Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

Магистерская программа Финансовые технологии и анализ данных

КУРСОВАЯ РАБОТА

На тему (рус.) «Обработка текстов онлайн социальных сетей с использованием нейросетевых методов»

На тему (англ.) «Online Social Network Text Analysis Using Neural Network Methods»

Студент

Янаков Дмитрий Спартакович

_____

(подпись)

Руководитель КР

ст. преп.
ДАДИИ ФКН НИУ ВШЭ
Карпов Илья Андреевич

_____

(подпись)

Москва, 2024

Аннотация

Курсовая работа посвящена исследованию методов обработки текстов онлайн социальных сетей с использованием нейросетевых моделей на примере задачи классификации сентимента постов в Twitter.

Цель данной работы заключается в улучшении качества классификации с использованием модели BERT (Bidirectional Encoder Representations from Transformers) и дополнительного знания о пользователе, полученного различными алгоритмами извлечения числовой информации из текстовой. Модель BERT была выбрана ввиду ее способности эффективно обрабатывать контекстную информацию в тексте, что делает ее одной из наиболее передовых моделей для задач обработки естественного языка (NLP).

Методология исследования включает несколько этапов. На первом этапе проводится предварительная обработка данных, включающая удаление шумовых элементов и нормализацию текста. На втором этапе осуществляется обучение модели BERT на собранных данных, после чего производится внедрение (интеграция) дополнительной информации о пользователях. Для оценки качества классификации используется F1-мера, которая учитывает как точность, так и полноту классификации.

Результаты работы демонстрируют, что добавление пользовательских данных позволяет улучшить качество классификации сентимента постов по сравнению с базовой моделью BERT. Полученные результаты подчеркивают важность использования дополнительного контекста при анализе текстов в социальных сетях.

Содержание

Введение

Задача классификации сентимента в обработке естественного языка является одной из ключевых и востребованных в современных приложениях [15]. Сентимент-анализ позволяет автоматизировать процесс определения эмоциональной окраски текста, что находит широкое применение в различных областях, таких как маркетинг, социальные сети, анализ отзывов и т.д. В условиях огромных объемов текстовых данных, генерируемых ежедневно, автоматизация анализа сентимента становится не только актуальной, но и необходимой для эффективного управления информацией.

Современные модели NLP [3,13], такие как BERT [4], демонстрируют высокую точность в задачах классификации текста благодаря своей способности учитывать контекст слов в предложении. Однако, несмотря на значительные успехи, существует потенциал для дальнейшего улучшения качества классификации за счет интеграции дополнительной информации об авторе текста [5]. Это особенно важно в случае анализа данных социальных сетей, где личностные характеристики пользователя могут существенно влиять на интерпретацию текста.

Целью данной курсовой работы является улучшение качества классификации сентимента текстов пользователей (твитов) в социальной сети Twitter путем интеграции дополнительной информации о пользователях в модель BERT.

Для достижения этой цели были поставлены следующие задачи:

1. Исследовать методы получения векторных представлений (эмбеддингов) пользователей на основе их твитов с использованием моделей LDA (Latent Dirichlet Allocation) [2], GloVe (Global Vectors for Word Representation) [10], Word2Vec [8] и BERT.

2. Разработать подходы к интеграции дополнительной информации в модель BERT.

3. Провести экспериментальное сравнение предложенных методов интеграции с точки зрения их влияния на качество классификации сентимента.

4. Оценить результаты и определить наиболее эффективный метод интеграции.

Новизна работы заключается в разработке и экспериментальной оценке методов интеграции дополнительной информации о пользователях в модель BERT для задачи классификации. Предложенные подходы включают:

1. Использование тематического моделирования (LDA) для получения распределения тем в твитах пользователей.

2. Применение алгоритмов GloVe, Word2Vec и модели BERT для создания векторных представлений пользователей на основе их твитов.

3. Интеграция дополнительной информации о пользователе в модель BERT двумя способами: добавление информации непосредственно в текст твита и конкатенация эмбеддинга пользователя с эмбеддингом [CLS] токена, получаемого на выходе BERT.

Для обеспечения достоверности результатов все эксперименты были задокументированы в виде ipynb ноутбуков, которые содержат код и результаты каждого этапа исследования. Код ноутбуков выложен на GitHub и доступен по ссылке https://github.com/yanakidis/bert_with_injection. Это позволяет повторить эксперименты и проверить полученные результаты.

Теоретическая значимость работы заключается в расширении знаний о методах интеграции дополнительной информации в модели трансформеров для задач NLP. Результаты исследования могут быть использованы для дальнейшего развития моделей обработки естественного языка и улучшения их точности за счет учета контекста пользователя.

Практическая ценность работы состоит в возможности применения разработанных методов для улучшения качества анализа сентимента в реальных приложениях, таких как мониторинг социальных сетей, анализ отзывов клиентов и автоматизация взаимодействия с пользователями. Полученные результаты могут быть полезны для компаний и организаций, занимающихся анализом больших объемов текстовых данных, а также для исследователей и разработчиков в области NLP.

1. Обзор используемых методов

Идея использования моделей LDA, Word2Vec, GloVe и BERT для получения эмбеддингов пользователей заимствована из [9], где также присутствуют и альтернативные методы.

1. Latent Dirichlet Allocation (LDA)

LDA — это статистический метод тематического моделирования, который используется для обнаружения скрытых тем в коллекции документов. Основная идея заключается в том, что каждый документ представляет собой совокупность различных тем, а каждая тема — это распределение вероятностей по словам.

LDA использует байесовский подход для моделирования документов. Он предполагает, что:

1. Каждое слово в документе связано с одной из тем.
2. Каждая тема представлена распределением слов.
3. Каждый документ представлен распределением тем.

LDA широко используется для автоматической классификации текстов, анализа мнений, информационного поиска и других задач, связанных с обработкой текста.

В данной работе модель LDA будет использована для получения эмбеддингов пользователей двумя следующими способами:

1. **User-LDA**. Все твиты одного конкретного пользователя рассматриваются как один большой документ и подаются на вход в модель LDA. Полученное распределение тем в этом документе считается эмбеддингом пользователя.
2. **Post-LDA**. Каждый твит одного конкретного пользователя рассматривается как отдельный документ и подается на вход в модель LDA. Далее для каждого пользователя распределения тем во всех его твитах усредняются. Полученный вектор считается эмбеддингом пользователя.

2. Word2Vec

Word2Vec — это метод представления слов в виде векторов чисел. Основная идея заключается в том, чтобы обучить нейронную сеть таким образом, чтобы слова с похожим контекстом имели схожие векторные представления.

Основными моделями Word2Vec являются:

1. Continuous Bag of Words (CBOW). Предсказывает текущее слово по его контексту (окружающим словам).

2. Skip-gram. Предсказывает контекстные слова по текущему слову.

Обученные эмбеддинги могут использоваться для различных задач NLP, таких как классификация текста, кластеризация и машинный перевод.

В данной работе рассматривается модель CBOW. Она используется для получения эмбеддингов пользователей путем усреднения эмбеддингов всех слов, используемых им.

