Санкт–Петербургский государственный университет

Оглоблина Алена Дмитриевна

Выпускная квалификационная работа

Система рекомендации тегов для публикации постов на портале Пикабу

Направление: 02.04.02 «Фундаментальная информатика и информационные технологии
ООП ВМ.5503: Технологии баз данных

Руководитель научно-исследовательской работы:
Заведующий кафедрой компьютерного моделирования и многопроцессорных систем,
доктор физ.-мат. наук, профессор
Андрианов Сергей Николаевич

Рецензент:
Лидер команды машинного обучения,
кандидат тех. наук
Кузнецов Андрей Сергеевич

Санкт-Петербург
2022 г.

Содержание

Введение

При публикации контента в интернете пользователь имеет возможность указать набор тегов - релевантных ключевых слов для аннотирования ресурсов [1]. Как правило, автору приходится вручную искать подходящие по смыслу слова. Ускорить процесс публикации возможно с помощью рекомендательной системы тегов, которая, проанализировав введенную информацию и публикации на портале, предложит пользователю список тегов.

Построение рекомендательных систем является важной темой исследования в сферах поиска информации, розничной торговле, электронной коммерции и др. Задача заключается в предсказании объектов, которые вызывают наибольший интерес у пользователя [2].

Один из способов создания рекомендательной системы - классификация пользовательских текстов. Модель классификации пользовательских текстовых документов может использоваться для распределения данных по категориям, которые позволяют строить рекомендательные системы. Крупные сайты также используют результаты классификации подбирая персональную рекламу для каждого пользователя, основываясь на его группе интересов.

Разработки в данной области позволяют обрабатывать и систематизировать большие объемы текстовых данных. Несмотря на то, что существует множество алгоритмов классификации, таких как “Метод опорных векторов”, “Метод k-ближайших соседей”и “Байесовский подход”[3], вопросы о том, как повысить точность классификации и подобрать алгоритм для конкретной задачи, являются актуальными.

В настоящее время пользовательские сообщества и форумы, такие как Reddit [1], Хабр[2] и Пикабу[3], ежедневно публикуют миллионы пользовательских текстов. С постоянным развитием информационных технологий количество таких данных увеличивается [3], поэтому появляется возможность производить анализ данных, классификацию, кластеризацию и дру-

[1] https://www.reddit.com/

[2] https://habr.com/ru/all/

[3] https://pikabu.ru/

гие методы для поиска схожих объектов в выборке [4].

Текстовые документы описанных ресурсов возможно использовать для обучения модели классификации пользовательских текстов, для построения рекомендательной системы.

Постановка задачи

Цель работы - разработать систему рекомендации тегов для текстовых данных. Для этого необходимо:

1. Собрать датасет из постов с сайта Пикабу, которые должны включать в себя тексты и наборы тегов.

2. Определить набор тегов, который будет использоваться для обучения.

3. Преобразовать тексты в числовые векторы.

4. Используя несколько подходов, классифицировать тексты используя их теги как метки классификатора, проэксперементировать с параметрами, сравнить результаты.

5. Оценить качество предсказаний полученной модели.

Глава 1. Обзор литературы

1.1 Рекомендательная система

В последнее время существует большое количество приложений с персонализированными подборками контента. Из-за объемов информации в интернете, у пользователя есть потребность в автоматическом извлечении некоторых элементов, представляющих интерес [5].

Авторы публикации [6] сформировали профиль интересов пользователей по данным из социальной сети twitter. Разработанная система делает вывод, к какому жанру относится опубликованный текст. Далее подбирается список публикаций на портале Reddit, которые также относятся к этим жанрам и пропорциональны интересам пользователя. Таким образом разработанное решение рекомендует слабо связанные темы Reddit, которые вероятнее всего заинтересуют читателя.

В сфере электронной коммерции часто используется подход контентной фильтрации для построения рекомендательной системы. Таким образом элементы для рекомендации выбираются по сходству между содержимым контента и предпочтениями пользователя. В статье [7] рассмотрен способ рекомендации продуктов, при котором подбираются товары, приобретенные пользователями со схожим покупательским интересом.

Также, в работе [8] представлена система, предлагающая теги при публикации изображений. В рамках исследования, использовали готовую модель для сегментирования фотографии, однако выбранные на картинке объекты слабо напоминали пользовательские теги. Поэтому авторы использовали классификацию по уже опубликованным изображениям с тегами, что повлияло на точность рекомендации.

