Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
Физтех-школа Прикладной Математики и Информатики
Кафедра интеллектуальных систем

Направление подготовки / специальность: 03.03.01 Прикладные математика и физика
Направленность (профиль) подготовки: Математическая физика, компьютерные технологии и математическое моделирование в экономике

ПОИСК СВЯЗИ ФРАГМЕНТОВ МАНИПУЛЯЦИЙ С ИМЕНОВАННЫМИ СУЩНОСТЯМИ В ТЕКСТАХ

(бакалаврская работа)

Студент:
Жаров Георгий

_____
(подпись студента)

Научный руководитель:
Воронцов Константин Вячеславович,
д-р физ.-мат. наук

_____
(подпись научного руководителя)

Консультант *(при наличии)*:
_____
_____
(подпись консультанта)

Москва 2023

Аннотация

В данной работе рассматривается задача поиска пропаганды. Впервые задача обнаружения пропаганды в англоязычных новостных текстах была поставлена как задача обработки естественного языка в 2019. В такой постановке задача является совокупностью двух различных задач, а именно, выделения фрагмента пропагандистского содержания и его последующая классификация. В данной работе ставится аналогичная задача на русском языке и подробно рассматривается задача классификации фрагментов пропаганды.

Цель данной работы состоит в разработке базовой модели нейронной сети для классификации манипулятивных фрагментов в русскоязычных новостных текстах. Для реализации этого был проведен анализ существующих методов классификации текстов, изучены различные варианты архитектур нейронных сетей, пригодных для решения поставленной задачи.

В результате работы была разработана рабочая модель классификации, основанная на большой предобученной языковой модели типа трансформер. Для обучения и тестирования модели была использована специально подготовленная и размеченная лингвистами выборка текстов, содержащих фрагменты пропаганды. В работе подробно описывается процесс сбора и подготовки данных, процесс обучения модели, а также приводятся подробно результаты всех поставленных вычислительных экспериментов.

Содержание

Глава 1

Введение

Наличие пропаганды в новостных текстах является актуальной и серьезной проблемой в современном мире. Каждый день люди сталкиваются с информацией, которая пытается повлиять на их мнения и взгляды на окружающий мир. Однако, не всегда легко определить, какая информация является объективной, а какая – пропагандой. Данная проблема рождает необходимость создания эффективного инструмента для выявления манипулятивных элементов в новостных текстах. Одним из таких инструментов могут послужить модели нейронных сетей, обученные для выявления пропаганды в текстовых данных. Как следствие этого, задача поиска и классификации пропаганды стала одной из наиболее интересных и актуальных задач обработки естественного языка.

Впервые задача поиска пропаганды в новостных англоязычных текстах была поставлена как задача машинного обучения исследовательской группой под руководством Преслава Накова. Изначально, в 2019 году ими была рассмотрена задача классификации на уровне документов. Для каждого документа, который являлся новостной статьей, было необходимо определить, является ли он пропагандистским или нет. Позже, группой Преслава Накова была поставлена новая задача уже на уровне предложений и слов. В рамках такой постановки решается уже две отдельные задачи: задача выделения пропагандистских фрагментов (либо далее в тексте фрагментов манипуляций) и задача классификации выделенных фрагментов. В данной работе подробно рассматривается именно задача классификации.

До недавнего времени задача в такой постановке не ставилась на русском языке. Как следствие, до этого не существовало готовых размеченных текстов, используя которые можно было бы обучать модели нейронных сетей, что само по себе является отдельной проблемой. Отличительной особенностью рассматриваемой задачи в силу ее специфики является необходимость наличия подготовленной профессиональными лингвистами разметки. Подробнее о процессе подготовки данных, использованных в данной работе будет подробно написано ниже в отдельной главе.

Важным отличием в постановке задачи, рассматриваемой в данной работе, является наличие известной цели, на которую направлено действие пропаганды. Это

отличает рассматриваемую задачу от задачи сформулированной группой Преслава Накова. Таким образом, везде далее в работе под задачей поиска пропаганды подразумевается совокупность трех задач: задачи выделения фрагмента пропаганды, задачи определения связи между фрагментом пропаганды и целью и задачи классификации фрагмента пропаганды.

Как уже было сказано, данная работа в основном посвящена задаче классификации фрагментов манипуляций. Следовательно, основной целью данной работы является создание базовой нейросетевой модели классификации манипулятивных фрагментов. Для достижения цели в работе исследуются различные аспекты обучения модели, позволяющие повысить качество классификации, рассматриваются различные варианты архитектуры нейронной сети и способы обучения в условиях сильного дисбаланса классов.

Глава 2

Постановка задачи

2.1 Особенности постановки

В данной работе ставиться задача классификации фрагментов новостных текстов. Каждый рассматриваемый фрагмент представляет из себя последовательность слов, выделенную из новостного текста и отмеченную лингвистами как пропаганда. Необходимо определить к какому из 18 размеченных классов пропаганды (классов манипуляций) относится рассматриваемый фрагмент. По сути ставится задача Sequence Classification.

Для получения размеченных данных привлекались лингвисты, которые выделяли в новостных текстах фрагменты, по их мнению, являющиеся манипулятивными, а также сопоставляли выделенному фрагменту определенный класс. Важно отметить, что один и тот же новостной текст размечался независимо несколькими асессорами. Так как задача обнаружения пропаганды является субъективной, некоторые фрагменты разными разметчиками могли быть отнесены к различным классам манипуляций. Вследствие этого разметка имеет некоторую несогласованность — в данных могут встречаться совпадающие текстовые фрагменты, имеющие при этом отличные друг от друга метки класса. Такое свойство задачи добавляет в данные дополнительный шум и значительно усложняет поставленную задачу классификации. Про способы учета шума в данных, вызванного несогласованностью разметчиков, при оценивании качества работы модели написано ниже.

Также в силу специфики задачи, классы в полученном в результате разметки датасете сильно несбалансированы, так, например, в обучающей выборке имеется 1376 объектов самого частого класса «Вкрапление депрессивов» и 8 объектов самого редкого класса «Поставка мишени в один ряд с негативно оцениваемым объектом». Такая несбалансированность данных также усложняет процесс обучения модели. Про сами данные и про рассмотренные методы борьбы с дисбалансом классов будет написано в отдельной главе ниже.

2.2 Математическая постановка

Итак, данные, использованные в работе, представляют из себя корпус текста (s, c), где s – это фрагмент текста, c – метка класса. Для того чтобы нейростетевая модель могла работать с текстом, последовательности необходимо предварительно токенизировать. Для этого в данной работе используется BPE-токенизатор предобученной модели-энкодера. Токенизатор сопоставляет входящей текстовой последовательности вектор токенов фиксированной длины, зависящей от выбранной модели-энкодера. Входная последовательность, представленная в виде уже последовательности токенов из предобученного словаря, либо обрезается до нужной длинны, либо, наоборот дополняется специальными токенами, если получившаяся длина меньше необходимой.

Пусть S – пространство текстовых последовательностей s, t – токенизатор, а V – словарь всевозможных токенов предобученной модели. Тогда токенизатор работает следующим образом

$$t : S \to (V)^n,$$

где n – это фиксированная длина входного вектора предобученной модели. В данной работе словарь V и токенизатор t зависят только от предобученной модели и не дообучаются на новые данные.

