САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МИРОНОВ Андрей Александрович

Выпускная квалификационная работа

Разработка метода извлечения ключевых фраз из постов социальной сети ”Твиттер”

Бакалавриат

Направление: 01.03.02 ”Прикладная математика и информатика”

Основная образовательная программа СВ.5005 ”Прикладная математика, фундаментальная информатика и программирование”

Научный руководитель:
Старший преподаватель
Кафедра технологии программирования
Попова Светлана Владимировна

Рецензент:
Старший преподаватель
Кафедра космических технологий и прикладной астродинамики
Давыденко Александр Александрович

Санкт-Петербург
2019 г.

Содержание

1. Введение

С постоянным увеличением потоков информации, наполняющих сеть Интернет, всё более актуальной становится задача извлечения из единицы контента некоторой ключевой части, позволяющей с высокой точностью определить основную мысль данного текста. Эту ключевую часть обычно называют ключевой фразой (*keyphrase*).

С помощью ключевых фраз конечный пользователь может получить концентрированную и исчерпывающую информацию об основных мыслях или тематике того или иного текста. Такую задачу постоянно вынуждены решать поисковые системы, новостные аггрегаторы и ресурсы, анализирующие мнение пользователей социальных сетей по какому-либо вопросу. Существует множество видов контента, из которого можно извлечь ключевую часть: это могут быть веб-страницы, научные статьи, книги или даже фильмы. В данной работе будут рассмотрены методы извлечения ключевых фраз из коротких(объёмом до 140 символов) текстов, в частности постов, размещаемых в социальной сети Twitter.

В то время как методы извлечения ключевых фраз из текстов на английском языке достаточно глубоко изучены(как для текстов большого обьёма, так и для небольших текстов, например твитов), подобных исследований для русскоязычных текстов по-прежнему очень немного. Следует обратить внимание на то, что мы отделяем задачу извлечения ключевых фраз от таких задач как извлечение ключевых слов(на эту тему есть достаточно исследований для русского языка, например [2, 4, 6]), извлечения терминологии предметной области и задачи извлечения коллокаций(словосочетаний, имеющих признаки синтаксически и семантически целостной единичы).

Из-за ограничения на максимальное количество символов стандартные методы, подходящие для работы с текстами большого объема, могут показывать неудовлетворительные результаты на постах из Твиттера или подобных ему ресурсов.

Изложенные выше обстоятельства и побудили меня избрать темой выпускной квалификационной работы извлечение ключевых фраз из русскоязычных постов в социальных сетях, т.к данная тема в настоящее время представляет как коммерческий, так и научный интерес.

2. Постановка задачи

Цель исследования - повышение качества извлечения ключевых фраз из постов, размещённых в социальной сети ”Твиттер”(или любой другой, удовлетворяющей ограничению в 140 символов для каждой записи) путём адаптации к русскому языку методов, использующихся для извлечения ключевых фраз из англоязычных твитов.

Для достижения поставленной цели в исследовании решались следующие задачи:

- Анализ литературы в данной предметной области
- Выбор наиболее подходящих инструментов для работы
- Разработка русскоязычной обучающей коллекции
- Реализация алгоритмов извлечения ключевых фраз
- Тестирование и оценка качества этих алгоритмов
- Анализ полученных результатов и сопоставление их с другими результатами, достигнутыми в данной области
- Подготовка выводов по итогам проекта

Также одной из ключевых задач проекта является проверка совместимости методов решения задачи извлечения ключевых фраз из коротких текстовых документов, изначально разработанных и применяемых для работы с англоязычным контентом.

3. Обзор литературы

Как уже было сказано во введении: большая часть литературы по данному вопросу существует исключительно на английском языке и представляет собой исследования на тему извлечения ключевых фраз из английских текстов. Хотя, конечно, существуют подобные исследования и для русскоязычных текстов, в частности [6]. Однако в данной работе анализируется задача извлечения ключевых фраз из достаточно крупных документов, которая имеет несколько иную специфику, нежели работа с короткими текстами.

Отдельной проблемой, о которой заявляют исследователи данного вопроса, является отсутствие достаточно больших корпусов для обучения моделей, причём данная проблема актуальна как для русского, так и для английского языка. Помимо прочего в данном случае возникает затруднение с созданием подобного датасета и оценкой качества работы алгоритма по причине субъективности объекта исследования, в связи с этим исследователи вынуждены полностью полагаться на мнение экспертов о качестве разметки [3, с. 71]. По этой причине часто для решения данной задачи используют методы, не требующие наличия какого-либо обучающего датасета.[1, 3] Однако(как утверждается в [1]) для подобных методов необходимы документы намного более крупные, чем твиты, не превосходящие по длине 140 символов, поэтому использование подобных методов при работе с короткими текстами даст намного меньшую точность.

Также у исследователей нет единого взгляда на классификацию методов решения данной задачи. Авторы в [6] делят существующие методы выделения ключевых фраз на 2 группы: статистические и гибридные. Статистические основаны на частоте употребления терминов, а гибридные представляют собой комбинацию статистических и лингвистических методов. По мнению авторов качество гибрибных методов превосходит качество статистических.

Помимо разделение на статистические и гибридные также выделяют обучаемые *(supervised)* и необучаемые *(unsupervised)*.[2] В частности к обучаемым относятся методы, использующие нейронные сети, причём количество исследований в области применения нейронных сетей в решении задачи выделения ключевых фраз резко увеличилось в последние 10 лет[2].

Более развёрнутая классификация подразумевает выделение четырёх типов:

- не требующие обучения простые статистические методы
- лингвистические методы
- методы, основанные на машинном обучении
- их всевозможные комбинации[13]

Отправной точкой моей работы является статья за авторством четырёх исследователей из университета Фудана (КНР), разработавших новый метод извлечения ключевых фраз из постов Твиттера, который превзошёл по точности другие основные методы решения поставленной задачи[12]. Описанный в их работе метод основывается на построении нейронной сети с соединёнными слоями на выходе, так называемой *joint-layer network*, в которой один из слоёв отвечает за поиск ключевых слов, другой - за поиск ключевых фраз, а результатом является комбинация двух, описанных выше слоёв. Такая архитектура нейросети позволяет наиболее точно предсказать является ли то или иное слово или словосочетание ключевым словом или частью ключевой фразы твита, и если является, то какой конкретно(начало, середина или конец)

Однако, также как и в предыдущих случаях, исследования, представленные в данной работе проводились с использованием корпуса англоязычных твитов, поэтому нет уверенности в том, что результаты полученные учёными из Фудана будут релевантны на твитах на другом языке(особенно учитывая, тот факт, что русский язык обладает более богатой морфологией и меньшей структурированностью, что затрудняет анализ предложений)[6, с. 77].

4. Выбор инструментов

4.1. Язык программирования и среда разработки

Для обработки данных и построения нейронной сети мной был выбран язык программирования Python версии 3.7 и интерактивная среда разработки Jupyter Notebook, т.к благодаря большому количеству компонентов для машинного обучения и построения нейросетей данная комбинация является наиболее популярной в области академических исследований нейронных сетей[14]. Выбор обусловлен широкими возможностями языка в плане работы с векторами и матрицами(библиотека numpy https://www.numpy.org/) а также библиотеками для работы с текстовыми данными(https://pypi.org/project/pymystem3/).

4.2. Фреймворк для машинного обучения

Для создания и обучения нейронных сетей как правило используются фреймворки, которые позволяют разработчику не обращать внимания на многие низкоуровневые аспекты конструирования моделей для обучения и вместо этого сосредоточиться на конкретных параметрах, определяющих архитектуру и метод обучения нейросети. Для данной работы была выбрана библиотека TensorFlow от компании Google, и её реализация для языка программирования Python.

4.3. Инструменты для морфологического анализа

В качестве инструмента для морфологического анализа слов, содержащихся в датасете была выбрана технология MyStem, разработанная сотрудниками ООО “Яндекс” Ильёй Сегаловичем и Виталием Титовым(https://tech.yandex.ru/mystem/) и её обёртка для языка Python (https://pypi.org/project/pymystem3/). Она позволяет произвести *лемматизацию* слов, т.е приведение слова к его начальной форме[10]. Это позволяет сократить размер словаря, не уменьшив при этом его семантической информативности, т.к все лексемы в различных формах приводятся к единой начальной форме.

4.4. Векторное представление слов

Т.к нейронные сети способны принимать на вход только числовые данные, то перед началом работы появляется необходимость представления лексем, из которых состоят тексты(в данном случае твиты) в цифровом виде. В качестве метода такого представления мной была выбрана технология word2vec Томаша Миколова[8]. Она позволяет получить представление слов в виде векторов необходимой размерности(сам автор рекомендует использовать размерность в 300 компонент), причём данные вектора являются некоторым отражением смысла данной лексемы и её положения в семантическом пространстве. Например: евклидово расстояние между векторами для слов МОСКВА и САНКТ-ПЕТЕРБУРГ будет меньше, чем между векторами для слов МОСКВА и НЬЮ-ЙОРК. Ещё одной особенностью данного метода представления слов является отношение векторов, отражающее их реальное значение: например разность между векторами для слов РОССИЯ и МОСКВА будет близка к разности между векторами для слов ФРАНЦИЯ и ПАРИЖ, что для человека является абсолютно логичным, однако среди программных методов векторного представления слов такой уровень качества впервые был достигнут именно word2vec[8].

