Федеральное государственное автономное образовательное учреждение высшего образования
«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ» (РУДН)

Факультет физико-математических и естественных наук

Кафедра математического моделирования и искусственного интеллекта

Основное учебное подразделение: факультет физико-математических и естественных наук
Направление/специальность: 02.03.01 Математика и компьютерные науки

ОТЧЁТ

о прохождении учебной практики

«Научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)»

Абу Сувейлим Мухаммед Мунифович

Курс, группа: 3, НКНбд-01-21

Место прохождения практики: научные центры института прикладной математики и телекоммуникаций

Сроки прохождения с «15» апреля 2024 г. по «15» июня 2024 г.

Руководители практики:

от РУДН Фомин М.Б., к.ф.-м.н, доцент
от организации (предприятия) Самуйлов К.Е., директор ИКНиТ

Оценка _____

Москва, 2024 г.

Содержание

1 Введение **2**
1.1 Цель работы 2
1.2 Задание 3

2 Основная часть **4**
2.1 Теоретическая часть 4
2.1.1 Извлечение именованных сущностей Named Entity Recognition 5
2.1.2 Анализ тональности текста по отношению к объекту . . 6
2.2 Листинг программы 7
2.3 Полученные результаты и их анализ 10

3 Выводы **15**

Список литературы **16**

1 Введение

1.1 Цель работы

Согласно программе учебной практики направления подготовки 02.03.01 «Математика и компьютерные науки», целями практики являются:

- формирование профессиональных навыков в проведении научных исследований;
- формирование навыков использования современных научных методов для решения научных и практических задач;
- формирование практических навыков написания вспомогательных программных комплексов для проведения вычислительных экспериментов;
- формирование общекультурных, общепрофессиональных и профессиональных компетенций в соответствии с ОС ВО РУДН;
- формирование навыков оформления и представления результатов научного исследования;
- формирование навыков работы с источниками данных.

Для достижения целей в рамках учебной практики был выполнен обзор публикаций российских и международных научных изданий по теме выпускной квалификационной работы (ВКР) бакалавра, которая определена как «Анализ тональности финансовых новостей».

Время проведения учебной практики – 15.04.2024г.-15.06.2024г., место проведения практики – Отдел технической поддержки пользователей (департамент технологических и информационных ресурсов) РУДН и научные центры института компьютерных наук и телекоммуникаций РУДН.

Последовательность прохождения практики, перечень работ, выполненных в процессе практики (таб. 1):

Таблица 1: Последовательность прохождения практики, перечень работ, выполненных в процессе практики

№ п/п	Работы и мероприятия	Пояснение	Сроки выполнения
1	Установочное занятие	Инструктаж по безопасности труда и правилам пожарной безопасности при выполнении лабораторных и практических работ. Обсуждение задания на практику. Разъяснение требований к заполнению	15.04.2024
2	Подбор материалов	Подбор материалов для написания НИР.	16.04.-18.04.2024
3	Оформление введения и теоретической части НИР. Накопление, систематизация и анализ теоретических и прикладных материалов. Составление библиографии по основным источникам.	Систематизация и обобщение научной и учебной литературы.	19.04-22.04.2024
4	Подготовка материалов НИР. Практическая часть.	Изучение научной и учебной литературы.	23.04.-03.05.2024
5	Подготовка материалов НИР. Практическая часть.	Проведение развернутого эксперимента для НИР. Обсуждение результатов эксперимента.	04.05.-07.05.2024
6	Написание второго раздела НИР.	Сбор данных для дальнейшей реализации алгоритмов.	08.05.-18.05.2024
7	Подготовка отчета и дневника по практике.	Разработка реализации алгоритмов, реализующих решение задачи НИР.	20.05-08.06.2024
8	Подготовка отчета и дневника по практике.	Собеседование с научным руководителем и руководителем практики по содержанию отчета и дневника по практике.	10.06-11.06.2024
9	Сдача отчета и дневника по практике руководителю практики.	Прикрепит нужные документы и файлы к ТУИС.	15.06.2024

1.2 Задание

Заданием является анализ тональности текста и состоит оно из двух частей. Получив текст на английском языке, модель должна определить оценку тональности текста и к какому объекту в тексте тональность относится. Рассмотрим следующий пример: «Apple stocks went up by 10% on Monday, as reported by TASS». В тексте две сущности: «Apple» и «TASS», поэтому программа должна вывести два двухэлементных кортежа: (Apple, Positive) и (TASS, Negative). На самом деле, в рамках НИР было достаточно определить оценку тональности, но было принято решение определить объекты/сущности в тексте.

2 Основная часть

2.1 Теоретическая часть

Анализ тональности текста – это подраздел обработки естественного языка (NLP), целью которого является классификация текста по тональности. Тональность — это мнение, отношение и эмоции автора по отношению к объекту, о котором говорится в тексте [SD21]. Также объекты можно разделить на разные категории. В Python библиотеке SpaCy, одна из самых распространнейных библиотек в сфере NLP, существует 18 категорий объектов, такие как человек (PERSON); национальности, религиозные или политические группы (NORP); здания, аэропорты, автомагистрали, мосты (FAC), компании, агентства, учреждения (ORG) и так далее []. Прежде чем начать, нужно дать определение нескольким важным терминам.