Пусть теперь $a(w)$ – наша рассматриваемая модель, w – параметры модели. P – пространство векторов из $\mathbb{R}^{18}$, таких что $\forall p \in P : \sum_{i=1}^{18} p_i = 1$. Таким образом модель работает как

$$a : V \to P$$

Пусть теперь $\hat{\text{c}}$ – предсказание модели, оно получается следующим образом

$$\hat{\text{c}} = arg\max_i a((V)^n, w)$$

Рассмотрим кросс-энтропийную функцию потерь $CE(y, p) = -\sum_{i=1}^{18} y_i \log p_i$, где $y_i \in \{0, 1\}$ – метка, которая показывает, является ли классификация верной. Тогда итоговая задача будет иметь следующий вид

$$CE(y, a(t(s), w)) \to \min_w$$

2.3 Относительные метрики качества

Стандартными метриками для оценки качества решения задачи классификации выступают accuracy (точность), precision (точность классификации), recall (полнота классификации), F1-мера. Перечисленные метрики использовались как базовые для оценки работы обучаемой модели. Метрики precision, recall и F1-мера подсчитывались с использованием макроусреднения, чтобы заметить возможное переобучение модели на самые частые классы и достоверно оценивать предсказание модели на редких классах. С такой же мотивацией в качестве целевой метрики была выбрана F1-мера, она в отличие от accuracy более корректно отражает качество классификации на 3 и более класса, а также комбинирует в себе метрики precision и recall.

Несмотря на то, что описанные выше метрики являются универсальными для задачи классификации, есть смысл рассмотреть другие способы оценки качества модели, более подходящие под особенности задачи. Как было написано выше, в силу специфики задачи поиска пропаганды, итоговая разметка получилась несогласованной. Такой шум данных замедляет процесс схождения алгоритма обучения модели и ухудшает ее прогноз. Поэтому предлагается учитывать несогласованность разметчиков при оценке качества модели. Другими словами, можно оценивать не отдельно взятую модель, а сравнивать качество решения задачи классификации пропаганды построенным алгоритмом, с качеством решения той же задачи человеком. Для этого в соответствии с обыкновенной F1-мерой вводится относительная F1-мера ($RF1$), которая формально определяется следующим образом.

Рассмотрим обычную F1-меру, но посчитанную между разметками одного и того же текста двумя лингвистами:

$$RelF1_i = F1(S_{ij}, S_{ik}),$$

где S_{ij}, S_{ik}, $-$ разметки i-ого текста j-м и k-м разметчиком соответственно. Далее посчитаем среднюю меру для разметчиков ($MAF1$), усреднив $RelF1_i$ по всем текстам:

$$MAF1 = \frac{1}{M}\sum_{i=1}^{M} RelF1_i,$$

где M $-$ число текстов в датасете. Далее, F1-мера, посчитанная по предсказанию модели, делится на $MAF1$ и получается относительная метрика $RF1$:

$$RF1 = \frac{F1(c, \hat{c})}{MAF1}$$

Таким образом, относительная метрика $RF1$ позволяет более адекватно оценивать качество решения поставленной задачи, так как позволяет учитывать шум в данных от которого нельзя избавится.

Помимо описанного выше положительного свойства относительной F1-меры важно обратить внимание на еще один момент. Задача поиска и классификации пропаганды на уровне слов ранее не ставилась для русскоязычных текстов, поэтому нет возможности сравнить метрики, полученные в данной работе с метриками из других работ по этой же тематике. Иными словами, тяжело оценивать результат модели по отдельно взятой метрике без какого-либо сравнения. Полученная безразмерная величина $RF1$ позволяет сравнивать модель напрямую с человеком — метрика $RF1$ показывает, как хорошо модель решает задачу относительно того, как ее решают лингвисты. Таким образом, если мы будем отталкиваться от относительной F1-меры, то максимально возможным качеством модели будет $RF1 \approx 1$, такое значение относительной метрики будет говорить, что модель решает задачу на уровне лингвистов, размечавших обучающую выборку.

Исходя из выше изложенного, в данной работе для оценки качества обученной модели будут использоваться F1-мера, относительная F1-мера ($RF1$) и accuracy. Точные значения полученных метрик указаны ниже в разделе, посвященном вычислительному эксперименту.

Глава 3

Обзор литературы

Задача поиска пропаганды впервые была поставлена в 2019 году командой исследователей под руководством Преслава Накова [8]. На первых этапах ставилась задача классификации на уровне документа — необходимо было ответить, является ли данный документ (статья) пропагандистским. После решения такой задачи была поставлена задача поиска пропаганды уже на уровне слов и предложений. Исходная задача разделилась на две побочные: выделение фрагментов пропаганды и классификация фрагментов пропаганды [2], [12].

В дальнейшем, данная задача ставилась как одна из задач в рамках открытого конкурса SemEval [7], [12], [1]. SemEval - это ежегодный конкурс, который проводится в рамках конференции Association for Computational Linguistics (ACL). Он посвящен оценке различных систем и методов обработки естественного языка с помощью задач, которые отражают разнообразные аспекты обработки естественного языка. Конкурс начался в 2007 году и до сих пор является одним из самых популярных и влиятельных мероприятий в области обработки естественного языка. Особенность SemEval заключается в том, что он предлагает разнообразные задачи, связанные с обработкой естественного языка, такие как оценка сентимента, семантический анализ, классификация текстов и многие другие. Каждая задача описывается в виде конкретных данных, которые выдаются участникам для обработки, а затем оцениваются по нескольким метрикам. Мероприятие также предлагает общий набор данных, используемых для тренировки и тестирования систем.

В рамках решения задачи классификации пропаганды рассматривались различные модели и архитектуры нейронных сетей. Среди них различные варианты CNN, RNN и моделей на основе трансформеров [8], [7]. В настоящая время для решения проблемы поиска пропаганды в основном используются большие языковые модели на основе трансформеров, так как они с наилучшим качеством решают данную задачу.

Глава 4

Данные

4.1 Процесс создания разметки и описание данных

В качестве исходных данных использовались русскоязычные статьи из различных новостных ресурсов в интернете. Разметка выгруженных текстов производилась с помощью сервиса "Яндекс.Толока". Особенность подготовки данных, использованных в этой работе, заключается в использовании профессиональных аннотаторов — специалистов в области лингвистики, социологии и политологии. Схема с видами манипулятивных техник, рассматриваемых в данной работе, изображена на рис. 1.