Ещё одним аргументом в пользу выбора именно этого инструмента является его независимость от языка, т.к для обучения модели используется обучение без учителя(*unsupervised learning*) основанное на частоте

встречаемости слов в одном *window-size* в корпусе, использованном для обучения.

В данном случае была использована уже готовая модель, обученная волонтёрами проекта Rusvectores на корпусе из русскоязычных новостей, собранных с более чем 1500 новостных ресурсов в феврале-марте 2019 года, имеющая объём в 2,6 миллиарда слов. Модель обучена методом Continious Skipgram и даёт возможность получить векторное представление для 249 318 слов. Модель можно найти по ссылке: http://vectors.nlpl.eu/repository/11/184.zip.

4.5. Оценка качества извлечённых фраз

Точность работы нейронной сети оценивалась с помощью нескольких метрик, таких как *Accuracy, Precision, Recall и F-Measure*. Объясним их значение более подробно. Accuracy(простая точность) - наиболее понятная и очевидная метрика из данного списка. Представляет собой простое отношение верно угаданных результатов к размеру всего датасета, т.е определяется по формуле:

$$Accuracy = \frac{P}{N},$$

где P - количество верных результатов(в нашем случае - количество твитов с вернно угаданными ключевыми словами, т.е таких твитов, в которых алгоритм верно и в нужном порядке пометил все слова, составляющие ключевую фразу и полностью верно пометил все, которые не являются частями ключевой фразы), а N - общий размер выборки.

Precision(точность) представляет собой более сложную метрику и определяется как доля документов действительно принадлежащих данному классу относительно всех документов которые система отнесла к этому классу(в нашем случае определяется как доля слов, действительно принадлежащих данному классу, где класс это местоположение слова в ключевой фразе, таких классов в контексте данной работы пять: не входящие в ключевую фразу, первые в ключевой фразе, последние в ключевой фразе, середина(не первые и не последние) и одиночные ключевые слова). Это значение определяется формулой:

$$Precision = \frac{TP}{TP + FP},$$

где TP - количество слов, верно отнесённых моделью к соответствующему классу(истинно-положительных решений), а FP - количество слов, неверно отнесённых моделью к тому же классу(ложно-положительных). Далее рассмотрим метрику Recall(полнота). Она определяется формулой:

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

где FN - ложно-отрицательные решения, т.е количество случаев, в которых слово, принадлежащее некоторому классу, не было помечено таковым нейросетью. Наконец, F-Measure представляет собой гармоническое среднее значений Precision и Recall, что позволяет найти наилучший алгоритм с учётом обоих значений данных метрик. В нашем случае Precision и Recall являются равнозначными(т.е одна не имеет приоритета над другой) и вычисляются как

$$F = 2\frac{Precision \times Recall}{Precision + Recall}$$

Т.к в данной работе рассматривается классификация слов(т.е задача отнесения слов к тому или иному классу), при том, что в нашей задаче необходимо распределить все слова между пятью классами, то Precision

и Recall будем определять как

$$Precision_{micro-average} = \frac{\sum\limits_{n} TP_n}{\sum\limits_{n} TP_n + \sum\limits_{n} FP_n}$$

$$Recall_{micro-average} = \frac{\sum\limits_{n} TP_n}{\sum\limits_{n} TP_n + \sum\limits_{n} FN_n},$$

где TP_n означает количество слов, верно отнесённых моделью к классу n, FP_n - количество слов, неверно отнесённых к классу n, а FN_n - количество слов, принадлежащих классу n, которые не были помечены таковыми моделью.

Для вычисления значений данных метрик перед началом обучения были выделены 1000 случайных твитов, которые были удалены из обучающей выборки, чтобы при тестировании система встретила их впервые, и обозначены как *тестовый датасет*. Далее на всех этапах обучения показатели качества работы нейросети определялись на основе результатов, показанных на этой тестовой выборке.

5. Разработка русскоязычной коллекции документов, размеченой ключевыми фразами

5.1. Поиск датасета

От первоначальной идеи самостоятельного создания датасета пришлось впоследствии отказаться из-за большого количества сложностей, сопряжённых с такой задачей(ограничение Twitter API итд.). Идеальным вариантом для данной задачи был бы датасет из твитов с размеченными ключевыми фразами на русском языке, но, к сожалению, как уже было сказано выше такого датасета на данный момент не существует. В связи с этим было принято решение использовать для исследований датасет созданный аспирантом института систем информатики им. А.П. Ершова СО РАН Юлей Рубцовой в рамках диссертационного исследования и впоследствии выложенный ей в свободный доступ в сеть интернет(на сайте https://study.mokoron.com/). Датасет представляет собой SQL-дамп, содержащий 17,639,674 записей из русскоязычного сегмента социальной сети “Twitter” , собранных в период с конца ноября 2013 года до конца февраля 2014 года[5]. После первичного ознакомления с датасетом мной было принято решение о возможности его использования в качестве источника данных для исследования. В дальнейшем датасет был препарирован для соответствия формату, который способна принимать на вход нейросеть(т.е сделана разметка для ключевых фраз, позволяющая использовать его для обучения нейронных сетей).

5.2. Очистка датасета

На этом этапе исследований предварительная обработка данных производилась способом, абсолютно идентичным тому, который был представлен в [12].

Сначала из корпуса твитов объёмом в 17,639,674 твита были выделены только твиты, содержащие хотя бы один хэштег, и не содержашие символа “@” , означающего, что твит является ответом на некоторый другой твит. Это сделано из-за того, что для обучения требуются только твиты, несущие в себе некоторую не вырванную из контекста информацию, и не являющиеся частью диалога пользователей. Таких твитов в датасете оказалось 1510835 штук. Затем из оставшейся части корпуса были удалены все URL-ссылки и слова содержащие символы, не подходящие под определение русских букв, латинских букв или цифр, удалены все знаки препинания, а также все слова переведены в нижний регистр.

Таблица 1. Статистические характеристики датасета

N	W	T	$\bar{N}$	$\tilde{N}$
17 639 674	42653	23853	9.638	1.0

Здесь N общее количество твитов в исходном датасете, W - словарь датасета(количество уникальных слов), T - количество твитов после обработки и отделения неподходящих для обучения нейросети, $\bar{N}$ - среднее количество слов в твите, $\tilde{N}$ - среднее количество хэштегов в твите

5.3. Имплементация алгоритма разметки датасета ключевыми фразами

Дальнейшее конструирование обучающего корпуса строилось на предположении о том, что в большинстве случаев хэштег твита выражает его основную мысль, т.е является его ключевым словом или ключевой фразой, в случае если хэштег содержит несколько слов. Например рассмотрим твит ”завтра ну сходите со мной кто нибудь на #голодныеигры ну давайте ну пожалуйста”. Очевидно, что фраза ”голодныеигры”может быть рассмотрена как ключевая для данного твита, т.к является основным объектом, о котором сообщает нам твит. Другие примеры твитов из размеченного датасета:

Таблица 2. Примеры твитов из размеченного датасета

Твит	Ключевая фраза
но то что 3 измениться очень врядли биатлон гаспарин виткова старых это значит второй человек из наших попадает в призы на кубке мира	биатлон
я играю в snark busters 2 присоединяйтесь	snark busters 2
все свое ношу с собой хорошая штука все же dropbox кто еще не в теме регистрируемся и получаем бонус	dropbox
на боях fight nights в москве тяжеловес михаил мохнаткин только что нокаутировал именитого голландца валентайна оверима	боях fight nights
был бы пилотом через 2 года ушел бы на пенсию старость не радость да	старость не радость

5.4. Проверка автоматически извлечённых ключевых фраз

Чтобы проверить насколько данное предположение, которое является ключевым для всей работы, можно считать общим правилом и перенести на всё множество твитов, было проведено проверочное исследование.

Его суть заключалась в том, что из всего корпуса твитов, подходящих под указанные выше ограничения, были случайным образом выделены 1000 твитов, из которых были извлечены ключевые фразы согласно данному предположению, другими словами - хэштеги были обозначены как их ключевые фразы. Затем данный список твитов и их ключевых фраз был выдан трём волонтёрам, которым, с помощью специально написанного для данной цели приложения(Приложение 1), предлагалось оценить точность работы данного метода. В случае, если указанная фраза, по мнению волонтёра однозначно может быть названа ключевой для данного твита - волонтёрам предлагалось поставить оценку ”2”. В случае, если представленная фраза не может быть названа ключевой с полной уверенностью, волонтёрам предлагалось поставить оценку ”1”. Наконец, если по мнению волонтёра выделенная фраза абсолютно не является ключевой - ему предлагалось поставить оценку ”0”. Если же из твита невозможно выделить ключевую фразу(чаще такое происходило если твит набран кирилицей, но не на русском языке, например на монгольском), волонтёрам предлагалось отметить твит, как имеющий ”неприемлемое содержание”. Полученные в результате исследования данные представлены на диаграмме(рис. 3).