Определение 2.1. *Мнение - это четырехэлементный кортеж*

$$(g, s, h, t),$$

где g - целевой объект тональности (sentiment target), s - тональность мнения относительно объекта g (sentiment), h - носитель мнения (лицо или организация, придерживающиеся мнения; opinion holder), а t - время, когда мнение было выражено (time). (13, [17a])

Отсутствие хотя бы одного элемента из кортежа резко ухудшает качество информации. Например, мнение, имеевшее место 5 лет назад, в большинстве случаев является неактуальным по сравнению с мнением на сегодня. И если не указать этот временной отрезок, то информация может оказать недостоверной. Также имеет большое значение, кто является носителем такого мнения, например, презедент РФ или средестатистический гражданин РФ.

Стоит отметить, что у мнения есть объекты/сущности. В предложении с несколькими объектами нужно определить конкретный объект для каждого положительного или отрицательного мнения. Рассмотрим в качестве примера следующее предложение: «Apple stocks went up on Monday, while Microsoft stocks went down». Оно имеет как позитивную,так и негативную тональность. Объект с позитивной тональностью - Apple, объект с негативной тональностью - Microsoft.

Определение 2.2. *Целевой объект тональности, также известный как целевой объект для выражения мнения - это объект, сущность, часть или атрибут объекта, в отношении которого было выражено мнение с эмоциональной окраской.(14, [17a])*

Существуют различные формулировки определений понятий «мнение/тональность», отличающиеся составом, например, Лю определил мнение как кортеж из 5 элементов: <носитель тональности, объект тональности, аспект/атрибут объекта тональности, оценка тональности, время, когда мнение было выражено>[Liu10].

В этой научно-исследовательской работе нас интересует оценка тональности (позитивная, негативная, нейтральная) и объект тональности финансовых новостей.

Анализ тональности финансовых новостей - это двухэтапная задача. Первая - это задача извлечения именованных сущностей (NER) из текста и классификация их по категориям, таким как имена людей, названия организаций, даты, местоположения, суммы денег и другие типы специфических объектов [Nit24]. Вторая - это задача анализа и классификации тональности в тексте по трем категориям: Positive (Положительный), Negative (Отрицательный), Neutral (Нейтральный). Такая задача называется Target (Aspect) Based Sentiment Analysis (TBSA или ABSA). В переводе на русский язык это звучит как анализ тональности/эмоциональности на основе аспектов. ABSA как тема исследования получила особое внимание на соревновании/воркшопе SemEval-2014 [14], где она была впервые представлена в качестве четвертого задания этого соревнования и снова появилась на SemEval-2015 [15] и SemEval-2016 [16] в последующие годы.

С появлением технологии трансформеров в 2017 г. [17b] в области обученных языковых моделей был сделан большой шаг вперед, пример таких технологий - generative pre-trained transformer (GPT) ([18]) и BERT [19a]. В этой научно-исследовательской работе мы будем использовать механизм модели Local Context Focus (LFC, механизм фокусировки на локальном контексте) [19c], использующий блоки внимания (Multi-head Self-Attention) и BERT. Для имплементации этой модели мы выбрали Python фреймворк pyABSA [YL22], который содержит множество моделей ABSA и упрощает процесс программирования моделей и визуализации графиков обучения и тестирования моделей.

При осуществлении анализа термины «настроение», «тональность» и «сентиментальность» будут использованы взаимозаменяемо. Для обозначения первой задачи вместо NERO используется термин Aspect Term Extraction (извлечение аспектного термина, APE), для задачи классификации используется термин Aspect Polarity Classification (классификация полярности аспектов, APC). Также не проводится разграничение между ABSA и TBSA.

2.1.1 Извлечение именованных сущностей Named Entity Recognition

В модели Fast LFC прямо не применяются какие-либо методы для извлечения именованных сущностей. Для этого в Fast LFC ипользуются слои Context features Dynamic (CDM) и/или Mask Context Features Dynamic Weighted (CDW). Слой CDM фокусируется на локальном контексте, маскируя выходные представления менее семантически относящихся к контексту слов. Слой CDW ослабляет характеристики менее семантически относительных контекстных слов.

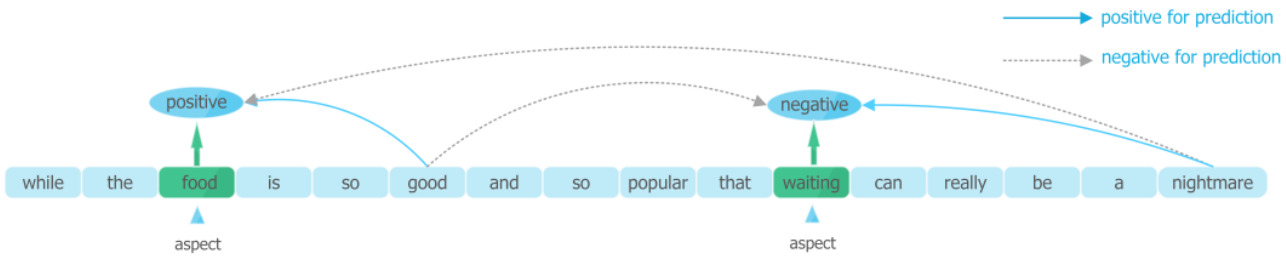

Рис. 1: Влияние контекста на предсказание тональности. Показано только влияние типичных сентиментальных контекстных слов. [19c]