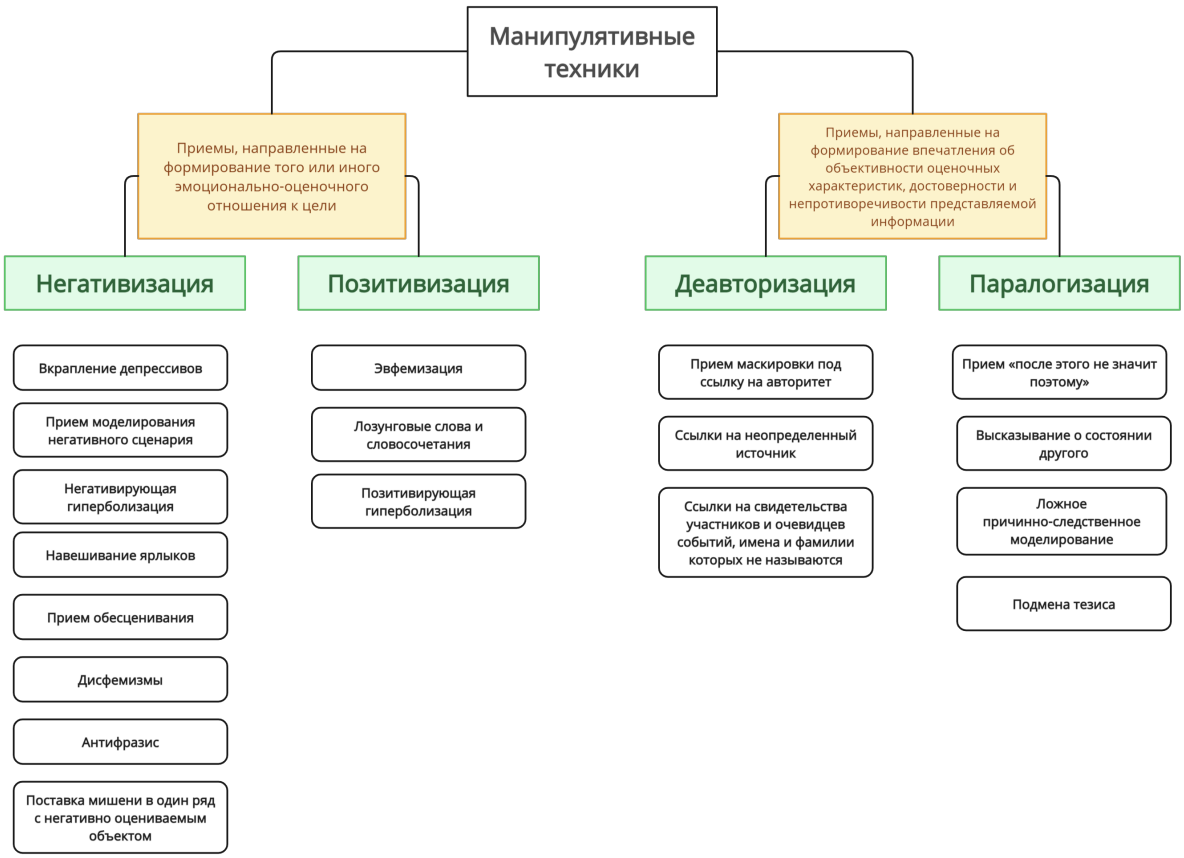


Рис. 1. Виды манипулятивных техник

В соответствии с заранее подготовленной инструкцией аннотаторы выделяли в текстах фрагменты, которые, по их мнению, являются манипулятивными, и присваивали им один из 18 видов манипулятивных техник (классов пропаганды). Представленные 18 классов сгруппированы в 4 больших класса: негативизация, позитивизация, парологизация и деавторизация.

Негативизация и позитивизация представляют собой ценностно противоположные группы приемов. Негативизация основана на внедрении в сознание читателя отрицательного отношения к объекту пропаганды. Позитивизация же направлена на формирование положительного отношения к объекту, вплоть до его возвеличивания, прославления, героизации. Негативизация и позитивизация способствуют изменению эмоционально-оценочной окраски текста. К третьей и четвертой группам относятся манипулятивные приемы, усиливающие оценочное воздействие за счет создания впечатление объективности, достоверности информации и логических выводов. Техники деавторизации помогают скрыть источники приводимой информации и создать впечатление объективности. Приемы паралогизации, в свою очередь, основаны на отступлении от законов формальной логики. Такие приемы направлены на создание необходимых автору текста логических рассуждений и выводов.

Манипуляцию не следует путать с тональностью текста (наличие слов с положительным или отрицательным оттенком, или эмоционально окрашенных слов) по отношению к цели. Разметка манипуляций дополняет оценку тональности текста, но не заменяет ее.

4.2 Распределение классов

Как уже было сказано, каждый текст был размечен несколькими аннотаторами. В среднем, на разметку одного текста у специалиста уходило 5 минут. Всего было размечено 1421 уникальных текстовых документа, в которых было выделено 5443 фрагмента манипуляции.

Распределение по 4 объединенным классам имеет следующий вид:

- Негативизация – 42.00% (2286 из 5443)
- Позитивизация – 36.74% (2000 из 5443)
- Деавторизация – 15.47% (842 из 5443)
- Паралогизация – 5.79% (315 из 5443)

Можно заметить, что уже на уровне крупных объединенных классов присутствует значительный дисбаланс. Ниже в таблице приведео количественное описание для 18 исходных классов.

Класс	**Название класса**	**Количество**
0	Прием «после этого не значит поэтому»	19
1	Вкрапление депрессивов	1376
2	Прием маскировки под ссылку на авторитет	256
3	Прием моделирования негативного сценария	331
4	Негативирующая гиперболизация	282
5	Навешивание ярлыков	444
6	Эвфемизация	259
7	Лозунговые слова и словосочетания	470
8	Ссылки на неопределенный источник	176
9	Ссылки на неназванных участников	114
10	Позитивирующая гиперболизация	194
11	Прием обесценивания	60
12	Дисфемизмы	142
13	Высказывание о состоянии другого	67
14	Ложное причинно-следственное моделирование	31
15	Подмена тезиса	45
16	Антифразис	9
17	Поставка в ряд с негативно оцениваемым объектом	8

Таблица 1. Нумерация и количественный состав классов манипулятивных техник

Для наглядности ниже приведена гистограмма, изображающая распределение основных классов. Номера классов соответствуют номерам в таблице 1.

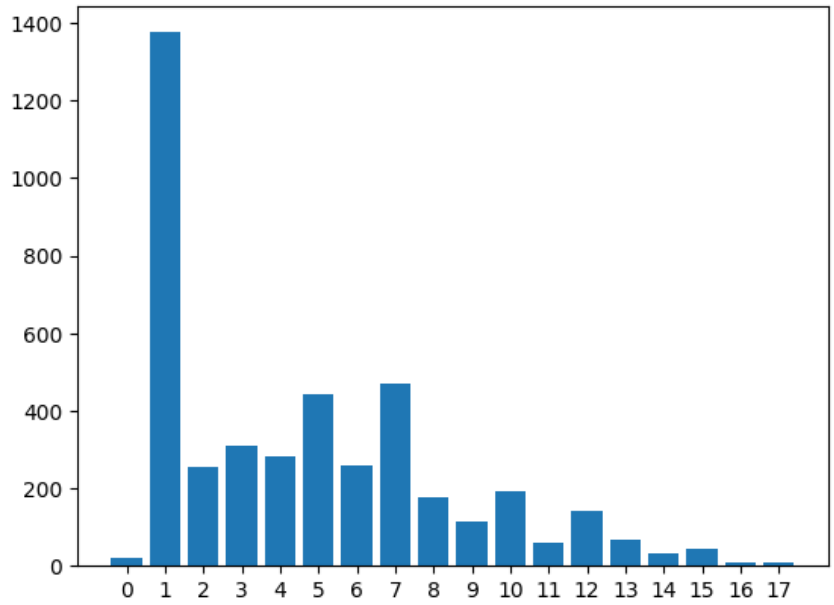


Рис. 2. Виды манипулятивных техник

Из гистограммы выше хорошо видно, насколько существенна несбалансированность классов в рассматриваемой задаче. Такая особенность данных значительно осложняет процесс обучения модели и негативно влияет на качество. О методах, которые использовались для борьбы с данной проблемой написано в следующем разделе.

4.3 Методы борьбы с дисбалансом классов

Говоря о дисбалансе классов в задачах машинного обучения, имеется в виду ситуация, когда один класс (в данном случае, «Вкрапление депрессивов») представлен значительно большим количеством примеров, чем другой класс (в данном случае, классы, представляющие паралогизацию). Это может привести к тому, что модель будет склоняться к предсказанию более частого класса и не сможет достаточно точно предсказывать редкий класс, что существенно скажется на качестве итогового предсказания.