Таким образом получается, что в среднем волонтёры оценили 42% извлечённых ключевых фраз на максимальную оценку ”2”и, помимо этого, в среднем 43,9% извлечённых ключевых фраз получили оценку ”1 т.е волонтёры посчитали, что они в целом отражают основную мысль твита. Абсолютно не совпадающи-

ми с основной мыслью твита волонтёры признали лишь 10,3% выделенных ключевых слов и фраз. И совсем малая доля(3,6% твитов от общего числа) по мнению волонтёров не поддаётся оценке. На основе результатов, полученных в ходе данного исследования, можно сделать вывод о том, что в 42% случаев хэштег полностью является ключевой фразой для твита, и, помимо этого, в 43,9% случаев хэштег можно квалифицировать как фразу скорее ключевую, чем нет, для данного твита. Т.е в сумме получаем, что в 85,9% случаев такой способ выделения ключевых фраз из твитов даёт верный результат и данный способ выделения ключевых слов и фраз может быть использован для построения обучающего датасета для нейронной сети.

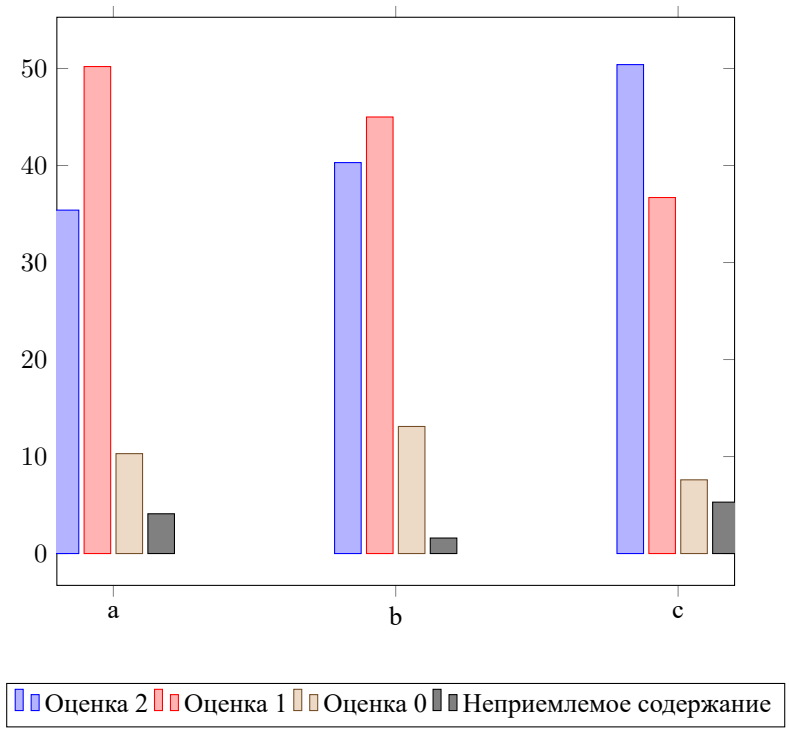

Рис.3 Диаграмма распределения оценок волонтёров.

Под *a, b* и *c* здесь понимаются обозначены первый, второй и третий волонтёры соответственно.

Затем, приняв во внимание тот факт, что пользователи социальной сети “Твиттер” используют хэштеги, написанные в одно слово, даже если в хэштеге содержится некоторая фраза(например ”голодныеигры”), было необходимо разделить эти сочетания букв на отдельные слова, чтобы в дальнейшем иметь возможность представить их в векторном виде, отражающем их значение. Для этого мной был разработан следующий алгоритм:

Algorithm 1 Выделение слов из хэштега

```
function EXTRACTWORDSFROMHASHTAG(hashtag)
    if rusWords contains hashtag || rusSurnames contains hashtag then
        return hashtag
    end if
    while length(hashtag) ≥1 do
        for i ← 0, length(hashtag) + 1 do
            if rusWords contains hashtag[0 : i] || rusSurnames contains hashtag[0 : i] then
                word ←hashtag[0 : i]
```

```
            end if
        end for
        words.append(word)
        hashtag ←hashtag[length(word) : length(hashtag)]
    end while
    return words
end function
```

В качестве входного значения функция принимает хэштег без символа #, и возвращает массив входящих в него слов. В случае, если хэштег квалифицирован алгоритмом, как состоящий из одного слова, выходным значением будет массив из одного элемента. Если нет, то алгоритм ищет в хэштеге наибольшую последовательность символов, пока не найдёт такую, которая присутствует в *rusWords* или *rusSurnames*, т.е такую, которая является словом. После нахождения слова в хэштеге найденная часть отсекается, а над оставшейся частью хэштега проводятся те же манипуляции. Например в хэштеге “голодныеигры” сначала будет найдено слово “голод”, однако, т.к слово “голодные” является наибольшей последовательностью символов, начиная с первого, которую можно характеризовать как слово, то именно слово “голодные” будет выделено, как первое слово данного хэштега. Оставшаяся часть хэштега(“игры”) присутствует в *rusWords*, поэтому это слово будет выделено, как второе в хэштеге. Таким образом хэштег “голодныеигры” алгоритм разделяет на массив из двух слов “голодные” и “игры”. Переменными *rusWords* и *rusSurnames* в данном алгоритме обозначены коллекции русских слов и русских фамилий соответственно, найденные мной в сети Интернет по адресу https://github.com/danakt/russian-words. В коллекции *rusWords* содержатся 1 601 915 слов русского языка во всевозможных формах, а в коллекции *rusSurnames* - 1 486 681 русская фамилия во всех падежах.

Далее для всех твитов и их ключевых фраз была произведена лемматизация с помощью инструмента *MyStem*, описанного в п. 4.3 и для лемматизированных твитов был составлен словарь, в котором каждой основе присваивался свой уникальные порядковый номер.

5.5. Обработка коллекции для получения тестовой и обучающей выборок

На данном этапе подготовки данных были сформированы массивы, непосредственно подающиеся на вход нейросети: массив включений слов, массив меток для ключевых слов и для ключевых фраз. Например, если твит включает в себя слова, имеющие в словаре индексы 1, 2, 3, 4 и 5, где ключевыми являются 2, 3 и 4 слова, то массив включений слов и метки для обучения будут выглядеть так:

1	2	3	4	5

Рис. 4(а) - Твит в виде массива включений слов

0	1	1	1	0

Рис. 4(б) - Разметка для ключевых слов

0	1	2	3	0

Рис. 4(в) - Разметка для ключевых фраз

Также на этом этапе коллекция разделена на тестовую и обучающую выборки: из всего датасета были случайным образом выделены 1000 твитов, которые не подавались на вход нейросети при обучении и использовались только при проверке качества работы модели.

При этом стоит отметить что при данном способе разметки датасета в каждом твите будет выделена ровно одна ключевая фраза. Таким образом и нейронная сеть, обученная на таком датасете, будет искать в каждом твите одну и только одну ключевую фразу.

6. Извлечение ключевых фраз из твитов с помощью joint-layer neural networks

6.1. Общие сведения о нейронных сетях

Нейронная сеть — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Нейронная сеть состоит из узлов, называемых нейронами, которые составляют слои нейронной сети и представляют собой простейшие процессоры, которые занимаются обработкой поступающих сигналов и передачей их следующим в цепочке нейронам.

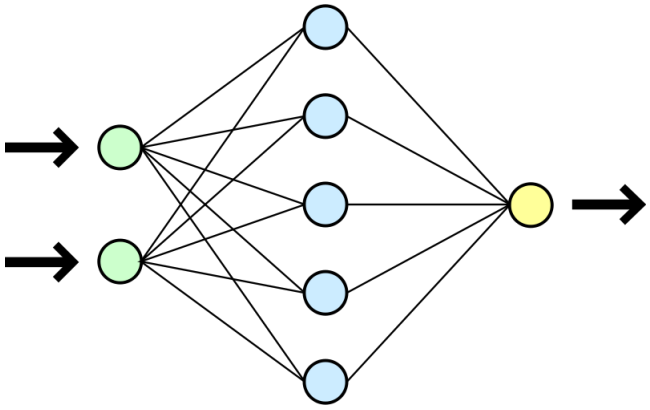

Рис. 1 Схема простейшей нейросети. Здесь зелёным цветом обозначены входные нейроны, синим - нейроны скрытого слоя, а жёлтым - выходной нейрон.

Параметр, определяющий, насколько сильно значение некоторого нейрона влияет на выходное значение(например значение на которое умножается сигнал идущий из нейрона A в нейрон B) называют *весом*. Матрица таких параметров для некоторого скрытого слоя образует *матрицу весов скрытого слоя*. Обучение нейронной сети представляет собой постепенную корректировку весов скрытых слоёв нейронной сети на основе разницы между каждым предсказанным и верным(поданным на вход нейронной сети в составе обучающего датасета) значениями. Например элемент a_{ij} матрицы весов **U** является значением, на которое умножается сигнал при переходе от нейрона i к нейрону j. Состоянием слоя l в момент времени t называется такая величина h_t^l, которая имеет величину значений нейронов слоя l в момент времени t(т.е например для входных данных $x(t)$). В начальный момент времени состояние скрытого слоя нейросети равно 0($h_0 = 0$).