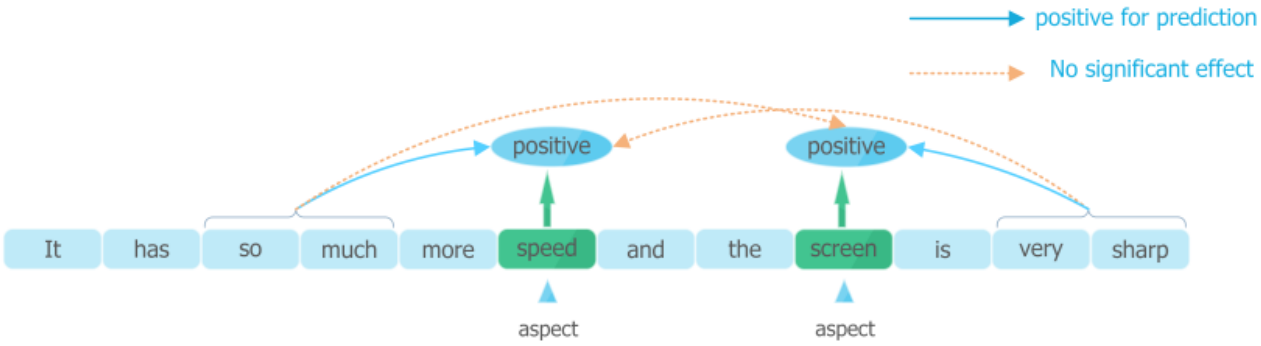

Рис. 2: Влияние контекста на предсказание тональности. Показано только влияние типичных сентиментальных контекстных слов. [19c]

Если рассматривается только задача извлечения именованных сущностей, то для этого существуют более простые варианты/методы в библиотеках spaCy и NLTK.

2.1.2 Анализ тональности текста по отношению к объекту

В Fast LCF модели используется версия DeBERTA от BERT. BERT (и его разные версии DeBERTA, RoBERTa) позволяет получать высокие (state-of-the-art) результаты по задачам NLP, такие как оценка GLUE, составляющая 80.5% (абсолютное улучшение на 7.7%); точность MultiNLI до 86,7% (абсолютное улучшение на 4,6%), оценка F1 ответов на вопросы SQuAD v1.1 до 93,2% (абсолютное улучшение на 1,5%) и SQuAD v2.0 Тестируют F1 до 83,1% (абсолютное улучшение на 5,1%) [19b].

В отличие от Open AI GPT и ELMo, BERT одновременно рассматривает как левую, так и правую сторону контекста. Хотя ELMo тоже использует двухнаправленный подход, но это лишь конкатенация контекстуальных представлений каждого токена слева направо и справа налево. С другой стороны, BERT устраняет ограничение однонаправленности, используя предварительную цель обучения “замаскированной языковой модели” (masked language model, MLM), вдохновленную задачей Клозе 1953 г. [19b].

BERT использует вложения/эмбеддинги WordPiece со словарным запасом в 30 000 токенов. Первый токен каждой последовательности (текста) всегда является специальным классификационным токеном (special classification token, [CLS]). Конечное скрытое состояние/пространство (hidden state), соответствующее этому токену, используется в качестве представления совокуп-