3. Global Vectors for Word Representation (GloVe)

GloVe — это метод обучения векторных представлений слов на основе матрицы совместной встречаемости слов в корпусе текстов. В отличие от Word2Vec, который фокусируется на локальном контексте, GloVe использует глобальную статистику текста.

GloVe обучает эмбеддинги таким образом, чтобы отношения между словами отражали их глобальные статистические свойства. Это достигается путем минимизации функции потерь, которая учитывает частоту совместной встречаемости слов.

Получение эмбеддингов для пользователей при использовании GloVe аналогично Word2Vec.

4. BERT

BERT — это одна из самых популярных и мощных моделей для обработки естественного языка. Она основана на архитектуре трансформеров, предложенной в [14]. Трансформеры используют механизмы внимания для обработки текста, что позволяет моделям учитывать контекст слова с обеих сторон (слева и справа), в отличие

от предыдущих моделей, таких как Word2Vec или GloVe, которые являются однонаправленными (моделируют контекст либо слева, либо справа). Архитектура BERT использует только энкодерную часть трансформера, которая будет подробнее рассмотрена в теоретической части.

BERT можно использовать как для получения эмбеддингов слов, так и для различных задач NLP путем дообучения на конкретных данных:

1. Классификация текста.

2. Named Entity Recognition (NER). BERT можно использовать для выделения именованных сущностей в тексте, таких как имена людей, названия мест и т.д.

3. Вопросно-ответные системы.

5. Многослойный перцептрон (Multilayer Perceptron, MLP)

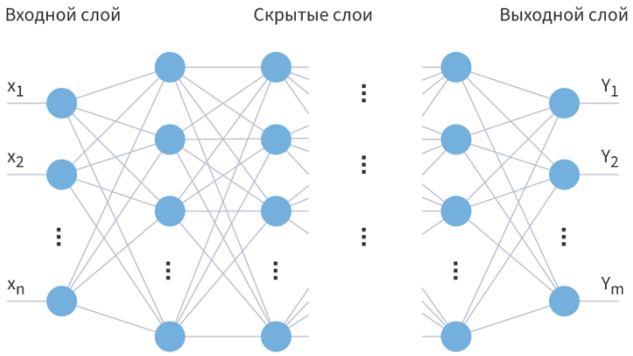


Рисунок 1. Многослойный перцептрон.

Многослойный перцептрон [11] — это тип нейронной сети, который состоит из нескольких слоев нейронов и используется для решения различных задач машинного обучения. Рассмотрим его архитектуру (см. Рис. 1) и применение более подробно.

Входной слой состоит из нейронов, которые принимают входные данные. Количество нейронов во входном слое соответствует размерности входных данных. Между входным и выходным слоями находятся один или несколько скрытых слоев. Каждый скрытый слой состоит из множества нейронов, которые связаны с нейронами предыдущего и следующего слоя. Нейроны в скрытых слоях применяют нелинейные активационные функции (например, ReLU [1], сигмоида, тангенс гиперболический), что позволяет сети моделировать сложные нелинейные зависимости. Выходной слой состоит из одного или нескольких нейронов, в зависимости от задачи. Например, в задачах классификации количество нейронов в выходном слое соответствует числу

классов, и часто используется функция активации SoftMax для получения вероятностей классов.

Принцип работы MLP заключается в следующем: каждый нейрон вычисляет взвешенную сумму входных сигналов и применяет к ней функцию активации. Результат передается на следующий слой и т.д.

Для обучения многослойного перцептрона используется метод обратного распространения ошибки (backpropagation), который корректирует веса связей между нейронами. Метод использует градиентный спуск для минимизации функции потерь.

В данной работе MLP будет применяться для интеграции дополнительного знания в BERT. Подробнее про это будет описано в теоретической части.

2. Теоретическая часть

Рассмотрим, как устроена модель BERT и как в нее можно внедрить дополнительное знание.

1. BERT

BERT, как было упомянуто ранее, использует архитектуру трансформеров (см. Рис. 2), которая состоит из двух основных компонентов: энкодера и декодера. Однако BERT использует только энкодерную часть трансформера, которая состоит из механизма внимания (Multi-Head Self-Attention), а также из полносвязных слоев (Feed-Forward Neural Network). Механизм внимания позволяет модели учитывать различные части входного предложения одновременно, что помогает захватить сложные зависимости между словами.

В BERT важную роль играют токены, так как они позволяют модели понимать и обрабатывать текст.

Есть несколько типов токенов:

1. токены слов;
2. специальные токены – **[CLS]**, **[SEP]**, **[MASK]**.

BERT использует метод токенизации WordPiece [12], который разбивает слова на подслова или токены. Это позволяет модели эффективно работать с редкими или

неизвестными словами, разбивая их на более частые компоненты. Например, слово "unhappiness" может быть разбито на "un", "##happiness". Префикс "##" указывает на то, что это часть слова, а не отдельное слово.

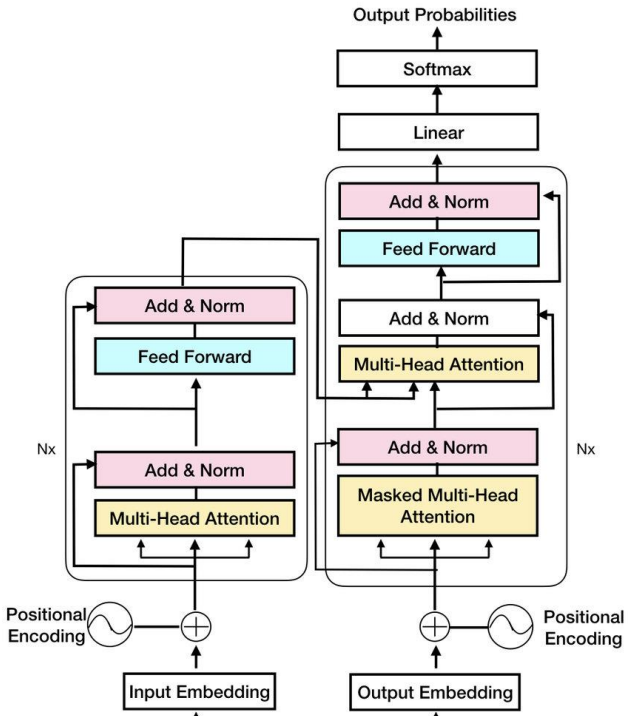


Рисунок 2. Архитектура трансформера. Слева - энкодер, справа - декодер.

[CLS] – это специальный токен, добавляемый в начало каждого входного текста. Выходной вектор, соответствующий этому токену, часто используется как представление всего входного текста и особенно полезен для задач классификации. Токен **[SEP]** используется для разделения различных частей текста. Применяется в задачах, где нужно обрабатывать пару предложений (например, задачи предсказания следующего предложения или задачи вопрос-ответ). Токен **[MASK]** используется в процессе обучения для маскирования слов.