6.2. Описание joint-layer neural networks

Рекуррентная нейронная сеть с совмещёнными слоями(*Joint-layer Recurrent Neural Networks*) является модификацией сложенной рекуррентной нейронной сети (*Stacked Recurrent Neural Network*) с двумя скрытыми слоями, т.к такая архитектура позволяет лучше приспособиться к поставленной выше задаче. Joint-Layer RNN имеет 2 выходных слоя и результирующий слой, учитывающий результаты, получаемые на обоих выходных слоях.

Рассмотрим Stacked RNN, состоящую из L слоёв и имеющую выходной слой для каждого скрытого слоя. В этом случае *l*-й слой определяется как:

$$h_t^l = f_h(\mathbf{h}_t^{l-1}, \mathbf{h}_{t-1}^l) = \phi_l(\mathbf{U}^l \mathbf{h}_{t-1}^l + \mathbf{W}^l \mathbf{h}_t^{l-1}),$$

где h_t^l является состоянием скрытого слоя *l* в момент времени *t*. $\mathbf{U}^l$ и $\mathbf{W}^l$ являются матрицами весов для данного слоя в момент времени *t* - 1 и для предыдущего слоя в момент времени *t* соответственно. При *l* = 1 это значение вычисляется как $\mathbf{h}_t^0 = x_t$. ϕ_l это поэлементная нелинейная функция, например сигмоид. Значения для выходного слоя с номером *l* вычисляются как:

$$\hat{\mathbf{y}}_t^l = f_o(\mathbf{h}_t^l) = \phi_l(\mathbf{V^l}\mathbf{h}_l^t),$$

где $\mathbf{V^l}$ это матрица весов для скрытого слоя $\mathbf{h}_l^t$. ϕ_l также может быть поэлементной нелинейной функцией, например *softmax*.

Joint-layer RNN это расширение для Stacked RNN с двумя скрытыми слоями. В момент времени *t* тренировочное значение x_t является совокупностью значений для элементов, находящихся внутри рассматриваемого окна(*window-size*). В данной работе в качестве таких значений используются векторные представления слов, составленные с помощью метода *word2vec*.

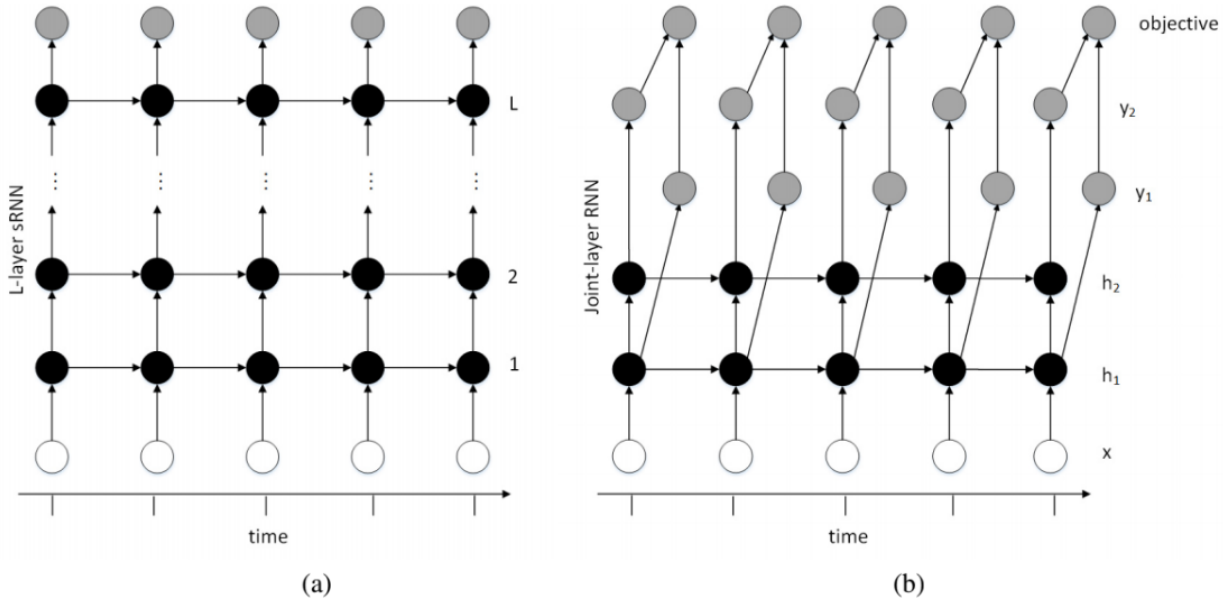

Рис. 2(a) - схема Stacked Recurrent Neural Network

Рис. 2(b) - схема Joint-layer Recurrent Neural Network

Выходные значения $\mathbf{y}_t^1$ и $\mathbf{y}_t^2$, а также значения $\hat{\mathbf{y}}_t^1$ и $\hat{\mathbf{y}}_t^2$ сообщают нейронной сети о том, является ли рассматриваемое слово ключевым или являтся ли оно частью ключевой фразы соответственно. $\hat{\mathbf{y}}_t^1$ может иметь одно из 2-х значений: *True* или *False*, в зависимости от того, является ли данное слово ключевым или нет. $\hat{\mathbf{y}}_t^2$ может иметь одно из 5-ти возможных значений: *Single*, *Begin*, *Middle*, *End* или *Not*, сообщающих нейронной сети о том, является ли рассматриваемое слово одиночным ключевым, началом ключевой фразы, её серединой (т.е стоящим в ней не первым и не последним), концом фразы или вообще не является ключевым для рассматриваемого отрывка.

Т.к решаемой задачей является извлечение ключевых фраз из последовательности слов, авторы в [12] адаптировали архитектуру нейронной сети для одновременного нахождения ключевых слов, и извлечения клю-

чевых фраз. Значения скрытых слоёв определяются как:

$$\mathbf{h}_t^1 = f_h(\mathbf{x}_t, \mathbf{h}_{t-1}^1)$$
$$\mathbf{h}_t^2 = f_h(\mathbf{h}_t^1, \mathbf{h}_{t-1}^2)$$

Значения выходного слоя определяются как:

$$\hat{\mathbf{y}}_t^1 = f_o(\mathbf{h}_t^1)$$
$$\hat{\mathbf{y}}_t^2 = f_o(\mathbf{h}_t^2).$$

6.3. Обучение нейросети

Обозначим параметры обучения как θ.

$$\theta = \{\mathbf{X}, \mathbf{W}^1, \mathbf{W}^2, \mathbf{U}^1, \mathbf{U}^2, \mathbf{V}^1, \mathbf{V}^2\},$$

где **X** это векторные представления слов, а остальные переменные описаны в п. 5.2. В одних и тех же выражениях размечаются как ключевые слова, так и ключевые фразы (фразы состоят из ключевых слов). На выходе первого скрытого слоя мы получаем из модели информацию о ключевых словах, на втором - о ключевых фразах. Затем результаты, полученные на выходе из этих слоёв комбинируются в финальный результат, вычисляемый как:

$$J(\theta) = \alpha J_1(\theta) + (1 - \alpha) J_2(\theta), \tag{1}$$

где α- линейный фактор веса. Для данных N тренировочных последовательностей $D = \{(\mathbf{x}_t, \mathbf{y}_t^1, \mathbf{y}_t^2)_{t=1}^{T_n}\}_{n=1}^N$ значения $J_1(\theta)$ и $J_2(\theta)$ определяются как:

$$J_1(\theta) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} d(\hat{\mathbf{y}}_t^1, \mathbf{y}_t^1) \tag{2}$$

$$J_2(\theta) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} d(\hat{\mathbf{y}}_t^2, \mathbf{y}_t^2), \tag{3}$$

где *d*(**a**, **b**) это некоторая мера близости(например евклидово расстояние или перекрёстная энтропия).

Формулы (2) и (3) демонстрируют, что мы вычисляем ключевые слова и извлекаем ключевые фразы на разных уровнях одновременно, что позволяет достигнуть более высоких результатов, по сравнению с другими методами решения задачи извлечения ключевых фраз[12].

6.4. Создание модели и обучение

На данном этапе создание модели и её обучение проводилось в полном соответствии с [12], чтобы проверить насколько точно данный алгоритм, изначально предназначенный для английского языка, может быть применён к русскоязычным твитам без каких-либо изменений.

В качестве переменных обучения были взяты следующие значения:

Window size:	3
Learning rate:	0.1
Нейронов первого скрытого слоя:	300
Нейронов второго скрытого слоя:	300
Batch size:	16

α: 0.5

Здесь имеется ввиду α из формулы (1)

Все переменные были взяты из [12], т.к модель, представленная авторами статьи показала наилучшие результаты именно при таких параметрах обучения. В качестве функции потерь(*loss function*) использовалась softmax cross-enthropy(*перекрёстная энтропия*). Функция Softmax вычисляется как:

$$p_i = \frac{e^{a_i}}{\sum\limits_{k=1}^{N} e_k^a}, \tag{4}$$

где вектор вещественных чисел размерности N преобразуется в вектор той же размерности, но каждая компонента p_i вектора p представлена числом в интервале [0,1] и сумма координат равна 1. Перекрёстная энтропия между двумя распределениями вероятностей измеряет среднее число бит, необходимых для опознания события из некоторого набора если используемая схема кодирования базируется на заданном распределении вероятностей*y* , вместо «истинного» распределения *p*, где *p* является вектором получающимся в результате преобразования с помощью функции (4). Функция потерь в таком случае выглядит так:

$$E(y, p) = -\sum_j y_j \log p_j,$$

где *p* - является предположением нейронной сети, а *y* - вектором меток, преобразованным с помощью (4).