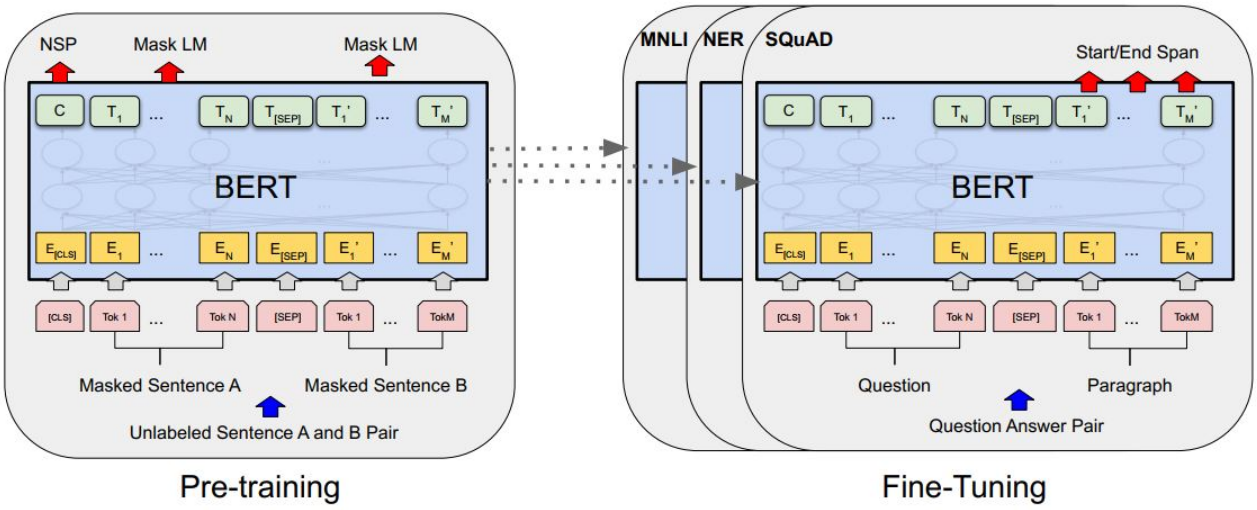

Рис. 3: Общие процедуры pre-training и fine-tuning для BERT. Помимо слоя вывода, для pre-training и fine-tuning используются одни и те же архитектуры. Одни и те же параметры предварительно обученной модели используются для инициализации моделей для различных (down-stream) задач, выполняемых в нисходящем потоке. Во время fine-tuning все параметры настраиваются точно. [CLS] - это специальный символ, добавляемый перед каждым примером ввода, а [SEP] - специальный разделительный токен (например, для разделения вопросов и ответов).

ной последовательности для задач классификации. Другими словами, токен [CLS] - это вся последовательность ввода или предложение в виде вектора. Пары предложений объединяются в единую последовательность. BERT различает предложения двумя способами. Во-первых, BERT разделяет предложения специальным символом ([SEP]). Во-вторых, BERT добавляет к каждому символу выученный эмбеддинг (learned embedding), указывающий, принадлежит ли он предложению A или предложению B. Как показано на рисунке 1, E обозначает вложение входных данных (input embedding) ,

$$C \in \mathbb{R}^H$$

- специальный токен [CLS] конечного скрытого вектора, а

$$T_i \in \mathbb{R}^H$$

- конечный скрытый вектор для i-го входного токена.

2.2 Листинг программы

Перед имплементацией модели обратим внимание на структуру набора данных. В отличии от обычной задачи классификации эмоциональной окраски текста, где у нас есть string текст и бинарный класс либо многоклассовый класс меток (позитивных, негативных и/или нейтральных), в задаче анализа тональности текста на основе аспектов ABSA также есть класс объектов тональности. Иногда добавляется класс категорий объектов и класс атрибутов/аспектов для каждого объекта. Это значительно усложняет задачу классификации, так как это приводит к отсутствию стандартной структуры набора данных. В этой НИР мы придерживаемся структуры набора данных, использованных в воркшопах на SemEval 2014, SemEval 2015 и SemEval 2016.

Листинг 1: Структура набора данны

```
Forward O -999
Markets O -999
Commision O -999
allows O -999
national B-ASP Neutral
commodity I-ASP Neutral
bourses I-ASP Neutral
to O -999
impose O -999
different O -999
deal O -999
charges O -999
```

В наборе следующие ATE метки:

- 'O' (Outside) - указывает, что слово не является частью аспектного термина.
- 'B-ASP' (Beginning of Aspect Term) - указывает на начальное слово аспектного термина.
- 'I-ASP' (Inside Aspect Term) - указывает, что слово находится внутри аспектного термина, т.е. является частью аспектного термина.

Метки тональности после каждого аспекта:

- 'Positive' - положительное значение/слово.
- 'Negative' - отрицательное.
- 'Neutral' - нейтральное.
- '-900' - фиктивное значение (placeholder/dummy value).

Переходим к имплементации модели анализа тональности финансовых новостей на основе аспектов используя Python фреймворк pyABSA. Выполнение кода шаг за шагом позволяет получить те же самые результаты, полученные в этой работе. Модель была обучена на Kaggle Notebook с использованием видеокарты NVIDIA Tesla T4.

Во-первых, нужно установить фреймворк pyABSA и пакет autocuda для обучения модели на видеокартах от NVIDIA.

Листинг 2: установка фреймворка

```
!pip install pyabsa -U
!pip install autocuda
```

Во-вторых, мы импортируем необходимые библиотеки.

Листинг 3: установка фреймворка

```
import autocuda
import random
from pyabsa import AspectTermExtraction as ATEPC
# Put your dataset into integrated_datasets folder, it this folder does not exist, you
    need to call:
from pyabsa import download_all_available_datasets
from pyabsa import DatasetItem
```

```
from pyabsa import ModelSaveOption, DeviceTypeOption
# MetricVisualizer to create graphs
from metric_visualizer import MetricVisualizer
import warnings
import torch
#to insure the models works use these versions
from pyabsa import __version__
assert __version__ >= '1.8.20'

from metric_visualizer import __version__
assert __version__ >= '0.4.0'
```

Далее, скачаем любой из наборов данных доступных в фреймворке либо прикрепляем собственный набор данных к директории. Так как нас интересуют финансовые новости, выбираем набор данных finNews.

Листинг 4: Выбар набора данных

```
warnings.filterwarnings('ignore')
download_all_available_datasets()
dataset = DatasetItem(r"integrated_datasets/atepc_datasets/133.finNews")
```

Существует разные модели на разных архитектурах. Больше всего для нас подходит модель Fast LCF.