Для обучения BERT используется две задачи: первая – метод маскированного языкового моделирования (Masked Language Modeling, MLM), вторая – предсказание следующего предложения (Next Sentence Prediction, NSP). В первом случае в процессе обучения случайные слова в предложении заменяются токеном [MASK], и задача модели — предсказать эти замаскированные слова на основе контекста. Это позволяет модели учиться захватывать контекстные зависимости между словами. Во втором случае модель получает пару предложений и должна определить, является ли второе предложение логическим продолжением первого. Эта задача помогает модели лучше понимать отношения между предложениями.

В данной работе будет рассмотрена модель **BERT-base-uncased**, которая состоит из 12 слоев энкодера трансформера, предобученная на наборе данных BookCorpus, состоящем из текстов около 11 000 неопубликованных книг, извлеченных из Интернета, а также на данных английской Википедии. «Uncased» означает, что текст был приведен

к нижнему регистру перед обучением. Например, слова "Apple" и "apple" будут рассматриваться как одно и то же слово.

2. Внедрение дополнительного знания в BERT

В данной работе будет рассмотрено три способа интеграции дополнительного знания в BERT:

1. Добавление дополнительной информации в текст твита.

2. Конкатенация эмбеддинга [CLS] токена с эмбеддингом пользователя и подача объединенного вектора на вход в MLP.

3. Комбинация 1 и 2.

Первый способ был предложен в [16] и его идея заключается в том, чтобы добавить дополнительную информацию о тексте, который требуется классифицировать, через специальный токен [SEP]. В экспериментальной части будет рассмотрено 4 модели, в каждой из которых будет использоваться разная информация (см. Таблицу 3).

Второй способ использует эмбеддинг пользователя, полученный с помощью одного из алгоритмов, описанных в части 1, а также эмбеддинг [CLS] токена, получаемый на выходе BERT. Далее два этих вектора конкатенируются и подаются на вход в MLP (см. Рис. 3).

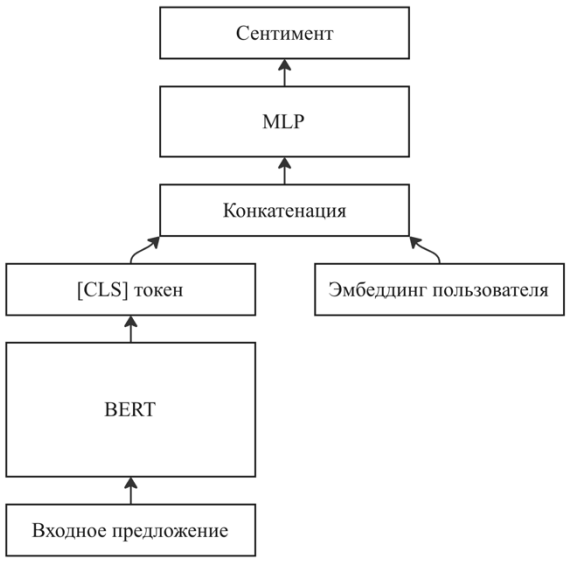


Рисунок 3. Интеграция эмбеддинга пользователя в BERT.

В настоящей работе рассматривается архитектура многослойного перцептрона с 4 скрытыми слоями, размерность каждого из которых меньше предыдущего в 4 раза. Например, если размерность эмбеддинга пользователя равняется 20, а размерность эмбеддинга [CLS] токена равняется 768, то размерность итогового вектора равна 798, а MLP имеет 4 скрытых слоя с размерностями 197, 49, 12 и 3.

3. Обзор альтернативных методов внедрения дополнительного знания

1. K-BERT

В статье [6] представлена модель K-BERT (см. Рис. 4), которая интегрирует знания из так называемых графов знаний (Knowledge Graphs, KGs) в процесс обучения языковых моделей на основе BERT.

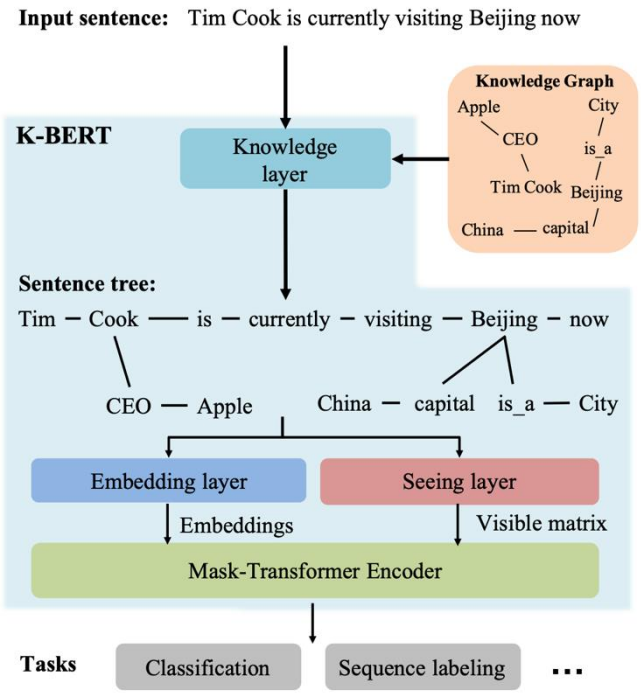


Рисунок 4. Структура модели K-BERT.

Основная идея заключается в том, чтобы улучшить представление языка, добавляя информацию о различных сущностях и их взаимосвязях. Например, в графе знаний может быть информация о том, что "Tim Cook is CEO of Apple".

В процессе предобработки текста K-BERT извлекает сущности из текста и находит соответствующие триплеты в графе знаний. Триплеты представляют собой отношения вида (субъект, предикат, объект), например, ("Tim Cook", "CEO", "Apple"). Эти триплеты затем внедряются в исходный текст, создавая расширенные последовательности, которые содержат как оригинальные слова, так и дополнительные знания.

Для обучения модели используется механизм маскирования, аналогичный BERT, но с учетом дополнительных триплетов. Модель обучается на предсказание как оригинальных слов, так и связанных с ними сущностей из графа знаний. Это позволяет ей лучше понимать контекст и семантику текста за счет дополнительной информации.

2. TwHIN-BERT

В статье [17] предложена модель TwHIN-BERT (см. Рис. 5). Вводится так называемая Twitter Heterogeneous Information Network (TwHIN) (см. Рис. 6), в которой содержится информация о пользователях и их взаимодействиях с твитами: лайк

(означает, что твит понравился пользователю), комментарий (пользователь оставил комментарий под твитом) или репост (пользователь опубликовал у себя такой же твит).