Результаты обучения нейросети, полностью идентичной [12] представлены на рис.5.

6.5. Анализ результатов

Рассмотрим результаты после 150 эпох обучения.

Таблица 3. Результаты обучения после 150 эпох

Precision	Recall	F-measure
78.6%	71.1%	74%

Окончательный результат после 150 эпох обучения составил 68% accuracy, 78.6% precision, 71.7% recall и 74% f-measure(значения вычислены согласно формулам из п.4.5). По причине того, что в [12] значения метрик для оценки качества вычислялись по другим показателям(китайские исследователи вычисляли Precision как долю верно угаданных фраз среди всех извлечённых фраз, а Recall как долю верно угаданных фраз среди размеченных экспертом), полное сравнение результатов будет не совсем корректным, однако если взглянуть на результаты, полученные в [12] (Precision - 80,74%, Recall - 81,19%, F-Measure - 80,97%), можно заметить, что значения аналогичных метрик довольно близки к результатам, полученным в настоящей работе, откуда можно сделать вывод о том, что результаты работы алгоритма для русскоязычных твитов стоит признать удовлетворительными, т.к в некотором смысле значения метрик Precision и Recall и в данной работе, и в [12] определяют точность и полноту, несмотря на то, что рассчитаны на основе разных показателей. Также необходимо пояснить, что мы извлекаем слова из фраз с указанием их позиции во фразе, так что, если задача решена качественно, то с высокой вероятностью фраза будет восстановлена корректно. Менее высокие результаты в сравнении с

англоязычными текстами можно объяснить множеством сложностей, с которыми связаны как в целом задачи из области NLP*(Natural Language Processing)*, так и конкретно задача извлечения ключевых слов и фраз из коротких текстов.

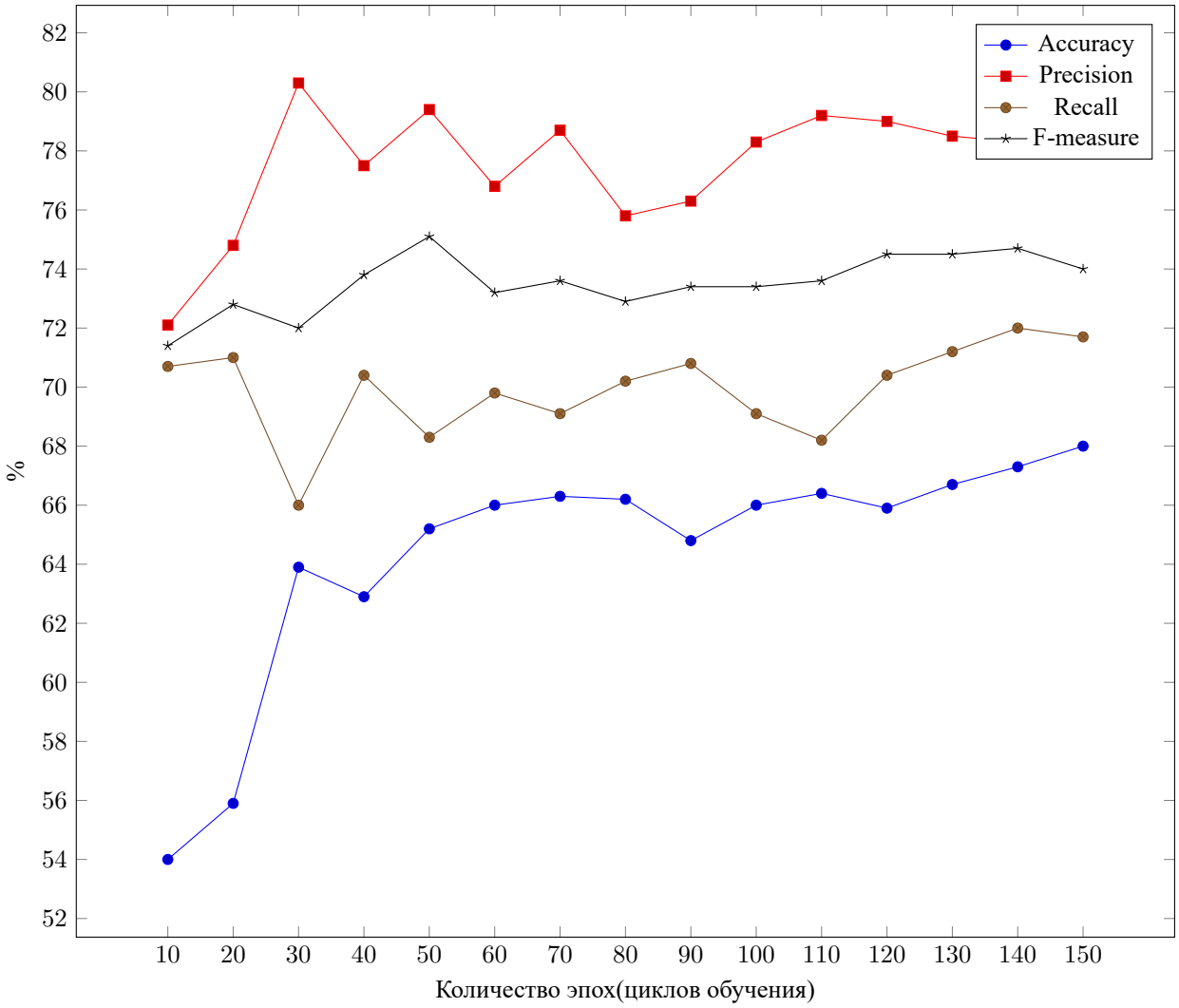

Рис.5. Результаты обучения Joint-layer neural network

Одной из главных проблем является строгий порядок слов в английских предложениях, в то время как в русском языке автор имеет большую свободу в конструировании предложений. В связи с этим предложения имеют менее структурированный вид и сложнее поддаются формализации и обработке и использованием математических моделей[6].

Таким образом данная методика построения нейронных сетей и составления обучающей выборки действительно может быть применена и для анализа коротких текстов в русскоязычных социальных сетях. Сравнительный анализ данного и других методов решения поставленной задачи будут представлены в следующих пунктах данной работы.

7. Поиск путей усовершенствования алгоритма извлечения ключевых фраз с помощью joint-layer neural networks

7.1. Расширение алгоритма для поиска нескольких ключевых фраз в одном твите

Принимая во внимание тот факт, что алгоритм показал вполне приемлемые результаты при обучении на выборке, из которой были удалены все твиты, содержащие больше одного хэштега, можно предположить, что и при увеличении допустимого количества хэштегов в твитах, составляющих обучающую выборку, алгоритм будет демонстрировать достаточную точность. Это предположение основано на том, что на вход нейронной сети подаётся не весь твит сразу, а только его часть, входящая в так называемый *window-size*, таким образом для нейронной сети не имеет значения количество хэштегов в твите в общем, значение имеет только то, что нейронная сеть анализирует внутри этого *window-size*.

Исходя из этого, логичным шагом представляется увеличение количества допустимых хэшегов для твита до двух. Для того, чтобы выяснить, насколько уверенно можно говорить о том, что и при выделении двух хэштегов в качестве ключевых слов или фраз для данного твита, выборка остаётся приемлемой для обучения на ней нейронной сети, было проведено исследование, аналогичное описанному в п. 5.3. Те же три волонтёра получили для анализа 1000 случайно выбранных твитов, содержащих ровно 2 хэштега. Им, также как и в п. 5.3, предлагалось оценить по трёхбалльной шкале то, насколько уверенно можно называть выделенные слова и фразы ключевыми для данного твита, где оценка ”2”означало, что слова и фразы определённо являются ключевыми для твита, оценка ”1 скорее являются ключевыми, а оценка ”0”означала, что представленные слова и фразы абсолютно нельзя назвать ключевыми для рассматриваемого твита. В результате опроса волонтёров были получены следующие результаты:

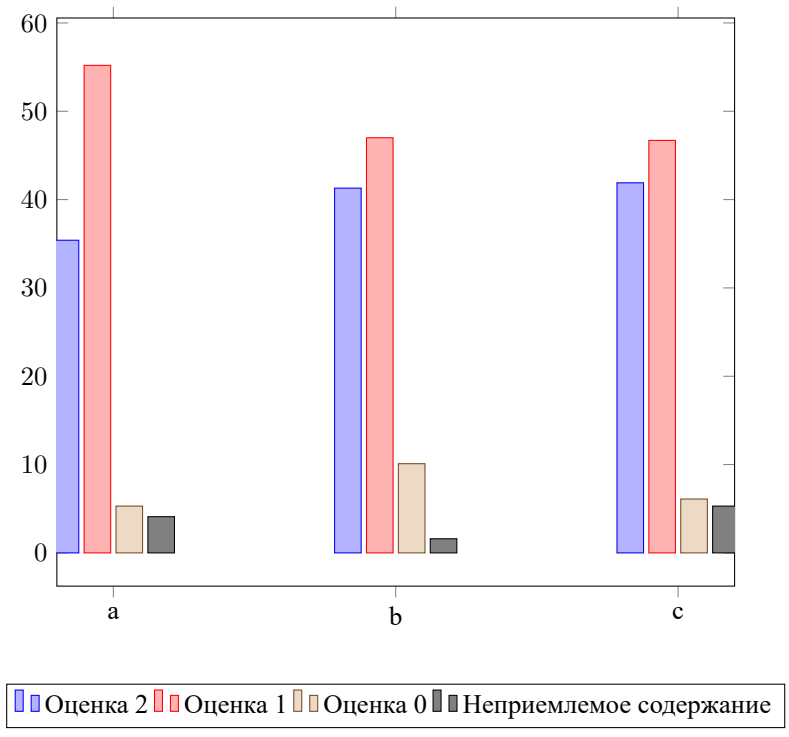

Рис.6 Диаграмма распределения оценок волонтёров для твитов, содержащих два хэштега.