Листинг 5: Выбар модели

```
config = (
    ATEPC.ATEPCConfigManager.get_atepc_config_english()
)  # this config contains 'pretrained_bert', it is based on pretrained models
config.model = ATEPC.ATEPCModelList.FAST_LCF_ATEPC  # improved version of LCF-ATEPC
```

Далее, настроим параметры конфигурации. Настройка параметров зависит от доступных вычислительных ресурсов. Мы построим три модели, для каждой модели параметр маскимальной последовательности токенов/слов составляет 60, 80 и 100 соответственно. Размер мини-пакета или размер батча также зависит от объема оперативной памяти видеокарты. По рекомендации автора фреймворка лучше начать с батча размером 64, постепенно его уменьшая, пока модель не будет обучаться. Было выбрано две эпохи обучения. По умолчанию функция потери это кросс-энтропия от PyTorch. Поменять ее через фреймворк нельзя, но можно в ручную изменить код самой модели Fast LFC (см. листинг 6). Одна эпоха на видеокарте NVIDIA T4 занимает около 20 минут, но чем длиннее последовательность токенов или слов, тем больше по времени модель будет обучаться.

Листинг 6: Параметры модели данных

```
seeds = [random.randint(0, 10000) for _ in range(3)]
max_seq_lens = [60, 80, 100]

#config['auto_device'] = True
config.pretrained_bert = "yangheng/deberta-v3-base-absa-v1.1"
config.optimizer = "adamw"  # Optimizer class and str are both acceptable (from pytorch)
config.learning_rate = 0.00003
config.show_metric = True
config.batch_size = 32
config.patience = 2
config.log_step = 50
```

```
config.dropout = 0.5
config.seed = seeds
config.lcf = 'cdw'
config.num_epoch = 2
config.verbose = True  # If verbose == True, PyABSA will output the model structure and several processed data examples
config.notice = "This is a training example for aspect term extraction"
config.eta = -1  # Ensure eta is set to a valid value
MV = MetricVisualizer('model')
config.MV = MV
```

Листинг 7: Функция потери

```
from torch.nn import nn.MSELoss
#rest of the code is hidden here
if labels is not None:
            #Instade of CrossEntropyLoss(ignore_index=0) we use MSE:
            criterion_ate = MSELoss(ignore_index=0)
            criterion_apc =MSELoss(
                ignore_index=LabelPaddingOption.SENTIMENT_PADDING
            )
            loss_ate = criterion_ate(
                ate_logits.view(-1, self.num_labels), labels.view(-1)
            )
            loss_apc = criterion_apc(apc_logits, polarity)
            return loss_ate, loss_apc
        else:
            return ate_logits, apc_logits
```

Для визуализации различных графиков, можно использовать встроенную библиотеку Metric Visualizer. К сожалению, Metric Visualizer немного ограничена по функциональности и документация старая и неактуальная. Например, в документации название графика «trajectory_plot» написано так: «traj_plot». И все остальные графики в документации имеют такой вид: «название_plot_by_trail». В процессе построения графиков, было потрачено много времени именно из-за устаревшей документации.Одно из преимуществ Metric Visualizer то, что она создает файлы в формате .mv для каждой модели и их можно использовать отдельно для пострения графиков. Сущесвует вероятность, что команда MV.summary будет работать нормально, а остальные необходимо будет исправить вручную.

Листинг 8: Metric Visualizer

```
config.MV.summary(save_path=None, xticks=max_seq_lens)
config.MV.trajectory_plot(save_path=None, xticks=max_seq_lens)
config.MV.violin_plot(save_path=None, xticks=max_seq_lens)
config.MV.box_plot(save_path=None, xticks=max_seq_lens)
```

2.3 Полученные результаты и их анализ

На тестовой выборке набора данных finNews максимальная точность модели достигла 87.38%. Это была модель с

$$max\_seq\_len = 80.$$

Metric Visulizer дает возможность получить отчет об оценках задачи извлечения сущностей ATE (см. табл. 2, 3, 4). Микро среднее значение (micro average)

вычисляет среднее значение путем подсчета общего количества истинно положительных (True Positive, TP), ложноотрицательных (False Negativity, FN) и ложноположительных (False Positive, FP) предсказаний. Используем формулы
для показателя точности:

$$Precision = \frac{TP_{positive} + TP_{negative} + TP_{neutral}}{TP_{positive} + FP_{positive} + TP_{negative} + FP_{negative} + TP_{neutral} + FP_{neutral}},$$

для показателя полноты:

$$Recall = \frac{TP_{positive} + TP_{negative} + TP_{neutral}}{TP_{positive} + FN_{positive} + TP_{negative} + FN_{negative} + TP_{neutral} + FN_{neutral}},$$

Макро среднее значение (macro average) вычисляет среднее значение для каждой метки и находит их невзвешенное среднее значение. При этом не учитывается дисбаланс между классами. Рассчитаем количество истинно положительных предсказаний (TP), ложноположительных предсказаний (FP) и ложноотрицательных предсказаний (FN) для каждого класса (всего три класса). Вычисляем точность и полноту для каждого класса следующим образом:

$$\frac{TP}{TP + FP}$$

и

$$\frac{TP}{TP + FN},$$

где N - это число всех предсказаний.

$$Precision = \frac{Precision_{positive} + Precision_{negative} + Precision_{neutral}}{N},$$

$$Recall = \frac{Recall_{positive} + Recall_{negative} + Recall_{neutral}}{N}.$$

Средневзвешенное значение (weighted average) вычисляет показатели для каждого класса и находит их среднее значение по поддержке/support (число истинных предсказаний для каждого класса). Это изменяет "макро"значение (macro) для учета дисбаланса между классами; это может привести к получению оценки F, которая не лежит в пределах точности и полноты.