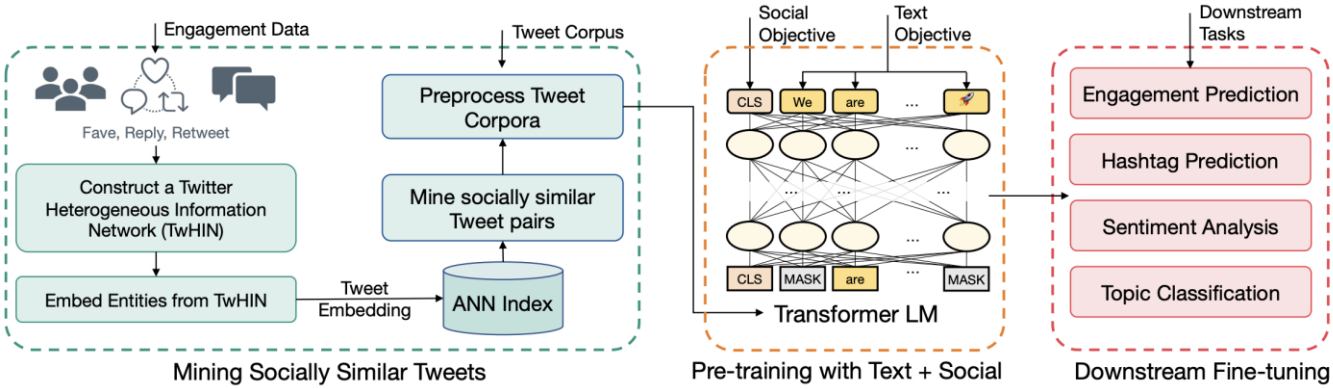


Рисунок 5. Процесс создания TwHIN-BERT. Включает три этапа: 1) поиск социально схожих твитов с помощью TwHIN, 2) обучение модели, 3) дообучение для последующих задач

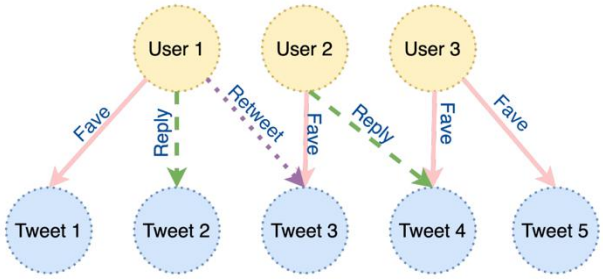


Рисунок 6. Twitter Heterogeneous Information Network

Основная идея заключается в том, что авторы сначала получают эмбеддинги твитов с помощью TwHIN, далее с помощью полученных эмбеддингов ищут социально схожие пары твитов, то есть такие, которые скорее всего будет взаимодействовать с одними и те ми же пользователями, а затем на полученных парах обучают BERT. Для обучения используются две задачи – MLM, как в случае с базовой версией, и задача, позволяющая модели выучить, являются ли два твита социально схожими или нет (в статье для этого вводится название Contrastive Social Loss). После этого модель может быть дообучена для последующих задач, например, классификации текста.

4. Экспериментальное исследование

1. Подготовка данных

Для проведения экспериментов был выбран набор данных **Sentiment140**, который содержит 1.6 млн. англоязычных твитов, их сентимент и автора. Сентимент в данном случае принимает два значения: положительный или отрицательный. Из этого датасета было отобрано 100 тыс. записей, соответствующих пользователям с наибольшим количеством твитов. Обработка текстов для последующей их классификации состояла из следующих этапов:

1) приведение текста к нижнем регистру;
2) удаление ссылок;
3) удаление упоминаний других авторов в тексте твита вида @username;
4) удаление всех элементов текста, не являющихся словами или пробелами;
5) токенизация (разделение предложения на слова) полученного текста;
6) удаление стоп-слов, то есть таких слов, которые используются часто, однако не вносят никакой дополнительной информации в текст, а наоборот добавляют шум в него (например, «the», «is», «a»);
7) лемматизация оставшихся слов, то есть приведение слов к их начальным формам;
8) объединение получившихся слов в цельное предложение.

После таких преобразований тексты некоторых твитов могли стать пустыми, поэтому они также были удалены из рассмотрения. В итоге осталось **99 535** записей. Пример преобразования одного из твитов см. в Таблице 1.

Твит до	@GabrielSaporta i think maybe you need to go to the store and buy some food. or hit up bk and get your big mac on! wait...
Твит после	think maybe need go store buy food hit bk get big mac wait

Таблица 1. Один из твитов из набора данных до и после преобразований.

Для тестирования данные были разбиты на обучающую, валидационную и тестовую выборки в отношении 70/10/20. Распределение получившегося разбиения по классам указано в Таблице 2. Дисбаланс классов отсутствует.

	Обучающая	Валидационная	Тестовая
Положительный	43145	6164	12327
Отрицательный	26529	3790	7580

Таблица 2. Распределение выборок (столбцы) по сентиментам (строки), шт.

2. Получение эмбеддингов для пользователей

Эмбеддинги для пользователей были получены с помощью методов, указанных в части 1 настоящей работы: LDA, Word2Vec, GloVe и BERT:

- Для модели LDA были рассмотрены количества тем – 2, 5, 10 и 20.
- Для модели GloVe были выбраны предобученные модели на данных из 2 млрд. твитов с размерностями эмбеддингов 100 и 200.
- Модель Word2Vec была обучена на данных, выбранных для тестирования, с размерностью эмбеддинга 300.
- Для BERT, как упоминалось ранее, была выбрана модель BERT-base-uncased, у которой размер скрытого состояния каждого токена равняется 768. Следовательно, эмбеддинги твитов имели такую же размерность.

3. Тестируемые модели

В тестировании участвовали следующие четыре группы моделей:

1. **BERT** без каких-либо модификаций;

2. BERT, где в текст твита добавлена дополнительная информация;

3. BERT с использованием эмбеддингов пользователей (см. раздел 2 в теоретической части);

4. BERT, где в текст твита добавлена дополнительная информация, а также использован эмбеддинг пользователя.

Во второй группе было использовано 4 модели:

1. **BERT_TU**, в которой к тексту твита дополнительно добавлялось «[SEP] The author of the text is {username}», где {username} – это ник пользователя в Twitter.

2. **BERT_U**, в которой к тексту твита дополнительно добавлялось «[SEP] {username}».

3. **BERT_TS**, в которой к тексту твита дополнительно добавлялось «[SEP] The sentiment is {positive/negative}», где {positive/negative} – это реальный сентимент подаваемого на вход текста.

4. **BERT_S**, в которой к тексту твита дополнительно добавлялось «[SEP] {positive/negative}».

Примеры входных предложений для указанных моделей см. в Таблице 3.

Модель	**Предложение на вход (без предварительной обработки)**
BERT	[CLS] May is almost over. [SEP]
BERT_TU	[CLS] May is almost over. [SEP] The author of the text is Julia [SEP]
BERT_U	[CLS] May is almost over. [SEP] Julia [SEP]
BERT_TS	[CLS] May is almost over. [SEP] The sentiment is negative [SEP]
BERT_S	[CLS] May is almost over. [SEP] negative [SEP]

Таблица 3. Примеры конструкций входных предложений.

В третьей группе было использовано 12 моделей:

1. **BERT_{post_lda/user_lda}_{2/5/10/20}**, где {post_lda/user_lda} – это один из способов получения эмбеддинга для пользователя с помощью модели LDA. 2, 5, 10, 20 – количества рассматриваемых тем.