Можно рассмотреть следующие методы борьбы с дисбалансом классов, которые так или иначе были использованы или рассматривались в данной работе:

1. Использование взвешенных функций потерь: при обучении модели можно использовать функции потерь, которые учитывают разницу в количестве примеров для каждого класса. Такие функции дают больший вес ошибкам на редком классе и помогают модели лучше учитывать его. Отдельно можно отметить особые функции потерь, созданные специально для работы с несбалансированными данными. К таким функциям потерь можно отнести так называемую фокальную функцию потерь (Focal Loss [5]).

2. Использование аугментации данных: при подготовке данных для обучения можно создавать новые примеры объектов для редкого класса путем изменения существующих примеров или генерации новых. Такой метод широко используется в задачах компьютерного зрения, например, можно изменять яркость или контрастность изображений, тем самым пополняя выборку новыми объектами. Стоит отметить, что в задачах обработки естественного языка такой способ применяется достаточно редко.

3. Использование методов подвыборки: можно брать в обучающую выборку не все объекты из более частого класса, а случайным образом сэмплировать примеры из него чтобы сбалансировать количество примеров для каждого из классов.

4. Использование методов копирования: можно повторять редкие объекты несколько раз, чтобы уравнять количество примеров в каждом классе.

Обсудим более детально методы, описанные выше, начиная с метода использования взвешенной функции. Как уже было сказано, основной функцией потерь, которая использовалась при обучении модели в данной работе, является многоклассовая кросс-энтропия. В программном пакете PyTorch, с помощью которого ставился вычислительный эксперимент, кросс-энтропия реализована следующим образом:

$$CE(x, y) = (l_1, \ldots, l_N)^T, l_i = -\sum_{c=1}^{C} w_c \log \frac{\exp(x_{i,c})}{\sum\limits_{j=1}^{C} \exp(x_{i,j})} y_{i,c},$$

где x – это входной тензор, y – тензор меток класса, C – количество классов, w_c – вес соответствующего класса, N – число объектов в батче, а $y_{i,c}$ – индикатор, показывающий относится ли i-й объект к c-ому классу. Таким образом, выбирая вектор весов классов w, мы можем регулировать на сколько сильно функция потерь будет штрафовать модель за ошибку на том или ином классе. В данной работе в качестве основных рассматривались веса, рассчитанные по следующим формулам.

$$w_i = 1 - \frac{Count_i}{CountAll},$$

а также

$$w_i = \ln \frac{CountAll}{Count_i},$$

где $CountAll$ это размер всей выборки, а $Count_i$ это число объектов i-го класса. Данные способы подбора весов для классов были выбраны эмпирическим путем. Модели, обучавшиеся с такими весами по результатам проведенных экспериментов, показывали лучшее качество в сравнении с другими моделями.

Также помимо кросс-энтропии были проведены эксперименты с фокальной функцией потерь. Данная функция потерь была предложена для борьбы с дисбалансом классов. Фокальная функция потерь хорошо зарекомендовала себя в задачах компьютерного зрения [5], в частности в задаче семантической сегментации. Ее идея заключается в том, чтобы меньшее внимание уделять тем объектам, к классификации которых модель более уверена. Эта функция потерь задается следующей формулой:

$$L = -(1 - p_i)^\gamma \log p_i,$$

где p_i –это вероятность принадлежности i-му классу, а γ – настраиваемый гиперпараметр. По результатам ряда экспериментов было выяснено, что использование при обучении фокальной функции потерь не дает какого-либо существенного прироста к качеству работы модели, поэтому в качестве основной функции потерь была оставлена кросс-энтропия.

Использование аугментации данных также рассматривалось на ранних этапах работы как еще один потенциальный метод борьбы с дисбалансом классов. В качестве аугментации, предполагалось использовать генеративные текстовые модели или синонимайзеры для расширения обучающей выборки и генерации новых объектов редких классов. Однако такой способ оказался слишком сложным в реализации. Помимо технической сложности исполнения, созданные таким образом данные могли внести дополнительный шум в исходную выборку. Также при добавлении таких синтетических объектов бы вставал вопрос о применимости относительных метрик, предложенных для оценки качества решения данной задачи. В силу изложенных недостатков идея использования описанных аугментаций в итоге была отвергнута.

Еще одним способом борьбы с переобучением модели на самые крупные классы является сглаживание меток (label smoothing [9]). Данный подход заключается в том, что в нем минимизируется кросс-энтропия между вектором таргетов и специальным сглаженным вектором вероятности $\hat{p}(x_i)$, который вычисляется как

$$\hat{p}(x_i) = (1 - \alpha)p(x_i) + \frac{\alpha}{C},$$

где p – это вектор вероятности объекта x_i, C – количество классов, а $\alpha \in (0, 1)$ – параметр. Использование сглаживания меток позволило улучшить сходимость алгоритма обучения модели и уменьшить переобучение.

Последним из методов борьбы с дисбалансом классов, использованных в данной работе, является метод, который представляет из себя комбинацию методов подвыборки и копирования, описанных выше. В данном подходе для обучения модели создавалась новая выборка на основе исходных данных. Процедура создания валидационной выборки происходила следующим образом. Исходная выборка делилась на две выборки, одна из которых являлась валидационной, при этом разделение происходило так, чтобы распределение классов в обоих выборках было одинаковым и совпадало с изначальным распределением всей выборки. После этого валидационная выборка уже не подвергалась никаким преобразованиям, это было сделано для обеспечения объективного тестирования модели. На основе же второй выборки готовилась новая обучающая подвыборка. Она заполнялась по следующему алгоритму: сначала из равномерного распределения сэмплируется номер класса, далее, равновероятно выбирается один объект из всех объектов, принадлежащих этому классу, и добавляется в подвыборку. Данная процедура повторяется пока размер новой выборки не станет равен размеру исходной. Иными словами, циклично осуществляется два выбора объектов с возвращением, сначала из множества классов, потом из множества объектов данного класса.

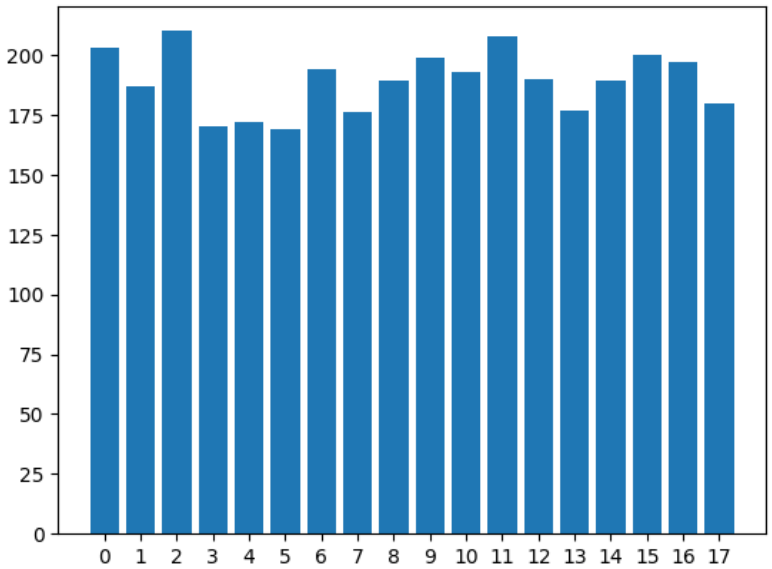


Рис. 3. Новое распределение классов в обучающей выборке

Таким образом, в новой выборке будет присутствовать только некоторое подмножество частых классов, в то время как самые редкие классы будут продублированы несколько раз. В итоге получается искусственно созданное равномерное распределение, пример которого изображен на рисунке 3. Стоит также отметить, что при создании подвыборки можно использовать другое распределение вместо равномерного, например, сделать вероятность выбора класса при сэмплировании обратно пропорциональной доли класса в датасете, тем самым сделав наиболее редкие классы наоборот самыми частыми. Однако, такой вариант не показал какой-либо существенной эффективности во время проведения экспериментов.