Рассмотренные статьи успешно используют результаты классификации для построения рекомендательной системы. Однако, рассмотренные источники используют разные подходы к классификации данных: рекурентные нейронные сети, наивный байесовский подход, бинарную классификацию, и т. д. Необходимо рассмотреть некоторые методы классификации, чтобы достичь высокой точности предсказания полученных данных.

1.2 Классификация

Существует несколько видов классификации:

1. Бинарная классификация. Такой способ классификации разделяет набор данных на два класса.

2. Мультиклассовая классификация. Заключается в присвоении текста одному из набора классов, количество которых больше двух [9].

3. Мультилейбл классификация. Каждому тексту соответствует набор классов (меток) [9].

Изучив публикации портала Пикабу, стало ясно, что пользовательские тексты публикуются с одним или несколькими ключевыми словами. В таком случае для решения задачи выбрана мультилейбл классификация. Для обучения будет использоваться текст пользовательского поста, а также теги, которые поставил автор при публикации.

Глава 2. Обзор решений

В качестве решений для проведения экспериментов были выбраны следующие подходы и модели классификации: One-Vs-Rest, BinaryRelevance, label-powerset, ClassifierChains, BiGRU, DistilBERT.

2.1 OneVsRestClassifier

Подход One-Vs-Rest заключается в разбиении задачи на непересекающиеся двоичные классификаторы. Данный подход подразумевает, что метки классов взаимоисключающие [10].

2.2 BinaryRelevance

В классификации с помощью BinaryRelevance каждую метку в задаче мультилейбл классификации рассматривают как отдельную задачу классификаций одного класса.

Глобальное обучение с несколькими метками разбивается на набор отдельных задач двоичной классификации с одной меткой. Данный подход можно считать модификацией подхода One-Vs-Rest [11].

2.3 Label Powerset

Классификация с помощью label-powerset, представляет комбинацию всех возможных классов как отдельные наборы меток. Стоит отметить, что при увеличении количества классов, количество созданных меток как комбинаций этих классов, имеет экспоненциальный рост.

Для выполнения классификации с таким походом к формированию классов можно использовать алгоритмы для мультиклассовой классификации, например: Gaussian Naive Bayes, Random Forest Classifier и другие. В отличие от One-Vs-Rest подхода, label-powerset учитывает корреляцию между классами [11].

2.4 ClassifierChains

Данный подход также сводится к задаче бинарной классификации. Алгоритм определяет класс для каждой метки в задаче мультилейбл классификации. Его особенность заключается в учитывании результатов классификации уже обученных данных, таким образом используется корреляция меток, как например и в выше описанном методе [10].

2.5 Нейронные сети

Авторы публикации [12] сравнивают качество классификатора пользовательских сообщений, использующего рекуррентные нейронные сети и предобученную сеть BERT. Среди первой категории сетей, самую высокую точность предсказания удалось достичь используя двунаправленное обучение в рекуррентных сетях.

Эксперименты с предобученными сетями BERT показали, что классификатор с DistilBERT, при которой компактная модель обучается воспроизводить поведение более крупной модели [12], по показателю AUC-ROC, превосходит все остальные модели. Отметим, что использование такого подхода требует больше времени для построения классификатора, чем рекуррентные нейронные сети и их модификации.

Для построения классификаторов на основе нейронных сетей в рамках текущей работы использована модель с двунаправленным обучением рекуррентных нейронных сетей - BiGRU, а также предобученная модель DistilBERT.

BiGRU представляет собой модификацию управляемых рекуррентных нейронных сетей. Это двунаправленный блок, позволяющий сохранять значения из будущих состояний и прошлых состояний контекстных функций, улучшенная сеть долгой краткосрочной памяти [13], [14]. Сеть позволяет уменьшить параметры обучения и повысить эффективность.

Предобученная сеть DistilBERT, на основе BERT, имеет в два раза меньше слоев для обучения. Такая модель подходит для неспецифичных текстовых данных, сокращая количество операций и время их выполнения [15]. При этом, для классификации специфичных данных, например, юри-

дических документов или медицинских заключений, возможно, используя тематический корпус слов, переобучить модель [16].

Глава 3. Сбор данных

Для создания модели классификатора необходимы наборы текстов и, соответствующие каждому из них, метки классов. В рамках данной работы наборами текстов будем считать пользовательские посты, а теги этих постов будем использовать как метки кассов.

3.1 Сбор статей

Портал Пикабу не предоставляет средств для свободного автоматизированного предоставления данных, поэтому датасет собирался с помощью последовательного скачивания всех постов.