2. **BERT_glove_100**, **BERT_glove_200** и **BERT_w2v_300**, в которых эмбеддинги получены с помощью методов Word2Vec и GloVe. Число в конце названия модели указывает размерность эмбеддинга.

3. **BERT_bert**, где эмбеддинги пользователей получены с помощью самого BERT.

В четвертую группу вошли комбинации моделей BERT_TU, BERT_U и BERT_bert, так как они показали наилучшее качество на тестовой выборке (см. раздел с результатами) – **BERT_TU_bert** и **BERT_U_bert**.

4. Дообучение моделей

В качестве оптимизатора был выбран алгоритм AdamW [7] с learning rate (темпом обучения) равным 10^{-5}. Для настройки learning rate во время дообучения использовался scheduler (планировщик), который линейно уменьшал изначальный learning rate до 0.

Количество эпох дообучения равнялось 5 для моделей из первой и второй групп и 10 для моделей из третьей и четвертой групп (поскольку там присутствовал MLP, у которого изначально случайно сгенерированы веса), размер батча – 32. Функция потерь – кросс-энтропия.

Из графиков значений функции потерь от номера эпохи (см. Приложение 1) видно, что для моделей из первой и второй групп переобучение начинается после 2 эпохи, для третьей группы – с 9 эпохи. Для четвертой группы переобучение не наблюдается. Следовательно, замер качества на тестовой выборке будет проводиться на соответствующих эпохах (то есть для первой и второй группы – на 2 эпохе, для третьей – на 9, для четвертой – на 10).

5. Результаты

В качестве метрики точности классификации была выбрана F1-мера. Результаты на тестовой выборке для всех моделей приведены в Таблице 4. Из экспериментов можно сделать следующие выводы:

1. Добавление в текст твита информации об авторе позволило увеличить качество классификации почти на 3% (модели BERT_TU и BERT_U).

2. Метод получения эмбеддингов Post_LDA показал качество лучше, чем User_LDA. Модели BERT_post_lda_5 и BERT_post_lda_10 улучшили качество по сравнению с исходной моделью BERT.

3. Из всех моделей, где производилась только конкатенация эмбеддинга [CLS] токена с эмбеддингом пользователя (вторая и третья группы), наилучшее качество показала модель, в которой эмбеддинг пользователя получался с помощью BERT (BERT_bert).

4. Прямой зависимости между увеличением размерности вектора эмбеддинга и качеством классификации не наблюдается, однако получение эмбеддинга пользователя с помощью GloVe показало качество лучше, чем с помощью LDA. Низкое качество BERT_w2v_300, возможно, связано с тем, что модель Word2Vec обучалась только на корпусе рассматриваемого набора данных (Sentiment140), в отличие от GloVe, которая была предобучена на большом массиве данных.

BERT 84.47			
BERT_TU 87.20	**BERT_U** 87.40	**BERT_TS** 84.46	**BERT_S** 84.12
BERT_post_lda_2 84.34	**BERT_post_lda_5** 84.68	**BERT_post_lda_10** 84.70	**BERT_post_lda_20** 84.47
BERT_user_lda_2 84.28	**BERT_user_lda_5** 84.46	**BERT_user_lda_10** 84.21	**BERT_user_lda_20** 84.21
BERT_glove_100 85.11	**BERT_glove_200** 84.98	**BERT_w2v_300** 84.23	**BERT_bert** 85.15
BERT_TU_bert 87.29		**BERT_U_bert** 86.94	

Таблица 4. Результаты тестирования на тестовой выборке (F1-мера), %

5. Модель BERT_TU_bert улучшила качество BERT_TU, однако в BERT_U_bert произошло снижение качества.

Таким образом, рассмотренные методы по интеграции дополнительного знания позволили улучшить качество классификации по сравнению с базовой моделью BERT.

Заключение

В настоящей работе рассмотрена задача классификации сентимента текстов на примере постов в социальной сети Twitter. В качестве основной модели для классификации используется модель BERT. Исследуются подходы к внедрению дополнительной информации об авторе текста в модель для улучшения качества классификации (F1-мера):

1. Добавление информации о пользователе в текст твита.

2. Конкатенация эмбеддинга [CLS] токена с эмбеддингом пользователя и подача на вход в многослойный перцептрон.

Результаты экспериментов показывают, что оба предложенных метода позволяют улучшить качество классификации по сравнению с базовой моделью BERT. Это указывает на значимость учета дополнительной информации о пользователе при обработке текстов в социальных сетях и открывает новые перспективы для дальнейших исследований:

- поиск дополнительной информации о пользователе, которая может оказать положительное влияние на качество классификации;

- исследование различных архитектур MLP для оптимального комбинирования текстовых и пользовательских эмбеддингов;

- анализ параметров дообучения модели BERT с учетом специфики данных социальных сетей.

Таким образом, проведенное исследование демонстрирует потенциал использования нейросетевых методов для улучшения обработки текстов в социальных сетях и подчеркивает важность комплексного подхода к анализу данных, включающего как текстовую, так и метаинформацию о пользователях.

Список литературы

[1] *Abien Fred Agarap*. (2018). Deep Learning using Rectified Linear Units (ReLU).

[2] *Blei David, Ng Andrew & Jordan Michael*. (2001). Latent Dirichlet Allocation. The Journal of Machine Learning Research. 3. 601-608.

[3] *Bubeck Sébastien, Chandrasekaran Varun, Eldan Ronen, Gehrke Johannes, Horvitz Eric, Kamar Ece, Lee Peter, Lee Yin Tat, Li Yuanzhi, Lundberg Scott, Nor Harsha, Palangi Hamid, Ribeiro Marco & Zhang Yi*. (2023). Sparks of Artificial General Intelligence: Early experiments with GPT-4.

[4] *Devlin Jacob, Chang Ming-Wei, Lee Kenton & Toutanova Kristina*. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

[5] *Karpov Ilia & Kartashev Nick*. (2021). SocialBERT -- Transformers for Online SocialNetwork Language Modelling.

[6] *Liu Weijie, Zhou Peng, Zhao Zhe, Wang Zhiruo, Ju Qi, Deng Haotang & Wang Ping.* (2020). K-BERT: Enabling Language Representation with Knowledge Graph. Proceedings of the AAAI Conference on Artificial Intelligence. 34. 2901-2908. 10.1609/aaai.v34i03.5681.

[7] *Loshchilov Ilya & Frank Hutter*. "Decoupled Weight Decay Regularization." International Conference on Learning Representations (2017).

[8] *Mikolov Tomas, Chen Kai, Corrado G.s & Dean Jeffrey*. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. 2013.

[9] *Pan Shimei & Ding Tao*. (2019). Social Media-based User Embedding: A Literature Review.

[10] *Pennington Jeffrey, Socher Richard & Mannin, Christopher*. (2014). Glove: Global Vectors for Word Representation. EMNLP. 14. 1532-1543. 10.3115/v1/D14-1162.

[11] *Popescu Marius-Constantin, Balas Valentina, Perescu-Popescu Liliana & Mastorakis Nikos*. (2009). Multilayer perceptron and neural networks. WSEAS Transactions on Circuits and Systems. 8.