В сравнении со всеми перечисленными описанный способ создания новой равномерной обучающей подвыборки позволил больше всего повысить стабильность обучения и улучшить итоговый прогноз. При этом, в при обучении итогового варианта модели использовались также взвешенная функция потерь и сглаживание меток, так как комбинация данных подходов давала наилучший результат.

Глава 5

Архитектура модели

5.1 Общие соображения о выборе модели

Задача поиска и классификации пропаганды является весьма сложной задачей, для решения которой необходим подготовленный специалист в области лингвистики. Помимо этого, данная задача в немалой степени является субъективной, что также добавляет сложности к ее решению. Поэтому, если мы ставим такую задачу как задачу обработки естественного языка и предполагаем решать ее с помощью моделей нейронных сетей, то мы должны требовать от модели высокого уровня понимания естественного языка, умения работать с текстовым контекстом, а также возможности понимать различные смысловые оттенки и эмоциональную окрашенность текста.

Архитектура нейронной сети на основе трансформера [10] является одной из наиболее эффективных для решения задач классификации пропаганды, так как она отвечает всем свойствам, описанным выше. Как уже было сказано в задаче поиска пропаганды необходимо учитывать контекст информации и последовательности слов, так как учет контекстной информации при анализе текста делает возможным обнаружение более тонких форм пропагандистской информации. Такая возможность проще всего достигается, как раз, с помощью архитектуры на основе трансформера. Также важным преимуществом трансформера является его способность моделировать длинные временные ряды (в данном случает текст) и обращать внимание на ключевые элементы последовательности. В такой задаче, где важен порядок слов и связь между ними, данная особенность помогает лучше справляться с обработкой и анализом входного теста.

Еще одним аргументом в пользу построения модели на основе трансформеров является наличие предобученных моделей, которые можно найти в открытом доступе. Большая предобученная модель, такая как BERT [3], изначально обладает хорошим уровнем понимания естественного языка, что позволяет ей с высоким качеством решать большой спектр задач, связанных с обработкой текстовой информации. Помимо этого, с точки зрения сложности реализации и затрат вычислительных ресурсов, дообучение существующей модели под конкретную задачу гораздо выгоднее

чем обучение новой модели на основе трансформеров. В силу огромного количества параметров такой модели и отсутствия достаточного количества данных, обучение с нуля трансформерной модели нейронной сети не представляется возможным.

Подводя итоги сказанного, в данной работе для решения поставленной задачи в качестве модели была выбрана модель, использующая в качестве векторизатора предобученный энкодер и добавляющая к нему ряд дополнительных слоев. Подробнее об энкодере написано в следующем разделе.

5.2 Описание работы энкодера

В данной работе было принято решение, в качестве векторизатора текста использовать предобученный энкодер, основанный на архитектуре трансформера. Первой из рассмотренных в качестве такого энкодера моделей была модель BERT.

BERT (от англ. “Bidirectional Encoder Representations from Transformers”) - это один из примеров больших предобученных энкодеров, работающих на принципе контекстнозависимой обработки текста с использованием механизма самовнимания. Обучение данной модели происходило на большом количестве неаннотированных текстов. Модель BERT обучалась без учителя на двух задачах: задаче предсказания следующего слова (Masked Language Model) и задаче определения того, является ли токен двухэлементным предложением (Next Sentence Prediction).

Как уже было сказано, одна из ключевых особенностей BERT — это контекстнозависимая обработка текста. Если говорить более подробно, такая обработка имеет следующие этапы. В начале исходная текстовая последовательность проходит через токенизатор, обучавшийся совместно с моделью, который имеет свой собственный словарь токенов. На выходе из токенизатора текстовая последовательность уже представлена в виде последовательности токенов из словаря фиксированной длинны. После этого, последовательность токенов проходит слой эмбедингов, после прохождения которого последовательность уже представляется в виде действительного вектора. Также на этом слое кодируется информация о порядке следования токенов в исходной последовательности, что является очень важным этапом для понимания смысла входного текста. Далее, вектор проходит через несколько подряд идущих линейных слоев и слоев самовнимания с несколькими головами (self-attention и multi-head attention). По итогу, на выходе появляется сложное векторное представление исходного текста. Итак, каждый токен на выходе из модели BERT представлен не только с использованием его собственного вектора, но и с использованием вектора, основанного на контексте, в котором он находится. Такое устройство архитектуры позволяет ей выявлять различные аспекты языка и использовать эту информацию для выполнения широкого спектра задач. Все выше сказанное делает BERT (и другие большие предобученные энкодеры) наиболее эффективным инструментом для решения задач обработки естественного языка.

В процессе выполнения работы и проведения вычислительных экспериментов было принято решение использовать несколько другую предобученную модель, а именно модель RoBERTa [6]. RoBERTa (от англ. "A Robustly Optimized BERT Pretraining Approach") является расширением модели BERT и была разработана для более эффективной процедуры предобучения и улучшения результатов на задачах обработки естественного языка.

Модель RoBERTa имеет ряд отличий от модели BERT. В целом RoBERTa использует ту же архитектуру трансформера, что и BERT, однако она имеет ряд значительных изменений в конфигурации, таких как увеличенное количество скрытых слоев и использование дополнительных батч-нормализаций (Batch Normalization). Помимо этого, RoBERTa обучалась на большем объеме текстов чем BERT (порядка 160 Гб текстовых данных). Также важным отличием является то, что RoBERTa обучалась только на задаче предсказания следующего слова (Masked Language Model). Помимо этого, в процессе обучения модели RoBERTa авторами было применено несколько оптимизационных улучшений, таких как детерминированный dropout, которые привели к лучшей производительности модели.

Благодаря внесенным изменениям модель RoBERTa успешно решает большой диапазон задач обработки естественного языка и демонстрирует лучшие результаты, чем оригинальная модель BERT. Важно отметить, что при этом обе модели имеют сопоставимую сложность и размер. Поэтому, в силу лучшей производительности и аналогичной вычислительной сложности, итоговый выбор был сделан в пользу RoBERTa.

Важно заметить, что описанные модели были разработаны в основном для английского языка и не всегда работают хорошо с другими языками, включая русский, что является критичным для рассматриваемой задачи. Для решения подобных проблем существуют адаптированные под русский язык большие языковые модели. Они были дообучены на больших корпусах текстов на русском языке и имеют более высокую точность при работе с этим языком. Примерами таких моделей являются ruBERT и ruRoBERTa, которые по результатам экспериментов показали значительно более хорошие результаты чем оригинальные модели. Тем не менее, нужно отметить, что адаптированные под русский язык модели все еще находятся в стадии развития и совершенствования. Однако, они уже доказали свою эффективность и могут быть полезными инструментами для работы с русским языком в задачах обработки естественного языка.