Ввод [72]:
```
df.head()
```

Out[72]:

	id	rating	meta-rating	data	author_id	comments	saves	author_name	title	tags	text
0	6443848	-93.0	-9634407868:-9634411402	2019-01-21T15:25:52+03:00	2589536	3	4	['сухоеморе']	['Родители']	['[моё]', 'Родители', 'Дети', 'Родители и дети...	как остро сейчас встаёт вопрос воспитания дете...
1	6443839	2019.0	-7352635456:-7352576905	2019-01-21T15:19:16+03:00	2476185	301	90	['Samco']	['NVidia P106 или как китайцы хотят сплавить н...	['Текст', 'Видеокарта', 'Geforce GTX 1060', 'О...	в интернете в последнее время начинается хайп ...
2	6443837	16.0	-6845500346:-6845499914	2019-01-21T15:17:40+03:00	2091005	6	1	['sendvi4']	['Теневой бизнес арни)']	['Текст', 'Пиво', 'Арнольд Шварценеггер']	NaN
3	6443835	1797.0	-6338473393:-6338428468	2019-01-21T15:15:35+03:00	2614756	326	48	['Anna.vanna04']	['О желторотых бизнесменах']	['[моё]', 'Работа', 'Директор', 'Офисные истор...	однажды мне довелось поработать на одного тако...
4	6443834	NaN	-6084888708:-6084888852	2019-01-21T15:15:26+03:00	2494038	18	1	['AlexUralec']	['Как такое может быть?']	['[моё]', 'Телефон', 'Оператор', 'Теле2', 'Тек...	ищу работу. просматриваю сайты. на авито нашел...

Рис. 1: Собранные данные.

Проанализировав раздел поиска на портале Пикабу, было замечено, что за несколько дней публикуются порядка десяти тысяч постов с тегом "Текст". Поэтому сбор поисковых страниц осуществлялся с указанием параметров: тега "Текст"и периодом времени, равному одному дню.

На странице поиска самый ранний период времени - 1 августа 2010 года, в числовом эквиваленте данный параметр указывается 943. Поэтому каждый шаг алгоритма будет увеличивать это число на единицу. Ссылка, по которой программа начала загружать страницы выглядит следующим образом: https://pikabu.ru/tag/Текст?d=943.

На странице результата поиска выводится 10 статей, остальные статьи необходимо получить используя пагинацию. Блок пагинации на странице представляет собой несколько ссылок вида
https://pikabu.ru/tag/Текст?d=5235page=5. Таким образом можно обходить и загружать страницы изменяя атрибут page в запросе от единицы до ста.

С помощью многопоточной реализации, описанного алгоритма на языке Python, удалось собрать порядка 700 тысяч постов за два часа. Полученные данные представлены на Рис. 1.

Глава 4. Выбор меток для классификации

Отметим, что в используемых данных содержатся порядка 125 тысяч уникальных тегов. Проводить классификацию используя все теги задача с высокими требованиями к вычислительным ресурсам. Поэтому стоить выбрать некоторое множество тегов, на которых будет произведена классификация.

4.1 Анализ часто используемых тегов

Изучив график распределения количества текстов по использованию самых часто встречающихся тегов Рис. 2, очевидно, что использовать такие теги как: “Текст”, “Длиннопост”и “[мое]”, не имеет смысла, так как они встречаются более чем в каждом втором тексте.