Таблица 2: классификация ATE; max_seq_len = 60

	precision	recall	f1-score
micro avg	0.8895	0.9549	0.9210
macro avg	0.6261	0.6366	0.6313
weighted avg	0.9392	0.9549	0.9469

Таблица 3: классификация ATE; max_seq_len = 80

	precision	recall	f1-score
micro avg	0.9010	0.9651	0.9320
macro avg	0.9011	0.9651	0.9317
weighted avg	0.9011	0.9651	0.9317

Таблица 4: классификация ATE; max_seq_len = 100

	precision	recall	f1-score
micro avg	0.9109	0.9665	0.9379
macro avg	0.6413	0.6443	0.6428
weighted avg	0.9620	0.9665	0.9642

Модель с максимальной последовательностью из 100 токенов показала наилучшие результаты micro average (F1-score = 0.9379) и weighted average (F1-score = 0.9642). Данная модель обладает высокой точностью, но низкое значение macro avg говорит о сильной склонности к классу 'O' (класс 'O' - слова, которые не являются аспектами). То есть, модель хорошо классифицирует слова, принадлежащие к классу 'O', но плохо классифицирует слова принадлежащие к другим классам. В модели с максимальной последовательностью из 80 токенов микро-, макро- и средневзвешенные значения (micro, macro и weighted averages) практически идентичны, что позволяет предположить минимальную погрешность и сбалансированную классификацию по классам. Эта модель демонстрирует наилучший баланс, поскольку склонение ко всем классам одинаковое.

На рисунке 4 видно, что распределение точности на тестовой выборке в задаче классификации эмоциональной окраски текста APC (синий цвет, см. рис. 4) является относительно компактным с центром значений около 86-87%. Аналогично, распределение F1-меры по задаче APC является относительно компактным с центром значений около 85-86%. Оба распределения указывают на стабильную производительность модели для задачи APC, с минимальной изменчивостью, что нельзя сказать о распределении F1 модели для задачи ATE, у которой распределение значительно шире, что свидетельствует о большей изменчивости в производительности. Центр распределения находится около 86%, но значительный разброс указывает на то, что производительность может значительно варьироваться по сравнению с метриками APC. Если рассматривать только точности трех моделей, получаем следующие значения (5):