Последняя тема, которую необходимо осветить в данном разделе это текенизатор, который использовался вместе с предобученной моделью, а именно BPE-токенизатор. BPE (от англ. Byte Pair Encoding [11]) является токенизатором, который широко используется в области обработки естественного языка для сегментации текста на значимые единицы.

BPE работает на основе метода сжатия данных, который заключается в постепен-

ном объединении наиболее часто используемых байтов, блоков символов или слов в одно кодовое слово, называемое BPE-словарем. В результате BPE-словарь содержит наиболее часто встречающиеся комбинации символов, что позволяет более эффективно компактно представлять текстовые данные. После создания BPE-словаря, текст разбивается на токены, где каждый токен представляет собой последовательность символов, присутствующих в BPE-словаре.

BPE токенизатор стал особенно популярен в нейронных сетях для представления текста в виде числовых векторов, который может быть использован в задачах классификации текста, машинного перевода и других задачах обработки естественного языка. Важным моментом является то, что токенизатор и модель обучаются совместно и в дальнейшем должны также совместно применятся в решении задач.

5.3 Процесс выбора итоговой архитектуры модели

Как уже было написано выше, для решения задачи классификации пропаганды была выбрана архитектура, представленная токенизатором, векторизатором, в качестве которого выступает предобученный энкодер, и дополнительных линейных слоев идущих после энкодера. В конечном итоге в качестве энкодера решено было взять предобученную модель ruRoBERTa (точнее, ее версию ruRoBERTa-large), в качестве токенизатора — соответствующий взятой модели предобученный BPE-токенизатор.

В процессе обучения все веса предобученного энкодера кроме весов последнего слоя внимания замораживались. Такое решение было принято по ряду причин. При разморозке всех весов энкодера модель, в силу небольшого объема обучающей выборки, очень быстро переобучалась и показывала плохой результат на валидационной выборке. Помимо проблемы с переобучением, сама процедура обучения большой модели энкодера очень затратная с точки зрения вычислительных ресурсов. При этом последний слой размораживался, чтобы позволить большой языковой модели ruRoBERTa дообучиться на рассматриваемую задачу. Обучение с размороженным верхним слоем энкодера, в независимости от архитектуры модели, показало себя лучше нежели обучение с полностью замороженным векторизатором.

Изначально рассматривались различные варианты архитектуры с несколькими подряд идущими после энкодера линейными слоями. Эксперименты проводились с вариантами от двух до четырех линейных слоев с числом нейронов равным размерности выхода энкодера, либо с большей размерностью. После добавлялся еще один линейный слой с числом нейронов равным числу классов. Помимо выбора размера и числа линейных слоев подбирались различные значения параметра dropout и различные нормализации. Однако модели с такими комбинациями слоев и параметров показывали низкое качество итоговой классификации и существенно не отличались друг от друга значением метрик.

Далее был рассмотрен вариант с четырьмя линейными слоями одинаковой размерности и использованием skip connection. В линейную архитектуру была добавлена дополнительная связь между выходным слоем энкодера и последним линейным слоем. Схема такой модели приведена на рисунке ниже.

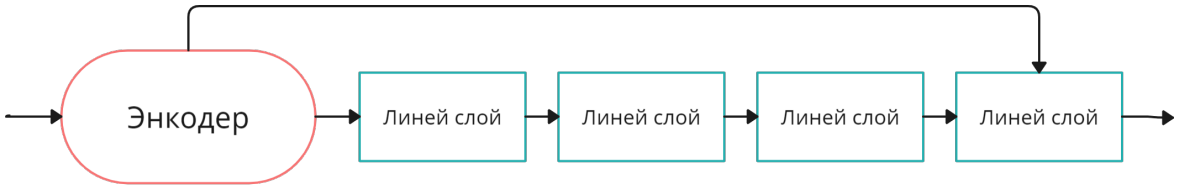


Рис. 4. Схема модели с использованием skip connection

Данный метод был использован с предположением, что новая связь поможет передавать на последний слой больше информации о последовательности, а также добавит дополнительную нелинейность в архитектуру. На эксперименте выяснилось, что использование такой архитектуры позволило значительно повысить значение целевых метрик. Для поиска других вариантов архитектуры были проанализированы англоязычные статьи по тематике поиска пропаганды. Были рассмотрены статьи с отчетами участников открытого конкурса по поиску и классификации пропаганды в англоязычных новостных текстах SemEval ([8], [7], [1], [12]), созданный исследовательской командой под руководством Преслава Накова. Наилучшие результаты классификации на англоязычных данных достигались моделями с одним линейным слоем после энкодера.

По результатам экспериментов выяснилось, что такая модель действительно не уступает по качеству модели, использующей skip connection, или даже превосходит ее. При этом такая модель имеет меньшее число параметров и быстрее обучается. Поэтому в дальнейшем в качестве основной модели выбрана модель с одним линейным слоем после энкодера. Все последующие эксперименты, в том числе подбор гиперпараметров, а также подсчет итоговых метрик, речь о которых пойдет в следующей главе, проводились с данной моделью. Схема модели приводится на рисунке ниже.

Рис. 5. Схема итоговой архитектуры

На рисунке выше d – это число классов манипулятивных техник, в экспериментах оно ранялось 18 либо 4.

5.4 Учет контекста в модели

Еще одним аспектом построения архитектуры является принятие решения о том, как будет учитываться контекст в модели. То есть будет ли помимо самой текстовой последовательности, являющейся фрагментом манипуляции, учитываться новостная статья, в которой этот фрагмент находится. Если обратиться к англоязычным статьям [4], [7], то можно увидеть, что самые лучшие результаты в задаче поиска пропаганды достигались моделями, учитывающими окружающий манипулятивные фрагменты контекст.

Учет контекста происходил следующим образом, на вход модели подавался текст, состоящий из манипулятивного фрагмента. Входной текст выглядит следующим образом: $<$*фрагмент*#*контекст*$>$. Однако, вопреки результатам для англоязычных данных, описанных в статьях по поиску пропаганды, на эксперименте модели, использовавшие при обучении контекст, отличались худшим качеством и меньшей стабильностью при обучении. Подробнее о метриках, посчитанных для различных вариантов модели, написано в следующей главе.

Глава 6

Вычислительный эксперимент

6.1 Общая информация об эксперименте

В данной работе было поставлено большое количество вычислительных экспериментов, под вычислительным экспериментом в основном понимается процесс обучения модели и последующее измерение целевых метрик. В процессе работы были произведены различные эксперименты по подбору архитектуры, подбору функции потерь и весов для нее, исследованию способов борьбы с дисбалансом классов. Для адекватного тестирования моделей и подсчёта метрик в начале была создана валидационная выборка, распределение классов в которой совпадает с распределением в исходной выборке. Валидационная выборка фиксировалась и была одинаковой во всех экспериментах. Реализации моделей нейронных сетей выполнялась с использованием программного пакета PyTorch.

В первую очередь проводились эксперименты по подбору архитектуры. Подробно о рассмотренных вариантах архитектур уже было написано в предыдущей главе. Процесс отбора моделей проходил следующим образом, каждая модель обучалась несколько раз (число эпох было зафиксировано для всех моделей) с различными значениями ядра генератора псевдослучайных чисел. После каждого обучения снимались метрики, которые далее усреднялись по всем обучениям. По этим усредненным метрикам уже выбиралась лучшая модель, в качестве целевой метрики качества, как уже было сказано, рассматривались F1-мера и соответствующая ей относительная F1-мера. В итоге, по качеству всех превзошла модель, состоящая из предобученного энкодера и одного линейного слоя размерности равной числу классов.