Таким образом в среднем в 39,5% случаев извлечённые ключевые фразы полностью соответствовали содержанию твита. Также в 49,6% случаев было выявлено почти полное соответствие ключевых фраз или

слов и содержания твита. Большее, чем в п.5.3 количество оценок ”1”можно объяснить большим количеством слов, помеченных как ключевые, т.е с большей вероятностью одно из слов или одна из фраз, помеченных как ключевые, на самом деле являются таковыми.

Для поддержки твитов с двумя хэштегами был незначительно преобразован алгоритм подготовки данных. Например, если один из хэштегов является фразой, а второй - словом, то последовательность, подающаяся на вход нейронной сети, будет выглядеть так:

1	2	3	4	5	6

Рис. 7(а) - Твит в виде массива включений слов

1	1	1	0	1	0

Рис. 7(б) - Разметка для ключевых слов

1	2	3	0	4	0

Рис. 7(в) - Разметка для ключевых фраз

После обработки датасета и выделения из него твитов, подходящих для обучения, был получен датасет со следующими характеристиками:

Таблица 4. Статистические характеристики датасета

N	W	T	$\bar{N}$	$\tilde{N}$
17 639 674	56605	52814	9.959	1.548

Здесь N общее количество твитов в исходном датасете, W - словарь датасета(количество уникальных слов), T - количество твитов после обработки и отделения неподходящих для обучения нейросети, $\bar{N}$ - среднее количество слов в твите, $\tilde{N}$ - среднее количество хэштегов в твите

Таким образом 54% твитов от обучающей выборки составили твиты, содержащие в себе 2 хэштега. Далее была обучена модель с помощью Joint-layer neural network. Результаты обучения представлены на рис. 7. Обучение модели длилось не 150 эпох, как в предыдущих случаях, а 90, по причине того, что обучающая выборка возросла почти в 2 раза и, следовательно количество записей, проходящих через нейронную сеть за одну эпоху, также увеличилось почти вдвое. На графике видно, что по мере обучения немного падает показатель Precision и также слегка увеличивается значение показателя полноты (Recall), т.е с каждой эпохой увеличивалось количество слов, которые были помечены ключевыми по ошибке, однако также уменьшалось и число слов, которые ошибочно были помечены как неключевые. По-сравнению с результатами, полученными для нейросети, обученной на выборке твитов, содержащих только один хэштег в п.6.4, качество работы нейросети незначительно снизилось, однако расширился её диапазон применения в результате увеличения возможного количества ключевых фраз, присутствующих в твите(результаты для сравнения представлены в Таблице 5).

Таблица 5. Сравнение результатов обучения для твитов с одним и двумя хэштегами.

	Precision	Recall	F-measure
Один хэштег	78,6%	71.1%	74%
Два хэштега	78.8%	65.8%	71%

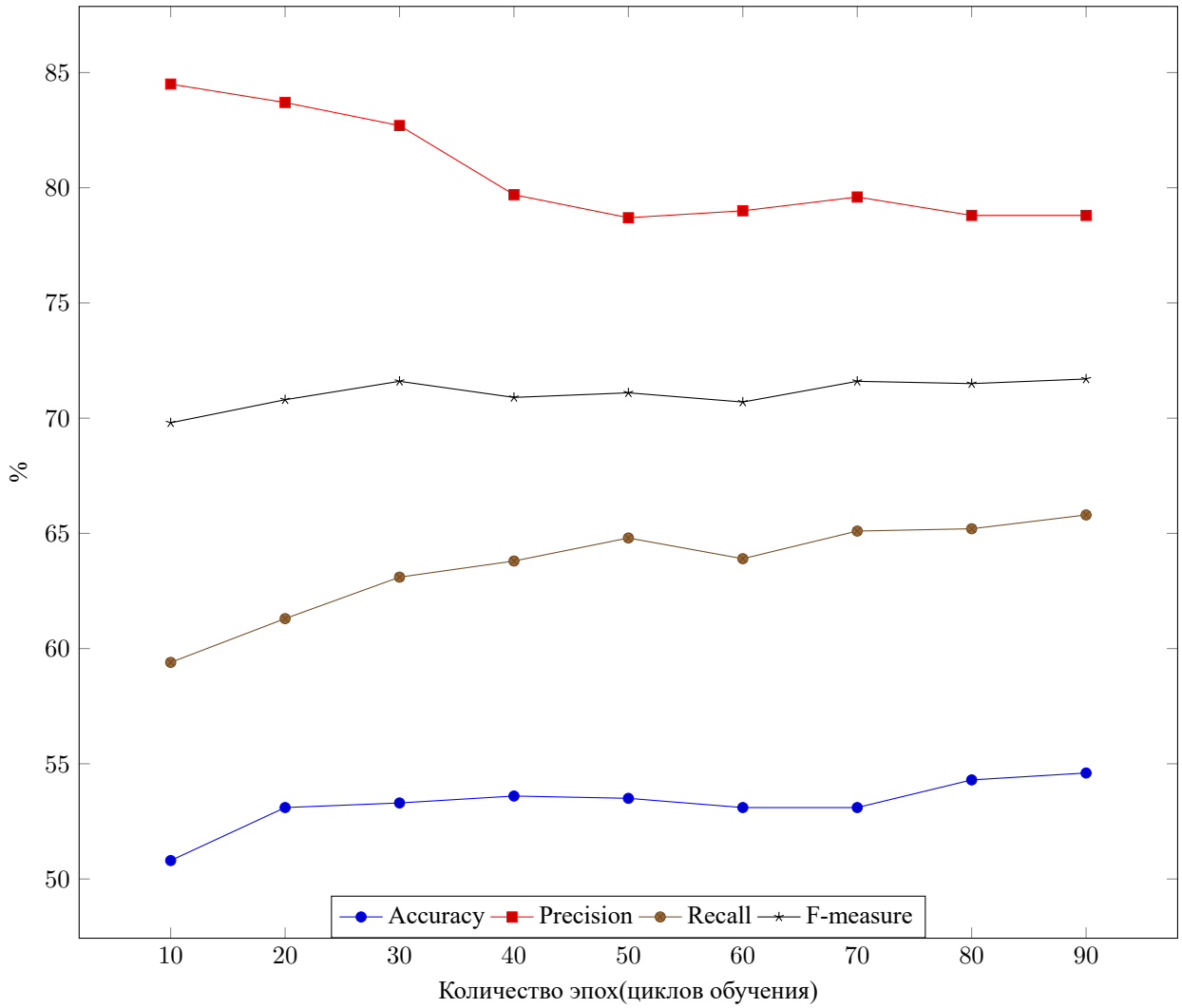

Рис.8. Результаты обучения после удаления из выборки твитов с наиболее популярными хэштегами.

Таким образом можно сказать, что данное расширение для решения задачи извлечения ключевых фраз из твитов с использованием нейронных сетей можно признать успешным, т.к при увеличении возможного числа ключевых фраз до 2 метрика F(на значение которой нужно ориентироваться в первую очередь, т.к она является гармоническим средним для двух других метрик) показала падение всего лишь на 3%, а значение Precision и вовсе возросло(правда всего лишь на 0,2%). Т.е данный эксперимент может с большой долеё вероятности быть признан удачным.

7.2. Выделение наиболее популярных хэштегов

Не секрет, что при работе с текстами, опубликованными в социальной сети “Твиттер” необходимо аккуратно анализировать такой важный атрибут данной социальной сети, как хэштег, и учитывать особенности построения исследования на анализе хэштегов. Как известно, часто пользователи сети “Твиттер” используют некоторые общепринятые хэштеги, особенно когда публикуют тексты, связанные с какими-либо сетевыми флешмобами, акциями поддержки или протеста итд(напр. #metoo). Очевидно, что в случае, если мы принимаем хэштеги за ключевые фразы или слова в нашем исследовании, данная ситуация приводит к тому, что обучающая выборка наполняется информационным шумом, т.к такие хэштеги семантически не являются частью твита, а часто вставляются в него только с целью привлечения внимания к данной записи или просто как дань моде.

Очевидным способом борьбы с наполнением обучающего датасета подобными твитами и хэштегами является частотный анализ хэштегов в препарируемом датасете с целью выявления наиболее часто встречающихся хэштегов и последующего их игнорирования для повышения ориентированности модели на обработку векторных представлений слов, а не простое заучивание популярных хэштегов.