[12] *Yonghui Wu, Schuster Mike, Chen Zhifeng, Le Quoc, Norouz, Mohammad, Macherey Wolfgang, Krikun Maxim, Cao Yuan, Gao Qin, Macherey Klaus, Klingner Jeff, Shah Apurva, Johnson Melvin, Liu Xiaobing, Kaiser Lukasz, Gouws Stephan, Kato Yoshikiyo,*

Kudo Taku, Kazawa Hideto & Dean Jeffrey. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.

[13] *Touvron Hugo, Martin Louis, Stone Kevin, Albert Peter, Almahairi Amjad, Babaei Yasmine, Bashlykov Nikolay, Batra Soumya, Bhargava Prajjwal, Bhosale Shruti, Bikel Dan, Blecher Lukas, Ferrer Cristian, Chen Moya, Cucurull Guillem, Esiobu David, Fernandes Jude, Fu Jeremy, Fu Wenyin & Scialom Thomas*. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models.

[14] *Vaswani, Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan, Kaiser Lukasz & Polosukhin Illia*. (2017). Attention Is All You Need.

[15] *Wnkhade Mayur, Rao Annavarapu & Kulkarni Chaitanya*. (2022). A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review. 55. 1-50. 10.1007/s10462-022-10144-1.

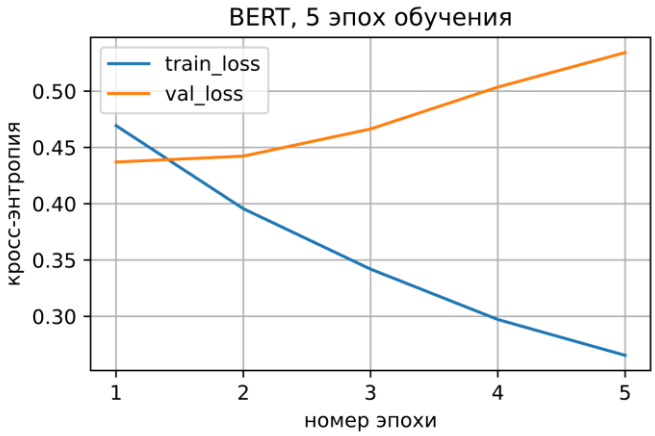
[16] *Yu Shanshan, Jindian Su & Luo Da.* (2019). Improving BERT-Based Text Classification With Auxiliary Sentence and Domain Knowledge. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2953990.

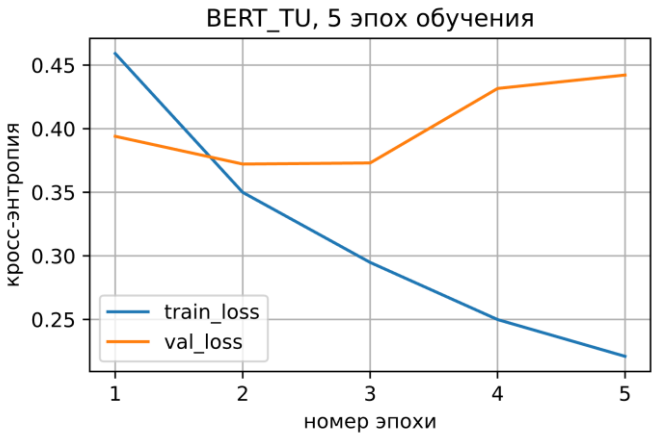
[17] *Zhang Xinyang, Malkov Yu, Florez Omar, Park Serim, McWilliams Brian, Han Jiawei & El-Kishky Ahmed.* (2022). TwHIN-BERT: A Socially-Enriched Pre-trained Language Model for Multilingual Tweet Representations. 10.48550/arXiv.2209.07562.

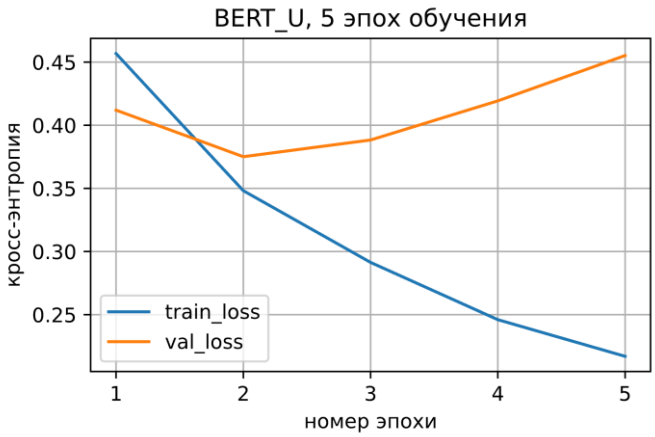
Приложение 1

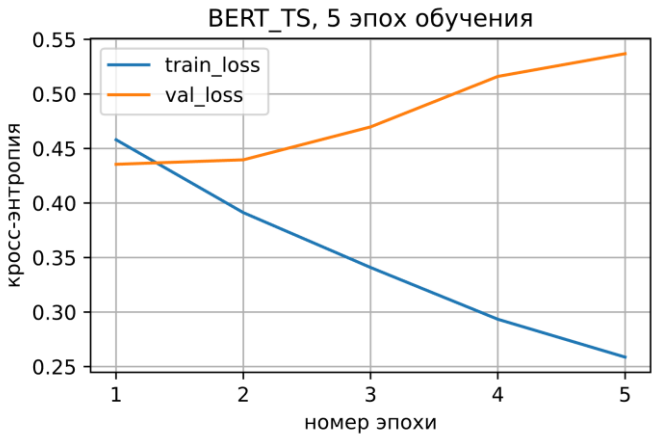
Здесь представлены графики значений функции потерь от номера эпохи для различных моделей во время дообучения. Оранжевая линия (val_loss) отвечает за значение кросс-энтропии на валидационной выборке, синяя (train_loss) – на обучающей выборке.

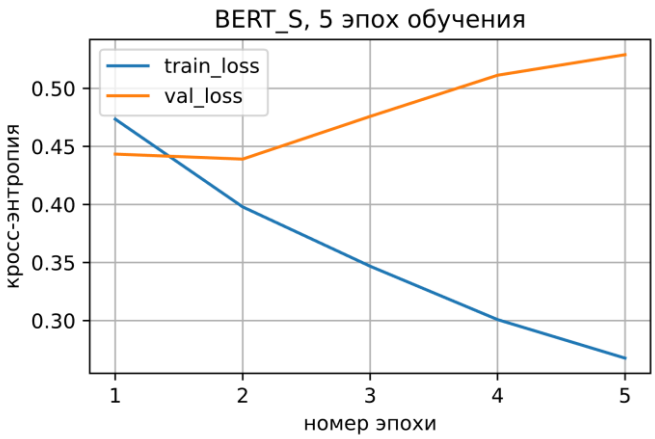
Для моделей из первой и второй групп:



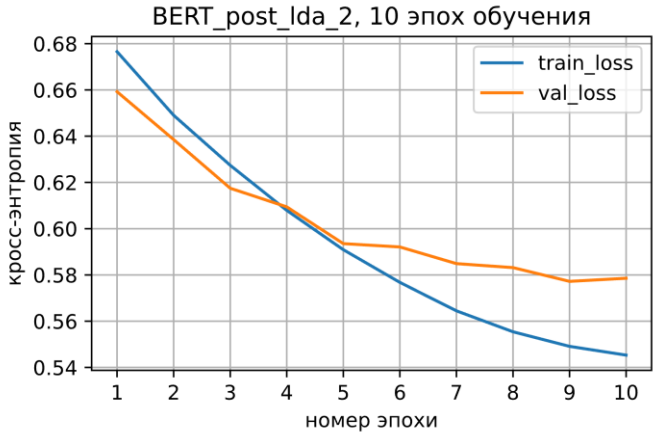


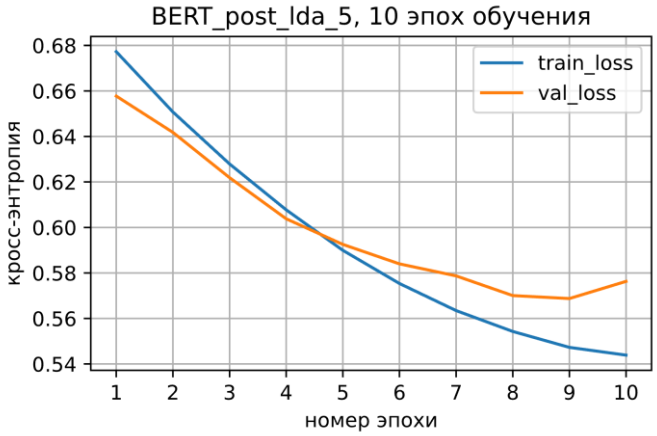


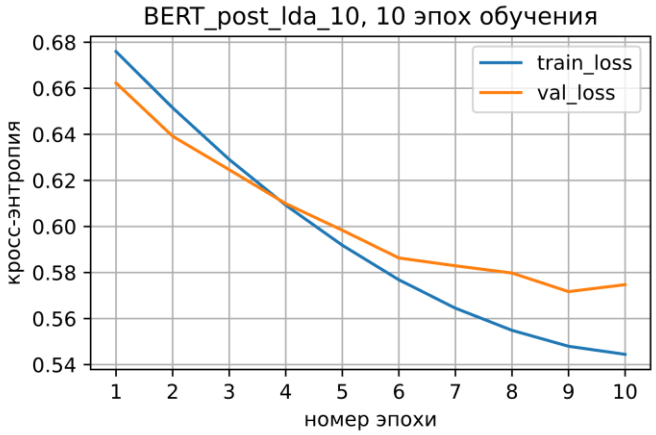


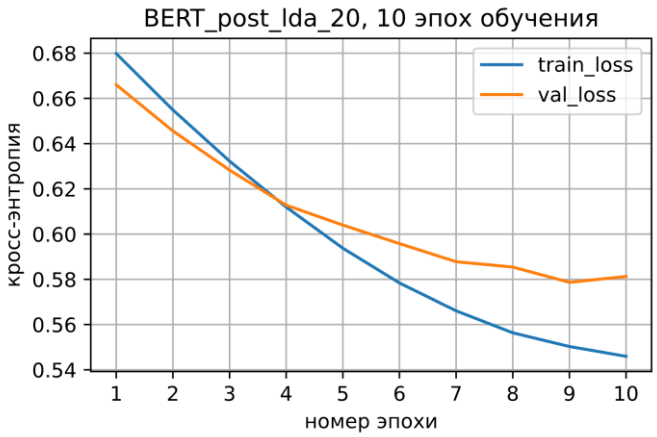


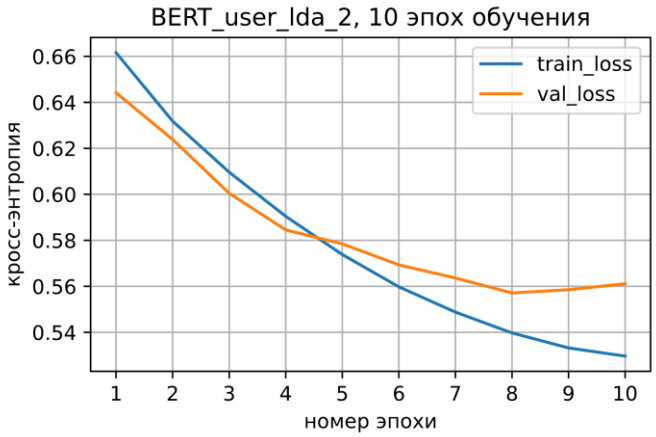
Для моделей из третьей и четвертой групп:

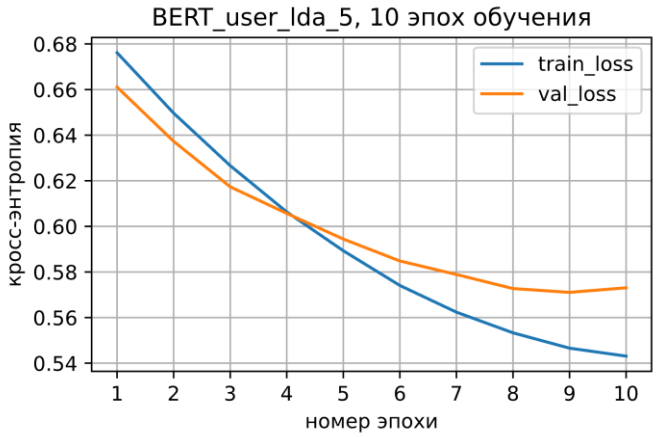


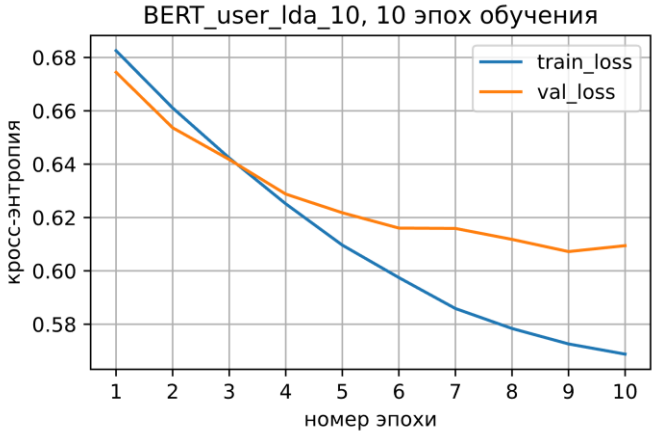


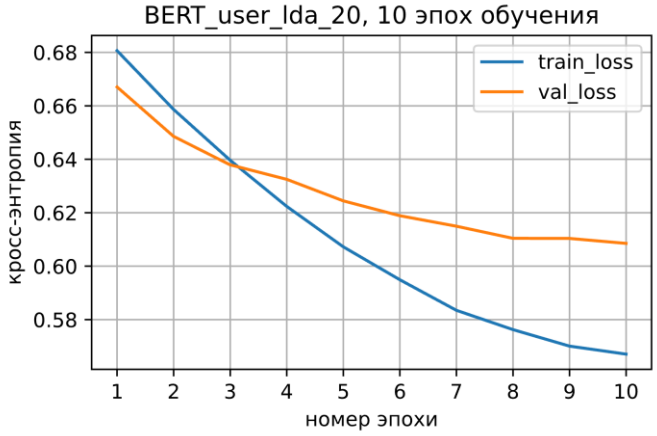


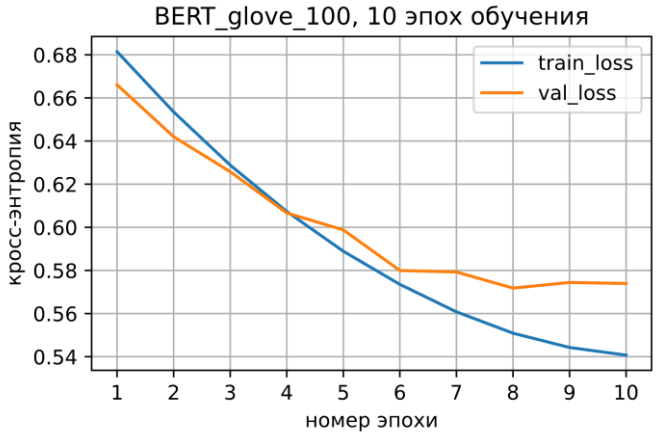


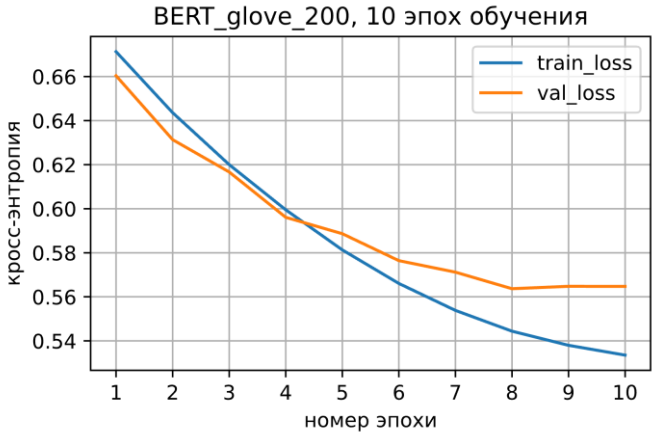


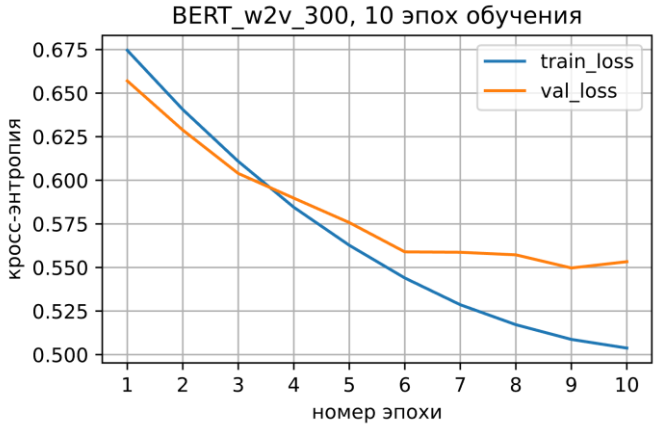


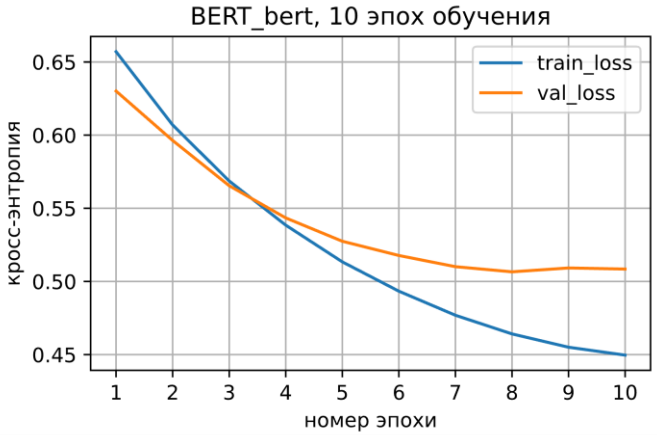


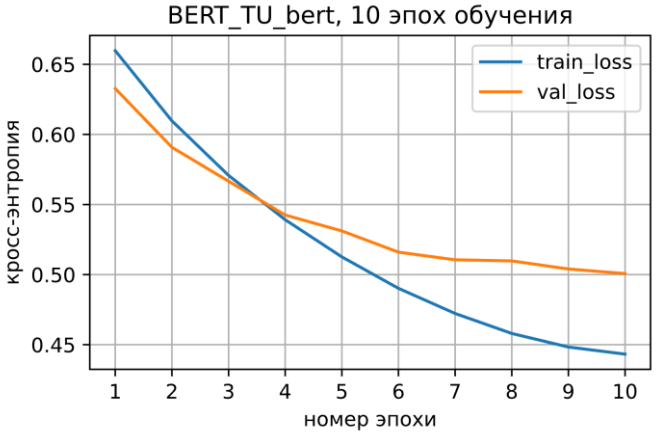


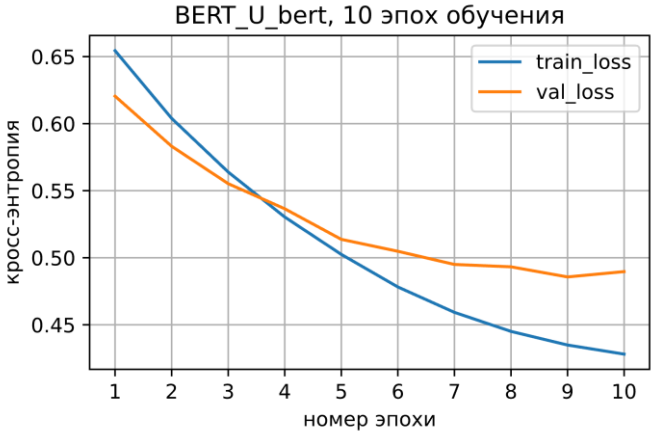

BERT_glove_100, 10 эпох обучения
train_loss
val_loss
кросс-энтропия
номер эпохи


BERT_glove_200, 10 эпох обучения
train_loss
val_loss
кросс-энтропия
номер эпохи


BERT_w2v_300, 10 эпох обучения
train_loss
val_loss
кросс-энтропия
номер эпохи


BERT_bert, 10 эпох обучения
train_loss
val_loss
кросс-энтропия
номер эпохи


BERT_TU_bert, 10 эпох обучения
train_loss
val_loss
кросс-энтропия
номер эпохи


BERT_U_bert, 10 эпох обучения
train_loss
val_loss
кросс-энтропия
номер эпохи