После выбора архитектуры модели производились эксперименты с различными предобученными моделями, которые использовались в качестве энкодера. Экспериментами проводились с четырьмя вариантами больших языковых моделей, эти модели: BERT-base, ruBERT-tiny, ruBERT-large, ruRoBERTa-large. Эксперименты проходили так же, как и в случае выбора архитектуры. В качестве итогового варианта была выбрана модель ruRoBERTa-large.

С уже выбранным энкодером далее был поставлен ряд экспериментов, в которых

при обучении размораживалось разное количество слоев энкодера (по умолчанию заморожены были все веса). В итоге, оптимальным с точки зрения качества и вычислительной сложности стал вариант с размороженным одним финальным слоем предобученного энкодера.

После того как итоговый вариант модели был зафиксирован, был проведен ряд экспериментов для выбора оптимальной функции потерь, а также для исследования различных методов противодействия дисбалансу классов. Были рассмотрены варианты с фокальной и кросс-энтропийной функциями потерь. Из них в качестве основной была выбрана взвешенная кросс-энтропия, так как на практике она оказалась эффективнее фокальной функции потерь — подобранные веса лучше справлялись с проблемой дисбаланса классов. Веса для кросс-энтропии также подбирались экспериментально, формулы подсчета весов подробно обсуждались в главе про данные. Также для преодоления проблемы дисбаланса классов использовались различные методы сэмплирования обучающей выборки. В качестве основного был выбран способ создания искусственной равномерной обучающей выборки. Этот способ очень хорошо показал себя в рамках поставленных экспериментов и был выбран как основной.

Единственным аспектом, рассмотренным в главе про архитектуру, но не описанным выше, остается вопрос учета контекста новостной статьи. В силу того, что учет контекста в модели является принципиальным, было принято решение рассмотреть несколько итоговых моделей, учитывающих и не учитывающих контекст соответственно. Также среди итоговых моделей есть модели для классификации на 18 исходных классов и для классификации на 4 обобщенных класса.

По завершении выбора всех основных характеристик модели и элементов архитектуры, был проведен ряд финальных экспериментов для подбора различных гиперпараметров модели.

6.2 Сравнение метрик и анализ результатов

Несколько моделей, а именно 4, были вынесены как итоговые. В данном разделе будут приведены итоговые метрики для этих моделей, графики с динамикой изменения метрик в процессе обучения моделей, а также итоговые подобранные гиперпараметры.

В качестве итоговых выступают модели со следующими названиями: cls18, cls18-context, cls4, cls4-context. Здесь cls18 — это модель, которая обучалась на данных, состоящих только из фрагментов пропаганды, cls18-context - аналогичная модель обучалась на фрагментах и их контексте, cls4-context и cls4 модели, классифицирующие фрагменты на 4 сгруппированных класса, обученные соответственно с контекстом и без. Поведение основных метрик в процессе обучения моделей изображены на рисунке ниже.