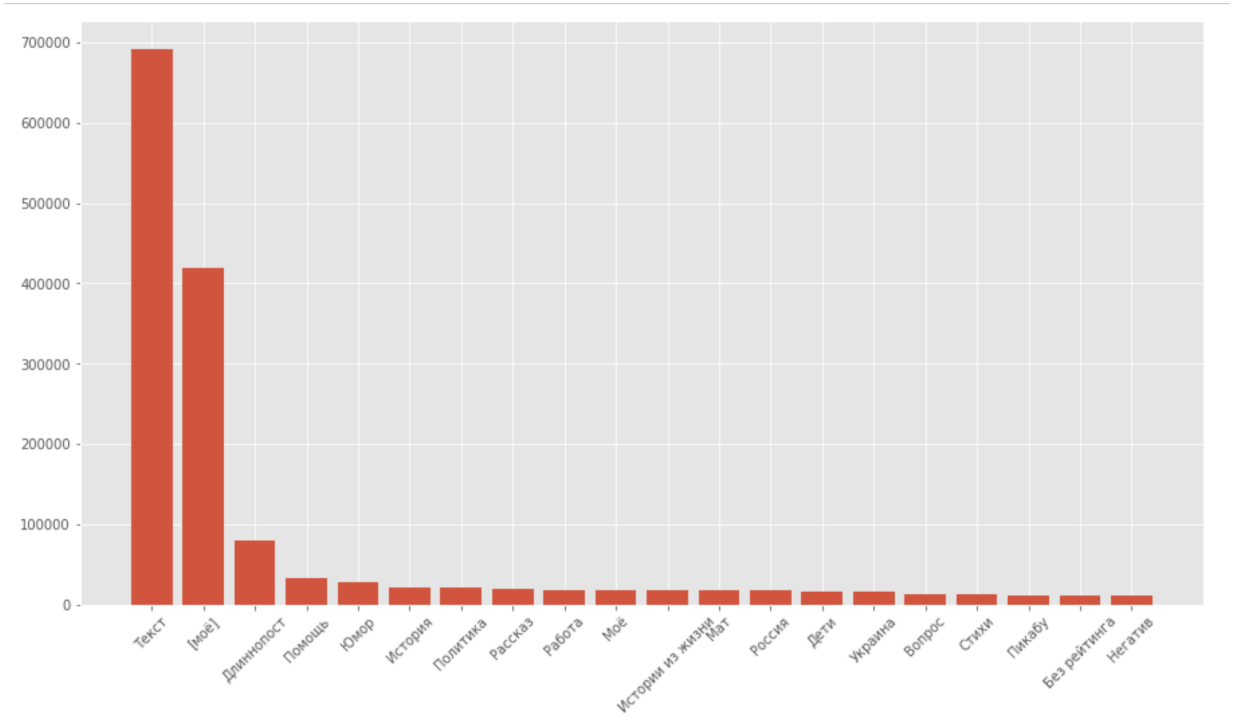
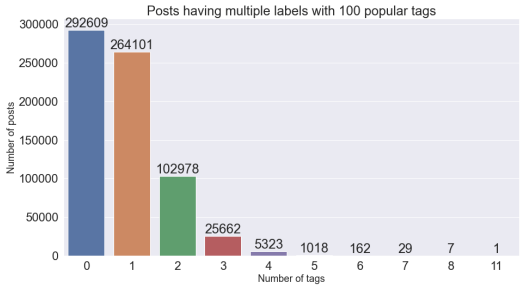
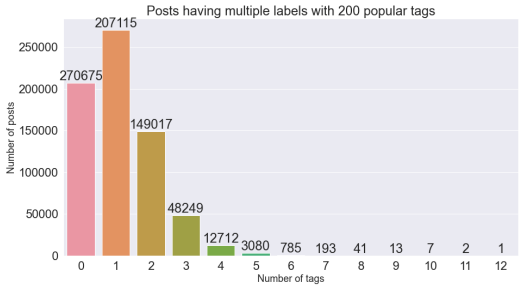


Рис. 2: Гистограмма распределения количества постов по часто встречающимся тегам.

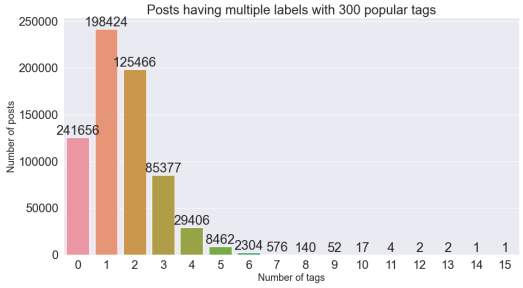
Поскольку основная задача в ходе работы именно мультилейбл классификация, стоит убедиться, что полученный датасет имеет достаточно постов, к которым нужно отнести несколько меток. На Рис. 3 представлена зависимость количества тегов и количества постов, при этом были выбраны 100, 200, 300, 400, 500 и 600 самых популярных тегов соответственно.