В среднем частота каждого хэштега в обучающей выборке равняется 5.549546118199708. Однако в датасете присутствуют хэштеги с частотой более 500, причём такие хэштеги присутствуют в количестве 16 штук. Необходимо проверить увеличится ли точность работы алгоритма, если исключить из рассмотрения твиты, содержашие в сеье указанные хэштеги и, если увеличится, то насколько. Значение 500 было найдено эмпирическим подбором. После подсчёта выяснилось, что из 22853 твитов, присутствующих в обучающей выборке, 6193 твита(27% от общего числа) содержат обозначенные выше стоп-слова. После их удаления из выборки и повторного обучения модели были получены следующие результаты:

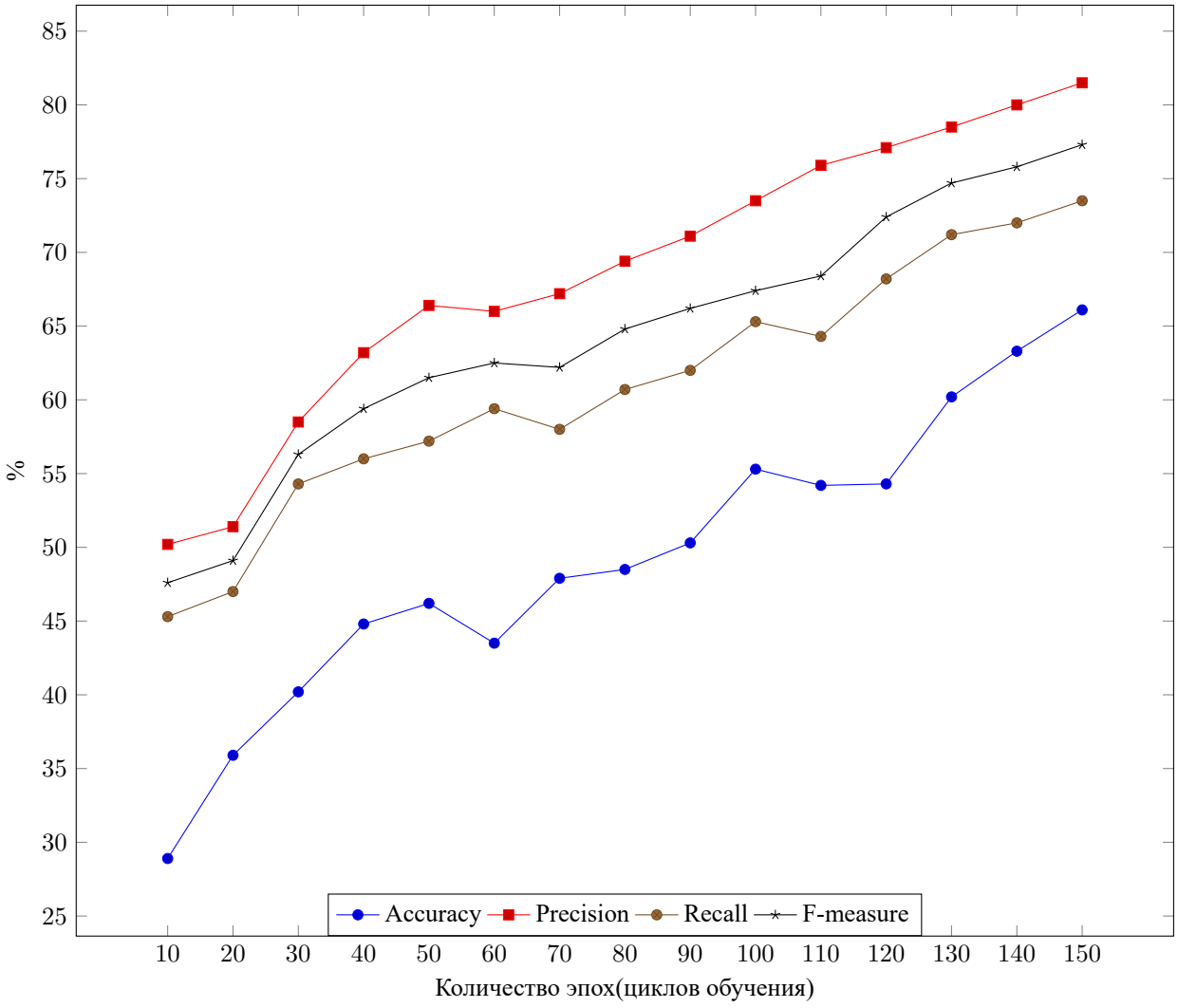

Рис.9. Результаты обучения после удаления из выборки твитов с наиболее популярными хэштегами.

Таблица 6. Сравнение результатов обучения для выборки без наиболее популярных хэштегов и с ними.

	Precision	Recall	F-measure
Все хэштеги	78,6%	71.1%	74%
С учётом популярности хэштегов	81,5%	73.5%	77.3%

Таким образом удаление из обучающей выборки наиболее часто встречающихся хэштегов дало вполне значительный прирост для всех 3 рассматриваемых метрик. Наибольший рост показала метрика F(+3.3%), что является отличным результатом, по причине того, что именно эта метрика для нас является основной, т.к она учитывает и полноту и точность работы нейросети. Таким образом данный эксперимент, как и предыдущий, с большой долей вероятности может быть признан вполне удачным.

7.3. Анализ результатов

Проанализировав результаты исследований можно заметить, что подход, представленный в [12] не только может быть без особенных потерь качества переориентирован на анализ русскоязычных твитов, но и с немалой долей успеха преобразован без потерь, а в некоторых случаях и с увеличением качества(п. 6.2) или увеличением его области применения(п.6.1). Таким образом можно говорить о том, что выбранные пути усовершенствования алгоритма извлечения ключевых фраз с использованием joint-layer neural networks применимы на практике.

Таблица 7. Сравнение результатов обучения

	Precision	Recall	F-measure
Результаты из пункта 6.6	78,6%	71.1%	74%
С учётом популярности хэштегов	81,5%	73.5%	77.3%
Обучение на выборке с расширением до 2-х ключевых фраз	78.8%	65.8%	71%

8. Сравнение с другими методами

8.1. TF-IDF

Наиболее простым методом выделения в тексте самых важных слов является выделение на основе меры *TF-IDF(term frequency - inverse document frequency)*. *TF* это отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, оценивается важность слова в пределах отдельного документа. Это значение вычисляется как:

$$tf(t,d) = \frac{n_t}{\sum\limits_k n_k}, \tag{5}$$

где n_t - число вхождений слова t в документ, а в знаменателе - общее число слов в документе. *IDF* - инверсия частоты, с которой некоторое слово встречается в документах коллекции. Учёт IDF уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF. Это значение вычисляется по формуле:

$$idf(t,D) = \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|}, \tag{6}$$

где |D| - общее число документов в коллекции, а $|\{d_i \in D | t \in d_i\}|$ - число документов из коллекции D, в которых встречается t. Большой вес в *TF-IDF* получат слова с высокой частотой в пределах конкретного документа и с низкой частотой употреблений в других документах[9]. Опираясь на предположение о том, что слова с наибольшим значением меры *TF-IDF* будут ключевыми, была рассчитана мера для всех слов, присутствующих в датасете. По причине того, что TF-IDF является всего лишь количественной мерой относительной семантической важности слова в некотором корпусе текстов, с помощью TF-IDF невозможно создать полноценный алгоритм, извлекающий ключевые слова и фразы из каждого текста, находящегося в корпусе. С помощью этой меры мы можем лишь рассчитать вероятность того, что то или иное слово будет входить или не будет входить в множество ключевых слов для каждого конкретного текста. Поэтому для сравнения результатов извлечения ключевых фраз с помощью TF-IDF с результатами извлечения с помощью *Алгоритма 1*, был разработан следующий алгоритм:

Algorithm 2 Проверка результатов TF-IDF

1: **for all** $tweet$ in $tweets$ **do**
2: **if** Ключевые слова $tweet$ = слова с наибольшим значением TF-IDF из $tweet$ **then**
3: Результат верный
4: **else**
5: Результат неверный
6: **end if**
7: **end for**

В данном алгоритме значение меры TF-IDF вычисляется по формулам (5) и (6). Предварительно все тексты из корпуса были лемматизированы, т.е все слова были преобразованы к своей начальной форме, чтобы уменьшить количество коллизий между ними.

Результаты извлечения с помощью меры TF-IDF оказались намного ниже, чем у других методов, рассмотреных в данной работе. Всего верных результатов(в значении, обозначенном в *Алгоритме 2*) было получено **12,2%**. Т.к данное значение соответствует значению метрики Accuracy(обе метрики представляют собой

долю твитов с полностью верно угаданными ключевыми словами среди общего числа твитов), то логично сравнить его со значением данной метрики, причём для модели, обученной разделять слова только на 2 категории: принадлежащие ключевой фразе и не принадлежащие ей. Значение основного алгоритма с использованием joint-layer neural networks составило **76,3%** для метрики Accuracy, что значительно результатов, полученных при и спользовании tf-idf. Такую низкую точность можно объяснить во-первых тем, что далеко не всегда наиболее важные(с точки зрения меры TF-IDF) слова из твита являются ключевыми в смысле данной работы. Также метод извлечения ключевых фраз из твитов и вообще текстов, размещённых в социальных сетях, с помощью меры TF-IDF часто приводит к ошибкам, таким как попадание в список наиболее важных слов тех, которые, допустим, образованы самим автором твита и вообще не существуют с точки зрения русского языка. Такие ошибки приводят к большому количеству “шума” и тоже не способствуют повышению качества работы алгоритма.

