстов для дообучения по причине того, что датасет cedr составлен исключительно из постов в социальных сетях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Куратов Ю. М. Специализация языковых моделей для применения к задачам обработки естественного языка. // Кандидатская диссертация – МФТИ – 2020.

2. Acheampong F. A., Wenyu C., Nunoo-Mensah H. Text-based emotion detection: Advances, challenges, and opportunities //Engineering Reports. – 2020. – Т. 2. – №. 7. – С. e12189.

3. Poria S. et al. Emotion recognition in conversation: Research challenges, datasets, and recent advances //IEEE Access. – 2019. – Т. 7. – С. 100943-100953.

4. Graterol W. et al. Emotion detection for social robots based on NLP transformers and an emotion ontology //Sensors. – 2021. – Т. 21. – №. 4. – С. 1322.

5. Самигулин Т. Р., Джурабаев А. Э. У. Анализ тональности текста методами машинного обучения //Научный результат. Информационные технологии. – 2021. – Т. 6. – №. 1. – С. 55-62.

6. Acheampong F. A., Nunoo-Mensah H., Chen W. Transformer models for text-based emotion detection: a review of BERT-based approaches //Artificial Intelligence Review. – 2021. – С. 1-41.

7. Yang X. et al. A Study of Text Vectorization Method Combining Topic Model and Transfer Learning //Processes. – 2022. – Т. 10. – №. 2. – С. 350.

8. Church K. W. Word2Vec //Natural Language Engineering. – 2017. – Т. 23. – №. 1. – С. 155-162.

9. Rish I. et al. An empirical study of the naive Bayes classifier //IJCAI 2001 workshop on empirical methods in artificial intelligence. – 2001. – Т. 3. – №. 22. – С. 41-46.

10. Medsker L. R., Jain L. C. Recurrent neural networks //Design and Applications. – 2001. – Т. 5. – С. 64-67.

11. Vaswani A. et al. Attention is all you need //Advances in neural information processing systems. – 2017. – Т. 30.

12. Han K. et al. Transformer in transformer //Advances in Neural

Information Processing Systems. – 2021. – T. 34. – C. 15908-15919.

13. Dehghani M. et al. Universal transformers //arXiv preprint arXiv:1807.03819. – 2018.

14. Garg S., Ramakrishnan G. Bae: Bert-based adversarial examples for text classification //arXiv preprint arXiv:2004.01970. – 2020.

15. Tenney I., Das D., Pavlick E. BERT rediscovers the classical NLP pipeline //arXiv preprint arXiv:1905.05950. – 2019.

16. Dong X. et al. A survey on ensemble learning //Frontiers of Computer Science. – 2020. – T. 14. – C. 241-258.

17. Wong S. C. et al. Understanding data augmentation for classification: when to warp? //2016 international conference on digital image computing: techniques and applications (DICTA). – IEEE, 2016. – C. 1-6.

18. Baldi P., Sadowski P. J. Understanding dropout //Advances in neural information processing systems. – 2013. – T. 26.

19. Zhuang F. et al. A comprehensive survey on transfer learning //Proceedings of the IEEE. – 2020. – T. 109. – №. 1. – C. 43-76.

20. Hossin M., Sulaiman M. N. A review on evaluation metrics for data classification evaluations //International journal of data mining & knowledge management process. – 2015. – T. 5. – №. 2. – C. 1.

21. Lever J. Classification evaluation: It is important to understand both what a classification metric expresses and what it hides //Nature methods. – 2016. – T. 13. – №. 8. – C. 603-605.

22. Stone M. Cross-validation: A review //Statistics: A Journal of Theoretical and Applied Statistics. – 1978. – T. 9. – №. 1. – C. 127-139.

23. Бодрова А. А. Разрешение кореференции методом кластеризации. // Магистерская диссертация – СБГУ – 2016.

24. Sreeja P., Mahalakshmi G. Emotion models: a review //International Journal of Control Theory and Applications. – 2017. – T. 10. – №. 8. – C. 651-657.

25. Sboev A., Naumov A., Rybka R. Data-Driven Model for Emotion

Detection in Russian Texts //Procedia Computer Science – 2021 Volume 190, Pages 637-642.

26. Burtsev M. et al. Deeppavlov: Open-source library for dialogue systems //Proceedings of ACL 2018, System Demonstrations. – 2018. – С. 122-127.

27. DeppPavlov/rubert-base-cased model trained on Russian part of Wikipedia and news data. – URL: https://huggingface.co/DeepPavlov/rubert-base-cased (дата обращения 9.04.2023)

28. Cointegrated/rubert-tiny model fine-tuned for emotion recognition in Russian language. – URL: https://huggingface.co/cointegrated/rubert-tiny2-cedr-emotion-detection (дата обращения 10.05.2023).