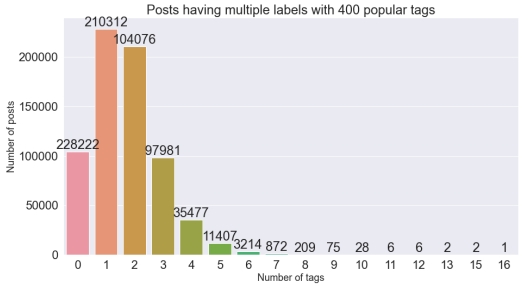
a) 100 самых популярных тегов



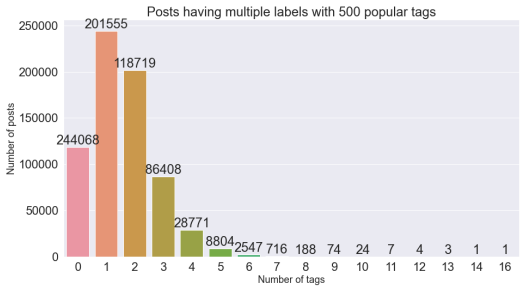
b) 200 самых популярных тегов



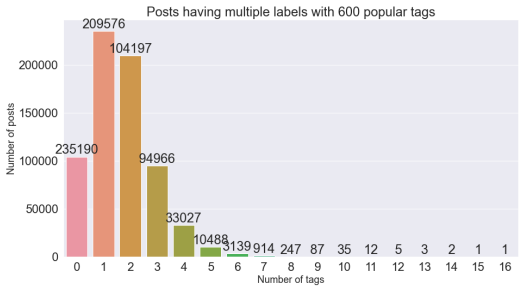
c) 300 самых популярных тегов



d) 400 самых популярных тегов



c) 500 самых популярных тегов



d) 600 самых популярных тегов

Рис. 3: Гистограмма зависимости количества постов от количества меток в каждом из них.

При 100 и 200 самых популярных тегов - значительная часть постов не содержит эти выбранные теги и не будет участвовать в классификации. Количество постов с двумя популярными тегами значительно больше при наборе из 300 и 400 тегах.

Если рассмотреть 500 самых популярных тегов, то достаточно много постов с двумя и более отмеченными тегами, при этом при 600 постах количество постов с двумя и более тегами меняется незначительно, поэтому в рамках работы будет использоваться набор из 500 самых часто используемых тегов на портале Пикабу.

Отметим также, что для удобства работы с алгоритмами обучения,

вместо хранения списка всех тегов, принадлежащих посту, каждому посту добавляется 500 полей, соответствующие выбранным меткам, которые заполняются “0”или “1”, характеризующими наличие или отсутствие определенного тега.

Глава 5. Предобработка текста

Работа с тестом достаточно трудоемкая задача, так как необходимо очистить текст: удалить пунктационные символы, удалить служебные части речи, привести слова в начальную форму [17].

5.1 Преобразование текста в числовой вектор

Модуль nltk позволяет загрузить списки стоп-слов для множества языков, в том числе и русского. Для подготовки текста использовали готовую функцию [18], которая с помощью регулярных выражений и библиотеки re [19] производит стемминг. Последовательно приведем слова в начальну форму. Отметим, что обработка текстов в датасете заняла около 40 минут времени.

Один из самых интуитивно понятных способов представить текст как набор чисел - создать словарь слов и использовать номер слова как его порядковый номер. Однако такой подход слабо выражает семантическую разность слов и их значимость в тексте.

TF-IDF является модификатором примитивного алгоритма. Он заключается в том, что если термин встречается несколько раз в одном или нескольких документах, то термин является существенным, и ему необходимо присвоить значение выше. Но когда термин встречается несколько раз во всех или в большинстве документов, этот термин считается типичным и имеет более низкое числовое значение [20]. Данный метод преобразования текста используется чаще всего при классификации с несколькими типами данных для обучения, а также для работы с формализованными документами, например, такими как резюме [21], так как при преобразовании не учитывается семантика слов [5].

Существует ряд подходов, использующих семантические значения слов, например Word-To-Vec. Такой способ преобразования текста в числовой вектор преобразует семантическую связь между словами в терминах векторных операций [22]. Таким образом, расстояние между векторами слов синонимов будет небольшим.

Современным подходом преобразования текста в вектор является уни-

версальный кодировщик предложений (Universal Sentence Encoder). Метод позволяет преобразовать тексты разного размера в высокоразмерные вектора фиксированной длинны [23], [24]. Такой подход, как и Word-To-Vec учитывает семантическую значимость, но не на уровне слов, а на уровне предложений. Модель USE была обучена на необработанных текстах, поэтому метод не нуждается в предобработке текста, что позволяет сэкономить время работы. Таким образом универсальный кодировщик предложений учитывает не только смысл слова, но и контекст всего предложения.

Изучив литературу и исследования в схожей области классификации пользовательских текстов, принято решение использовать универсальный кодировщик предложений. Такой подход обладает значительными преимуществами перед другими рассмотренными методами преобразования текстов.

Глава 6. Обучение

Преобразовав текстовые данные в наборы векторов, с помощью модели [25], доступной на платформе TensorFlow Hub, произведено разделение данных на тренировочные и тестовые в соотношении 8 к 2. Оценка качества предсказаний тегов, в построенной рекомендательной системе, выполнится с помощью тестовой выборки.