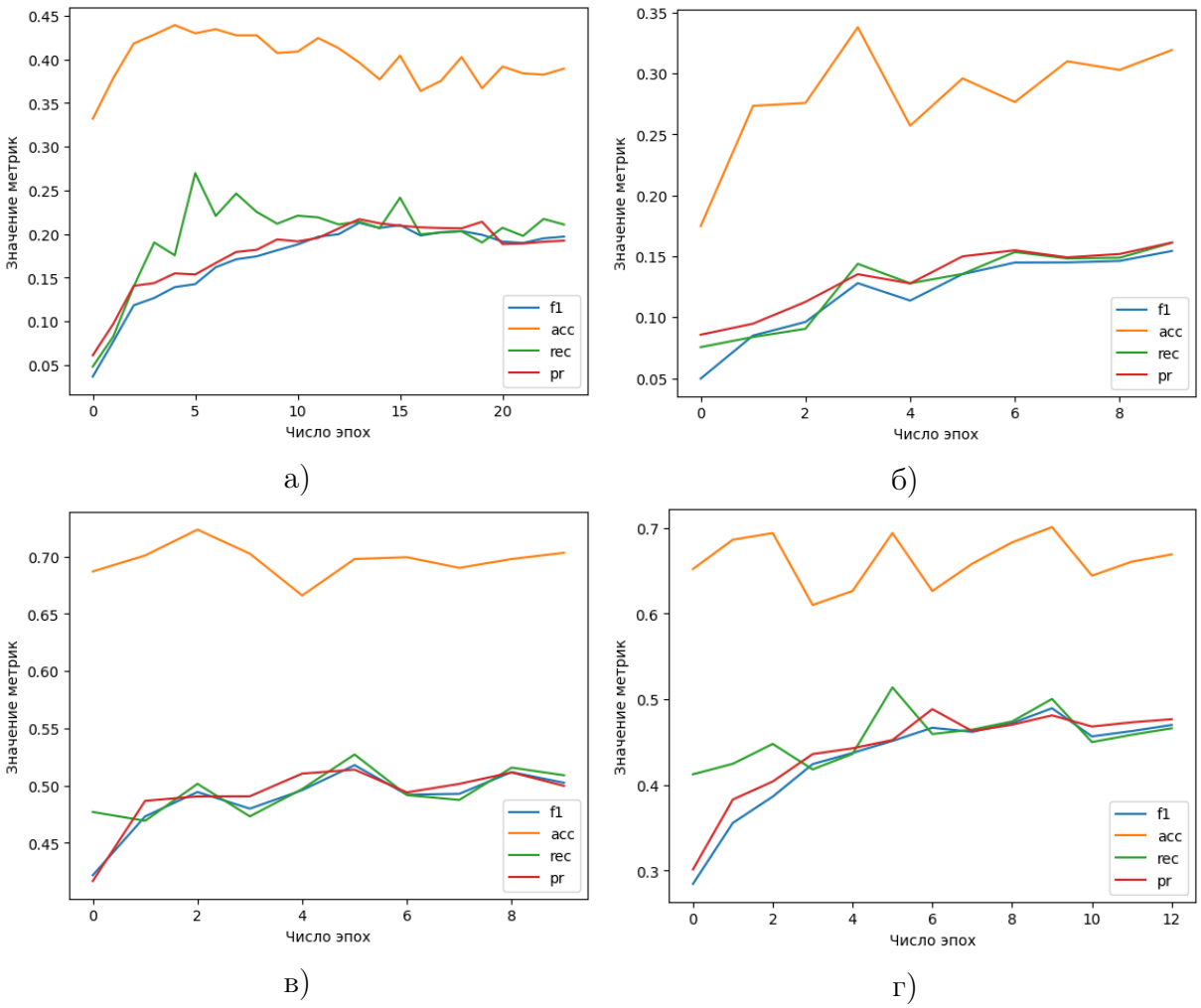


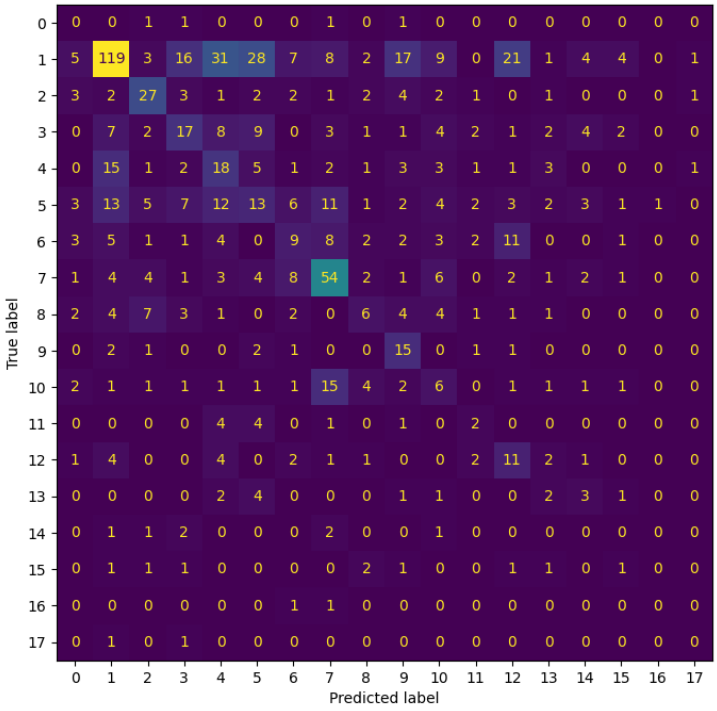
Рис. 6. а) cls18 б) cls18-context в) cls4 г) cls4-context

На графиках выше можно увидеть некоторое общее поведение — метрики растут на протяжении нескольких эпох, далее они выходят на плато и колеблются в пределах одного значения. Так же по графикам видно, что модели, использующие контекст, дольше обучаются и достигают меньшего качества. Такой результат противоречит статьям, посвященным поиску пропаганды. Возможно такое поведение связано с большим шумом в данных, вызванным низким качеством разметки. Также можно увидеть, что качество классификации на 4 класса существенно выше, что является достаточно ожидаемым результатом. Точные значения для метрик четырех рассматриваемых моделей можно увидеть в таблице ниже.

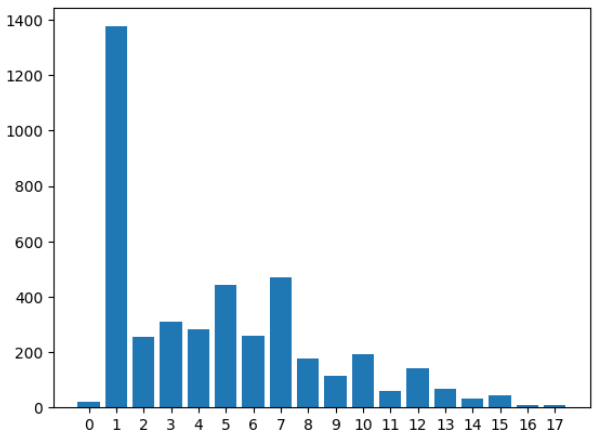
model	ACC	F1	RF1
cls18	0.353	0.249	0.508
cls18-context	0.357	0.166	0.339
cls4-context	0.620	0.443	-
cls4	0.676	0.509	-

Таблица 2. Метрики моделей

Так как изначально ставилась задача классификации на 18 классов, значение метрик именно на этой задаче является более приоритетным. Видно, что F1-мера у модели, обучавшейся с контекстом, значительно хуже меры модели, использовавшей при обучении только сам манипулятивный фрагмент. Такое поведение моделей кажется достаточно нелогичным и требует дальнейшего исследования. Таким образом, основной моделью, представляющей наибольший интерес, является модель cls18, которая достигла максимального значения F1-меры равного 0.249 и значения относительной метрики RF1= 0.508. Иными словами, модель достигает примерно половины качества решения задачи классификации пропаганды человеком. Ниже приведена матрица ошибок этой модели, по которой можно определить, на каких классах чаще всего ошибается модель. Для наглядности рядом также повторно изображено исходное распределение классов.



а)



б)

Рис. 7. а) Матрица ошибок моделиб) Распределение классов

Из матрицы ошибок можно увидеть, что модель хорошо справляется с определением самых частых классов, и почти не классифицирует редкие классы, несмотря на все использованные методы борьбы с дисбалансом классов.

Глава 7

Заключение

В ходе выполнения настоящей работы была проведен анализ, существующий методов классификации фрагментов пропаганды в англоязычных новостных текстах, а также разработана базовая модель для классификации манипулятивных фрагментов на русском языке. В работе были подробно описаны методология сбора и разметки данных, процесс выбора архитектуры модели нейронной сети и последующее ее обучение. Помимо этого, отдельно было изучено большое количество методов борьбы с дисбалансом классов — характерной проблемой для задачи поиска пропаганды. Множество из этих способов в последствии были применены в процессе обучения модели.

В рамках работы также было поставлено большое количество вычислительных экспериментов, направленных на определение наиболее подходящих архитектуры нейронной сети, методов обучения и различных гиперпараметров модели. Результаты экспериментов были зафиксированы, подробно они изложены в разделе, посвященном вычислительному эксперименту. Помимо этого, в процессе выполнения был написан и опубликован код для воспроизведения проведенных вычислительных экспериментов.

Модель, осуществляющая классификацию на 18 классов, которая была выбрана в качественно основной, по результатам эксперимента достигла значения метрики F1= 0.249, или значения относительной метрики RF1= 0.508. Такое значение относительной меры говорит о том, что модель достигает 50% качества решения задачи лингвистом. Такой результат можно считать успешным для первичной базовой модели. Подводя итоги, можно сказать, что цели, поставленные в работе, были выполнены в полной мере.

Список литературы

[1] Firoj Alam и др. “Overview of the WANLP 2022 Shared Task on Propaganda Detection in Arabic”. *arXiv preprint arXiv:2211.10057* (2022).

[2] Ola Altiti, Malak Abdullah, Rasha Obiedat. “Just at semeval-2020 task 11: Detecting propaganda techniques using bert pre-trained model”. *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. 2020, с. 1749—1755.

[3] Jacob Devlin и др. “Bert: Pre-training of deep bidirectional transformers for language understanding”. *arXiv preprint arXiv:1810.04805* (2018).

[4] Zhida Feng и др. “Alpha at SemEval-2021 task 6: Transformer based propaganda classification”. *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*. 2021, с. 99—104.

[5] Tsung-Yi Lin и др. “Focal loss for dense object detection”. *Proceedings of the IEEE international conference on computer vision*. 2017, с. 2980—2988.

[6] Yinhan Liu и др. “Roberta: A robustly optimized bert pretraining approach”. *arXiv preprint arXiv:1907.11692* (2019).

[7] G Martino и др. “SemEval-2020 task 11: Detection of propaganda techniques in news articles”. *arXiv preprint arXiv:2009.02696* (2020).

[8] Giovanni Da San Martino и др. “Fine-grained analysis of propaganda in news articles”. *arXiv preprint arXiv:1910.02517* (2019).

[9] Rafael Müller, Simon Kornblith, Geoffrey E Hinton. “When does label smoothing help?” *Advances in neural information processing systems* **32** (2019).

[10] Ashish Vaswani и др. “Attention is all you need”. *Advances in neural information processing systems* **30** (2017).

[11] Changhan Wang, Kyunghyun Cho, Jiatao Gu. “Neural machine translation with byte-level subwords”. *Proceedings of the AAAI conference on artificial intelligence*. T. 34. 05. 2020, с. 9154—9160.

[12] Seunghak Yu и др. “Interpretable propaganda detection in news articles”. *arXiv preprint arXiv:2108.12802* (2021).