Таблица 5: Максимальная точность модели в задаче ATE_APC

max_seq_len	точность
60	86.40%
80	87.38%
100	87.30%

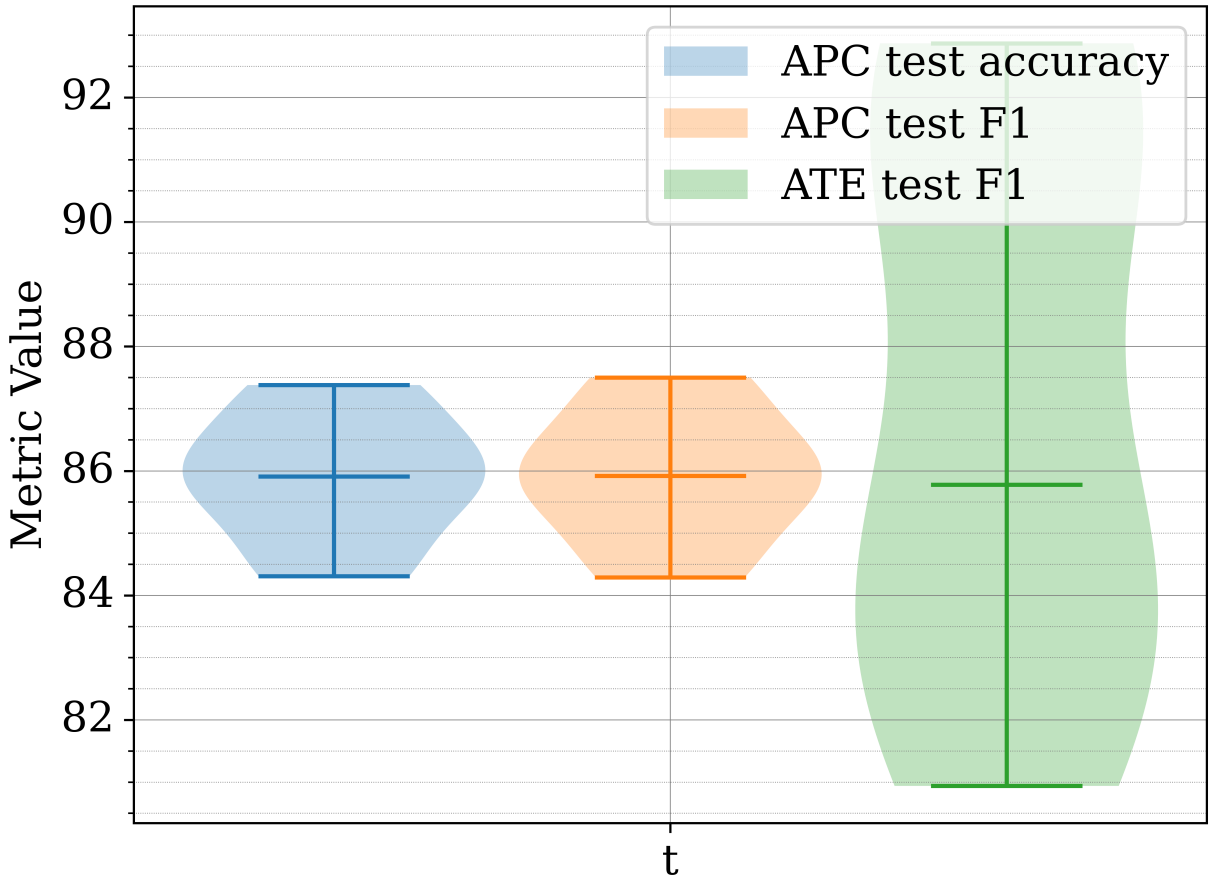

Рис. 4: Скрипичный график всех моделей на одном графике

Фреймворк pyABSA позволяет нам сохранить модель после обучения. При помощи метода predict можно предсказать оценку тональности и объект этой тональности. К примеру, используем следующее предложение «Apple stocks went up by 10 % on Monday, according to TASS». Это предложение сложное по структуре, так как в нем два объекта тональности Apple и TASS. Модель выводит следующую информацию:

Листинг 9: Predication

```
[{"sentence": "Apple stocks went up by 10 % on Monday , according to TASS .",
"IOB": ["B-ASP", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O"],
"tokens": ["Apple", "stocks", "went", "up", "by", "10", "%", "on", "Monday", ",", "
    according", "to", "TASS", "."],
"aspect": ["Apple"],
"position": [[0]],
"sentiment": ["Positive"],
"probs": [[0.008858616463840008, 0.16175325214862823, 0.8293880820274353]],
"confidence": [0.8294]}]
```

Модель смогла только правильно определить первую сущность Apple и правильнную тональность относительно Apple, но не смогла определить вторую сущность TASS. Это один из недостатков модели, она плохо распознает и извлекает сущности в тексте, если их больше одной. Иногда не распознает даже ни одну из сущностей. Например, в «Saudi's Biggest Listing of the

Year Fakeeh Rises in Riyadh Debut» Saudi, Fakeeh Rises и Riyadh Debut - это сущности в тексте модель не смогла извлечь даже одну из них.

Листинг 10: No Predication

```
[{"sentence": "Saudi ' s Biggest Listing of the Year Fakeeh Rises in Riyadh Debut",
"IOB": ["O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O"],
"tokens": ["Saudi", "'", "s", "Biggest", "Listing", "of", "the", "Year", "Fakeeh", "
    Rises", "in", "Riyadh", "Debut"],
"aspect": [],
"position": [],
"sentiment": [],
"probs": [],
"confidence":
[]}]
```

Листинг 11: No Predication

```
[{"sentence": "Treasury rubbishes Rishi Sunak ' s £ 2 , 000 tax hike election TV debate
     claim", "IOB": ["O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "
    O", "O", "O"], "tokens": ["Treasury", "rubbishes", "Rishi", "Sunak", "'", "s", "£",
     "2", ",", "000", "tax", "hike", "election", "TV", "debate", "claim"],
"aspect": [],
"position": [],
"sentiment": [],
"probs": [],
"confidence": []}]
```

```
[{"sentence": "Russia Oil Revenue Rose 50 % in May as Nation Adapts to Sanctions",
"IOB": ["O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O"],
"tokens": ["Russia", "Oil", "Revenue", "Rose", "50", "%", "in", "May", "as", "Nation",
    "Adapts", "to", "Sanctions"],
"aspect": [],
"position": [],
"sentiment": [],
"probs": [],
"confidence": []}]
```

Модель также не может распознать сарказм и шутки потому, что в финансовых новостях шутки как правило не используются, а, соответственно, она не склонна их извлекать в принципе.

```
[{"sentence": "Market Rejoices as Investors Discover New Trend : Losing Money Faster
    Than Ever Before !",
"IOB": ["O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O"],
"tokens": ["Market", "Rejoices", "as", "Investors", "Discover", "New", "Trend", ":", "
    Losing", "Money", "Faster", "Than", "Ever", "Before", "!"],
"aspect": [],
"position": [],
"sentiment": [],
"probs": [],
"confidence": []}]
```

Помимо вышесказанного, особенность fine_tuned модели BERT - плохая работоспособность на данных, на которых модель не была дополнительно обучена. Этот феномен называется Model Sensitivity в сфере NLP и трансформеров.