6.1 OneVsRestClassifier

Как уже говорилось ранее, данный подход разбивает задачу мультилейбл классификации на подзадачи бинарной классификации, используя метки как отдельные классификаторы [26]. Для реализации данного классификатора реализована последовательная бинарная классификация с помощью Логистической регрессии, используя библиотеку sklearn. Таким образом с помощью вероятностной модели построено 500 классификаторов.

6.2 BinaryRelevance

Для обучения набора однокомпонентных двоичных классификаторов был использован модуль BinaryRelevance из библиотеки skmultilearn, использующий наивный байесовский подход с помощью модуля GaussianNB.

6.3 Label Powerset

Используя модуль LabelPowerset и логистическую регрессию с помощью библиотеки sklearn, были построены классификаторы используя все комбинации меток.

6.4 ClassifierChains

Реализация метода построения цепочек классификатора так же как и реализация некоторых подходов содержит Логистическую регрессию.

6.5 BiGRU

Используя библиотеку keras [27] создается модель с сигмовидной функцию активации, для оптимизации используется метод стохастического градиентного спуска.

6.6 DistilBERT

Использование данной модели с начальными значениями, возможно с помощью библиотеки Transformers [28]. Модель содержит 6 слоев.

Глава 7. Анализ полученных данных

Далее будут описаны метрики для оценки качества классификатора, которые используются при бинарной классификации. Оценивая предсказание мультилейбл классификации, будем использовать среднее значение для всех классификаторов. В программной реализации используется библиотека sklearn.metrics.

7.1 Метрики оценки качества классификации

7.1.1 Accuracy

Одной из самых часто используемых метрик для оценки качества классификации является точность [29]. Ее значение вычисляется отношением количества правильно классифицированных данных к неправильным[30]. При этом к правильно классифицированным данным относятся как истинные положительные результаты, так и истинные отрицательные. Для подсчета всех данных суммируем также ложные отрицательные и положительные результаты. В программной реализации используется функция accuracy_score [31] из библиотеки sklearn.metrics.

7.1.2 Precision

Данная метрика представляет собой долю правильных прогнозов среди всех прогнозов определенного класса. Таким образом наилучший результат достигается, когда средняя точность равна единице. Чем больше значение этой метрики, тем точнее результат предсказания для конкретного класса [12].

7.1.3 Recall

Метрика характеризует долю правильных прогнозов среди всех прогнозов классификатора [32]. В рамках разработки рекомендательной системы для публикаций Пикабу, уделим внимание данной метрике, так как

при предсказании меток важно определить подходящую метку. Не предсказание верной метки, уменьшает числовое значение данной оценки [33].

7.1.4 F1-score

Метрика f1-score связывает предыдуще описанные метрики precision и recall, является их средним гармоническим.

7.2 Время обучения

В качестве сравнения алгоритмов будем также использовать время обучения моделей.

7.3 Результаты

Учитывая полученные результаты, в целом, модели, сводящиеся к бинарной классификации точнее предсказывают метку класса чем модели, использующие нейронные сети.

Сравнив результаты классических подходов, можно заметить, что метод использующий ансамбли бинарных классификаторов показал наихудшие результаты по выбранным метрикам, а также обучение заняло продолжительное время. Цепочки классификаторов показали результат несколько выше, особенно по ключевой для решаемой задачи метрике - recall.

Таблица 1: Результаты проведенных экпериментов.

Метод	Accuracy	Precision	Recall	F1-score	Время, s
One-Vs-Rest	0.933	0.786	0.876	0.828	1139
Binary Relevance	0.871	0.763	0.856	0.806	11155
Label Powerset	0.962	0.821	0.932	0.872	23412
Classifier Chains	0.893	0.856	0.761	0.805	1774
BiGRU	0.742	0.673	0.807	0.733	889
DistilBERT	0.643	0.505	0.616	0.555	11859

Сравнивая результаты, отметим, что подход, использующие возможные корреляции между метками имеет высокую точность, среди классических подходов. Учитывая полученные результаты можно сделать вывод,

что для обучения на выбранных данных для мультилейбл классификации подход Label-Powerset достаточно удачный.

Стоит напомнить, что для классификации использовались 500 различных меток, что является достаточно большим числовым значением для задачи классификации, так как при увеличении количества меток, количество их комбинаций растет экспоненциально [10]. Следовательно, при увеличении количества классов не возрастет вычислительная сложность метода, поэтому для его обучения потребовалось большое количество времени.

Высокую точность предсказания на используемых данных также показал подход One-Vs-Rest. Так как количество постов, содержащих только одну или две метки сравнимо велико, относительно постов, имеющих более трех меток, бинарная классификация успешно предсказывает отношение к классу.

Модели, использующая нейронные сети классифицировали тексты с наименьшей точностью. При этом, ключевое для задачи, значение метрики recall, у рекуррентных нейронных сетей выше, чем у предобученной модели на основе BERT.

Программная реализация

Рекомендательная система построена с помощью языка Python. Исходный код подготовки данных, анализа тегов и экспериментов доступен по ссылке [34].

Технические характеристики устройства, на котором выполнялась классификация: MacBook Pro (16-inch, 2019), 2,3 GHz Intel Core i9, 16 ГБ 2667 MHz DDR4.

Заключение

В рамках данной работы была построена рекомендательная система тегов, основанная на мультилейбл классификации текстов. Предложенную реализацию возможно использовать при написании поста на портале Пикабу, для ускорения времени работы автора над публикацией.

Поскольку Пикабу не позволяет скачать все статьи с портала автоматически, в ходе работы описан способ сбора данных, представляющий собой последовательное скачивание html-страниц, парсинг, предобработку полученных текстовых данных. Кроме того, произведено сравнение популярных способов преобразования текста в числовой вектор.

В ходе работы была решена задача мультилейбл классификации пользовательских постов по тегам. Сравнив классические подходы классификации, сводящиеся к бинарной, можно сделать вывод, что точность предсказания достаточно высока, при небольшом количестве времени выполнения. Однако такие подходы нуждаются в предобработке текста.

Подходы для классификации с использованием нейросетевых предобученных моделей, предсказали метки несколько хуже, чем классические подходы. Преимущества таких подходов в работе с языковыми моделями, которые учитывают семантическое сходство на уровне слов, предложений и текстов.

В качестве эксперимента произведено сравнение: качества предсказания по основным метрикам точности и времени обучения нескольких подходов. При этом, высокие значения, наиболее значимых оценок для решаемой задачи - recall и accuracy, достигаются классификацией с помощью подходов One-Vs-Rest и Label Powerset. Можно сделать вывод, что для рекомендательной системы тегов, необходимо использовать результаты именно этих классификаторов.

В дальнейшем при работе с этими данными следует провести дополнительные эксперименты, включающие в себя как уменьшение, так и увеличения набора меток для классификации, изменение параметров обучения моделей нейронных сетей, а также в целом использование других предобученных моделей.

Для модификации работы над поставленной задачей, возможно использовать персональную рекомендацию, и предлагать пользователям теги, основываясь, не только на тегах, проставленных на похожих текстах, но и в целом на тегах, используемых конкретным пользователем.

Список литературы

[1] Subramaniyaswamy V., Vijayakumar V., Indragandhi V., Logesh R. Data Mining-Based Tag Recommendation System: An Overview // WIREs Data Mining Knowl Discov. 2015. P. 87-112.

[2] Alsukhni B. Multi-label arabic text classification based on deep learning // 12th International Conference on Information and Communication Systems (ICICS). 2021. P. 475–477.

[3] Li Z., Shang W., Yan M. News text classification model based on topic model // IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). 2016. P. 1–5.

[4] Jindal R., Taneja S. Ranking in multi label classification of text documents using quantifiers // IEEE International Conference on Control System, Computing and Engineering (ICCSCE). 2015. P. 162–166.

[5] Zhang Y. , Li S. Technical Analysis of Multi -Text Video Standardization Based on Tag System // 2020 International Conference on Culture-oriented Science Technology (ICCST). 2020 P. 50-53.

[6] Nguyen H. , Richards R., Chan C. , Liszka K. "RedTweet: Recommendation engine for Reddit // ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). 2015. P. 1381-1388.

[7] Ding L., Zheng Y. Improve E-Commerce Recommendation by Classification Tree and Fuzzy Sets // International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI). 2016. P. 457-460.

[8] Quintanilla E. , Rawat Y. , Sakryukin A. , Shah M., Kankanhalli M. Adversarial Learning for Personalized Tag Recommendation // IEEE Transactions on Multimedia. 2021. Vol. 23. P. 1083-1094.

[9] Solving Multi-Label Classification problems (Case studies included) [Электронный ресурс]: `https://www.analyticsvidhya.com/blog/2017/08/`

introduction-to-multi-label-classification/ (дата об- ращения 18.12.2021)

[10] Deep dive into multi-label classification..! (With detailed Case Study) [Электронный ресурс]: https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-3384c40229bff (дата обращения 19.12.2021)

[11] Lakhdhar Y., Rekhis S. Machine Learning Based Approach for the Automated Mapping of Discovered Vulnerabilities to Adversial Tactics // IEEE Security and Privacy Workshops (SPW). 2021. P. 309-317.