3 Выводы

За период практики, которую я проходил на кафедре математического моделирования и искусственного интеллекта факультета физико-математических и естественных наук РУДН, были достигнуты все цели и решены все поставленные задачи, определенные в программе преддипломной практики направления подготовки 02.03.01 «Математика и компьютерные науки» (см. введение отчета по практике). При прохождении практики я разобрался с научной терминологией области исследований; научился осуществлять сбор, анализ и обработку данных, необходимых для решения профессиональных задач; собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям; осуществлять целенаправленный поиск информации о новейших научных и технологических достижениях в сети Интернет и из других источников; строить и анализировать математические модели объекта исследований; разрабатывать и отлаживать вспомогательные программные комплексы; проводить численный эксперимент; оформлять результаты своих исследований. Также я овладел необходимым математическим и программным аппаратом исследований; навыками математического моделирования, применения численных методов для выполнения необходимых расчетов и получения численных оценок по теме исследований. В результате прохождения данной практики я приобрел следующие практические навыки, умения, универсальные и профессиональные компетенции:

- способностью к самоорганизации и к самообразованию;
- способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
- способностью к самостоятельной научно-исследовательской работе;
- способностью находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем;
- способностью к определению общих форм и закономерностей отдельной предметной области;
- способностью математически корректно ставить естественнонаучные задачи, знание постановок классических задач математики;
- способностью строго доказать утверждение, сформулировать результат, увидеть следствия полученного результата;
- способностью публично представлять собственные и известные научные результаты;

- способностью использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач;
- способностью передавать результат проведенных физико-математических и прикладных исследований в виде конкретных рекомендаций, выраженной в терминах предметной области изучавшегося явления;
- способностью представлять и адаптировать знания с учетом уровня аудитории;
- способностью к проведению методических и экспертных работ в области математики.

Выполненный во время проведения преддипломной практики обзор публикаций научных изданий как по теме Обработка естественного языка NLP, так и по теме анализа тональности текста и по теме технологии трасормеров таких как BERT от Google и GPT от Open AI, позволит мне обосновать актуальность выбранной темы, а также более полно раскрыть постановку задачи и методы исследования при написании выпускной квалификационной работы бакалавра.

Список литературы

1. A Practical Guide to Sentiment Analysis / E. Cambria [и др.]. — 1st. — Springer Publishing Company, Incorporated, 2017. — с. 196. — ISBN 3319553925.
2. Attention is All you Need / A. Vaswani [и др.] // Advances in Neural Information Processing Systems. т. 30 / под ред. I. Guyon [и др.]. — Curran Associates, Inc., 2017. — URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
3. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin [и др.] // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) / под ред. J. Burstein, C. Doran, T. Solorio. — Minneapolis, Minnesota : Association for Computational Linguistics, 06.2019. — с. 4171—4186. — DOI: 10.18653/v1/N19-1423. — URL: https://aclanthology.org/N19-1423.
4. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin [и др.] // North American Chapter of the Association for Computational Linguistics. — 2019. — URL: https://api.semanticscholar.org/CorpusID:52967399.
5. Improving language understanding by generative pre-training / A. Radford [и др.]. — 2018.
6. LCF: A Local Context Focus Mechanism for Aspect-Based Sentiment Classification / B. Zeng [и др.] // Applied Sciences. — 2019. — URL: https://api.semanticscholar.org/CorpusID:202095577.
7. *Liu B.* Sentiment analysis and subjectivity //. — 01.2010. — с. 627—666.
8. *Nitin Mehrotra E. U.* Microsoft Learn. — 2024. — URL: https://learn.microsoft.com/ru-ru/azure/ai-services/language-service/named-entity-recognition/overview ; дата обращения: 03.06.2024.
9. *Samigulin T., Djurabaev A.* SENTIMENT ANALYSIS OF TEXT BY MACHINE LEARNING METHODS // Research result. Information technologies. — 2021. — март. — т. 6, № 1. — ISSN 2518-1092. — DOI: 10.18413/2518-1092-2021-6-1-0-7. — URL: http://dx.doi.org/10.18413/2518-1092-2021-6-1-0-7.
10. SemEval-2014 Task 4: Aspect Based Sentiment Analysis / M. Pontiki [и др.] // Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) / под ред. P. Nakov, T. Zesch. — Dublin, Ireland : Association for Computational Linguistics, 08.2014. — с. 27—35. — DOI: 10.3115/v1/S14-2004. — URL: https://aclanthology.org/S14-2004.

11. SemEval-2015 Task 12: Aspect Based Sentiment Analysis / M. Pontiki [и др.] // Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) / под ред. P. Nakov [и др.]. — Denver, Colorado : Association for Computational Linguistics, 06.2015. — с. 486—495. — DOI: 10.18653/v1/S15-2082. — URL: https://aclanthology.org/S15-2082.
12. SemEval-2016 Task 5: Aspect Based Sentiment Analysis / M. Pontiki [и др.] // Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016) / под ред. S. Bethard [и др.]. — San Diego, California : Association for Computational Linguistics, 06.2016. — с. 19—30. — DOI: 10.18653/v1/S16-1002. — URL: https://aclanthology.org/S16-1002.
13. spaCy. — URL: spacy.io ; дата обращения: 03.06.2024.
14. *Yang H.*, *Li K.* PyABSA: Open Framework for Aspect-based Sentiment Analysis // CoRR. — 2022. — т. abs/2208.01368. — DOI: 10.48550/arXiv.2208.01368. — arXiv: 2208.01368. — URL: https://doi.org/10.48550/arXiv.2208.01368.