[12] Maia M., Sales J., Freitas A., Handschuh S., Endres M. A Comparative Study of Deep Neural Network Models on Multi-Label Text Classification in Finance // IEEE 15th International Conference on Semantic Computing (ICSC). 2021. P. 183-190.

[13] Di L., Xiushuang Y., Ling X. Design of natural language model based on BiGRU and attention mechanism // International Conference on Networking, Communications and Information Technology (NetCIT). 2021. P. 191-195.

[14] Oliseenko V., Tulupyeva T. Neural Network Approach in the Task of Multi-label Classification of User Posts in Online Social Networks // 2021 XXIV International Conference on Soft Computing and Measurements (SCM). 2021. P. 46-48.

[15] Bai J., Cao R., Ma W., Shinnou H. Construction of Domain-Specific DistilBERT Model by Using Fine-Tuning // 2020 International Conference on Technologies and Applications of Artificial Intelligence (TAAI). 2020. P. 237-241.

[16] Bambroo P., Awasthi A. LegalDB: Long DistilBERT for Legal Document Classification // International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). 2021. P. 1-4.

[17] Rajabi Z., Shehu A., Uzuner O. A Multi-channel BiLSTM- CNN Model for Multilabel Emotion Classification of Informal Text // 2020 IEEE 14th International Conference on Semantic Computing (ICSC). 2020. P. 303-306.

[18] GitHub. gistfile1.py [Электронный ресурс]: https://gist.github.com/Kein1945/9111512 (дата обращения 02.12.2021).

[19] Regular expression operations [Электронный ресурс]: https://docs.python.org/3/library/re.html (дата обращения 03.01.2022).

[20] Dalaorao G. A., Sison A. M., Medina R. P. Integrating collocation as TF-IDF enhancement to improve classification accuracy // IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA). 2019. P. 282–285.

[21] Bagul D, Barve S. A novel content-based recommendation approach based on LDA topic modeling for literature recommendation // 6th International Conference on Inventive Computation Technologies (ICICT). 2021. P. 954-961.

[22] Parolin E. , Salam S. , Khan L. , Brandt P. , Holmes J. Automated Verbal-Pattern Extraction from Political News Articles using CAMEO Event Coding Ontology // IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity). 2019. P. 258-266.

[23] Kumar N. , Kumar S. , Dev A., Naorem S. Leveraging Universal Sentence Encoder to Predict Movie Genre // 7th International Conference on Advanced Computing and Communication Systems (ICACCS). 2021. P. 1013-1018.

[24] Use-cases of Google's Universal Sentence Encoder in Production [Электронный ресурс]: https://towardsdatascience.com/use-cases-of-googles-universal-sentence-encoder-in-production-dd5aaab4fc15 (дата обращения 15.02.2022)

[25] universal-sentence-encoder [Электронный ресурс]: https://tfhub.dev/google/universal-sentence-encoder/4 (дата обращения 10.03.2022)

[26] Tao W., Yongjia J., Xiangsheng R. A novel two-level One-vs-Rest classifier // 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE). 2019. P. 645–648.

[27] About Keras [Электронный ресурс]: https://keras.io/about/ (дата обращения 13.03.2022)

[28] Transformers [Электронный ресурс]: https://huggingface.co/docs/transformers/main/en/index (дата обращения 16.03.2022)

[29] Understanding Data Science Classification Metrics in Scikit-Learn in Python [Электронный ресурс]: https://towardsdatascience.com/understanding-data-science-classification-metrics-in-scikit-learn-in-python-3bc336865019 (дата обращения 22.12.2021)

[30] Classification: Accuracy [Электронный ресурс]: https://developers.google.com/machine-learning/crash-course/classification/accuracy (дата обращения 19.12.2021)

[31] Accuracy classification score [Электронный ресурс]: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html (дата обращения 18.12.2021).

[32] Precision, Recall, Accuracy, and F1 Score for Multi-Label Classification [Электронный ресурс]: https://medium.com/synthesio-engineering/precision-accuracy-and-f1-score-for-multi-label-classification-34ac6bdfb404 (дата обращения 12.11.2021)

[33] Основные метрики задач классификации в машинном обучении [Электронный ресурс]: https://webiomed.ru/blog/osnovnye-metriki-zadach-klassifikatsii-v-mashinnom-obuchenii/ (дата обращения 19.01.2022)

[34] GitHub. Tag-recommendation-system [Электронный ресурс]: https://github.com/alexy247/tag-recommendation-system