Ещё одним недостатком является уже заявленная выше невозможность выделения полноценных ключевых фраз с помощью данного метода. При проведении настоящего исследования было произведено некоторое упрощения с целью сведения данных, полученных из разных алгоритмов к некоторому единому виду, для того, чтобы получить возможность в дальнейшем сравнить их точнность работы. В частности при использовании метода извлечения с помощью меры TF-IDF мы заранее знали количество ключевых слов, присутствующих в твите и и извлекали с помощью метода TF-IDF уже известное количество ключевых слов или фраз, что, конечно, является некоторым допущением и неприменимо в случае использования данного алгоритма с реальными данными.

В случае, если исследования проводятся на корпусе твитов, в которых гарантированно присутствует ключевое слово, причём ровно одно, данный алгоритм показывает чуть более высокие(**15,1%**), но совсем неудовлетворительные результаты. В данном случае сравнивались слово, которое в твите было помечено, как ключевое и слово, с наибольшим значением меры TF-IDF среди всех слов твита.

Таким образом можно сделать вывод, что один из наиболее популярных методов, которым является извлечение наиболее важных слов с помощью меры TF-IDF, показывает несравнимо более низкие результаты, чем другие методы, представленные в данной работе.

8.2. Рекуррентные нейронные сети

Рекуррентные нейронные сети *Recurrent neural networks* далее *RNN* - вид нейронных сетей, где связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки[7]. Рекуррентные нейронные сети являются базовыми для данной работы, т.к данное исследование построено на преобразовании их структуры с целью получения результатов более высокой точности. Однако необходимо рассмотреть и метод решения задачи извлечения ключевых слов с помощью в том числе и простых RNN, как минимум для того, чтобы получить результат для сравнительного анализа точности обыкновенных RNN и Joint-layer neural networks.

Ранее на рис.2(а) была продемонстрирована схема построения обыкновенной RNN, которая и была применена в данном пункте. Далее было проведено исследование, в ходе которого основная задача данной работы была решена с помощью обычкновенной RNN, в результате были получены следующие результаты:

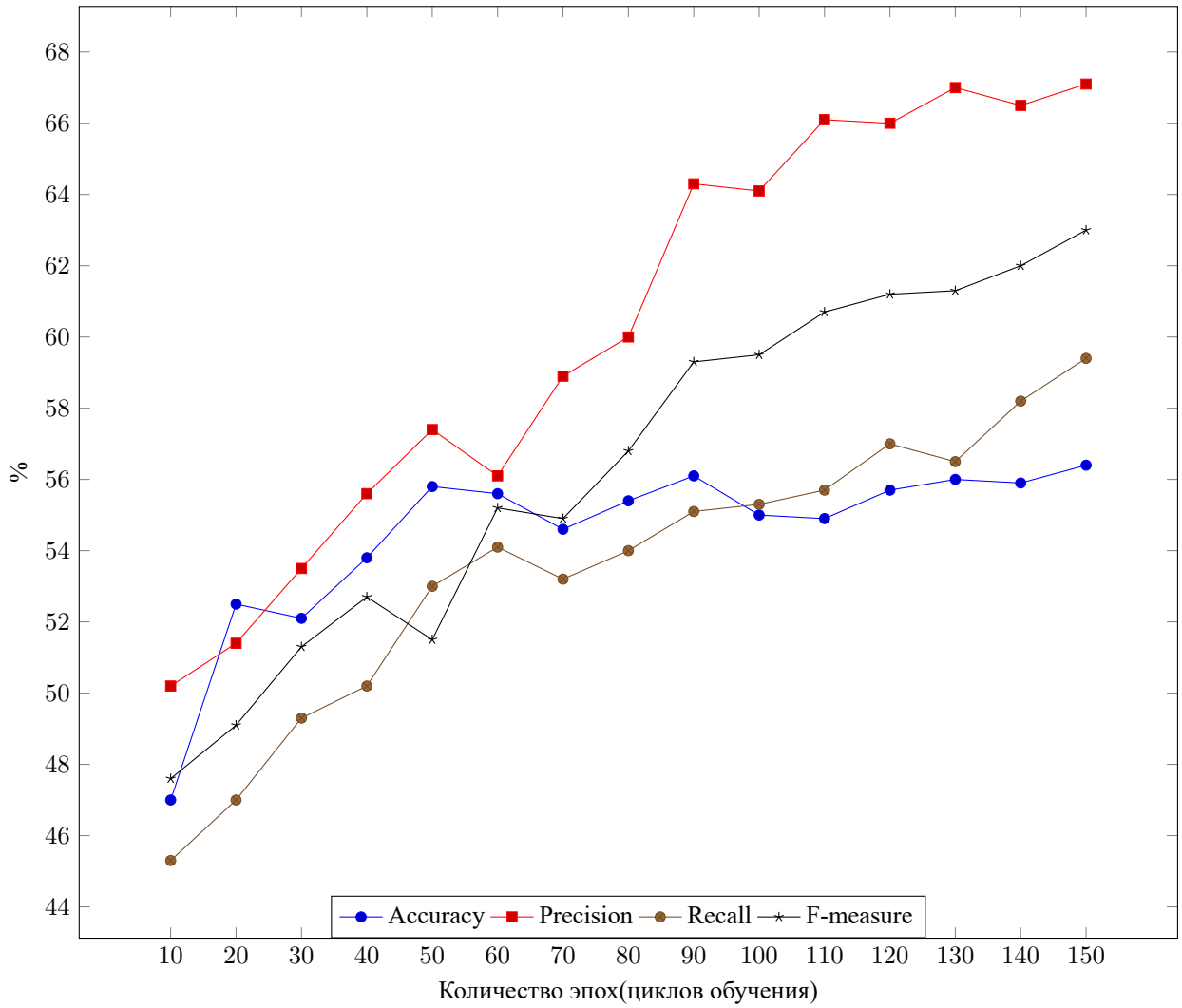

Рис.10. Результаты обучения с использованием обыкновенных RNN.

8.3. Анализ результатов

Рассмотрев и проанализировав результаты сравнения точности работы алгоритмов, описанных в данном пункте с аналогичными характеристиками для основного алгоритма, описанного в данной работе, можно сделать вывод о том, что и TF-IDF, и обыкновенные RNN уступают по точности и другим характеристикам Joint-Layer RNN. Причём, если различие в точности по-сравнению с RNN не так велико, что объясняется схожестью данных алгоритмов(Joint-Layer RNN является более совершенной модификацией RNN), то различие в точности по сравнению с алгоритмом, основанным на мере TF-IDF довольно внушительно(12,2% точности у TF-IDF против 76,3% у Joint-Layer RNN). Этот и другие результаты, полученные в ходе проведения данного исследования могут указывать на то, что методики, основанные на алгоритмов глубокого обучения(*deep learning*) являются более предпочтительными для данной задачи, чем любые другие методы решения[11].

9. Выводы

В данной работе была предпринята попытка исследовать различные методы решения задачи извлечения ключевых слов и фраз из постов в социальной сети “Твиттер”. Основанием для такой попытки было большое количество довольно подробных исследований данной задачи и смежных с ней для англоязычного сегмента указанной социальной сети и почти полное отсутствие исследований на данную тему, где объектом изучения выступали бы твиты из русскоязычного сегмента “Твиттера”.

За основу было взято исследование, подробно описанное авторами в [12], где, помимо прочего, доказывается, что метод, разработанный в данной работе(а именно - извлечение с помощью Joint-Layer neural network) превосходит по точности работы другие основные подходы, обычно применяющиеся в задачах Natural Language Processing-а. Основной задачей была трансформация метода так, чтобы иметь возможность использовать его для извлечения ключевых фраз из твитов в том числе и на русском языке. Данную попытку можно с большой долей вероятности признать успешной, т.к модель, построенная и обученная на русскоязычных твитах, показала результаты, соизмеримые с теми, которые были продемонстрированы в [12](с учётом допущений, описанных в п.6.6). Таким образом можно сделать вывод о том, что данный метод может быть использован в том числе и для работы с русскоязычными твитами.

Далее было предпринято несколько попыток усовершенствования вышеописанного подхода. Расширение области применения путём увеличения максимально возможного количества ключевых фраз в одном твите дало также положительные результаты, т.к падение качества работы алгоритма составило довольно незначительные 3%(имеется ввиду падение F-measure) при том, что количество твитов, которые могут быть обработаны с помощью усовершенствованной модели увеличилось почти вдвое(в датасете Ю. Рубцовой, использованном для данного исследования).

Обучение модели с учётом наиболее популярных хэштегов также дало положительный результат. После удаления из обучающей выборки твитов, содержащих наиболее популярные хэштеги, точность работы модели резко упала на ранних этапах обучения(это связано с тем, что модель была таким образом лишена возможности запомнить наиболее часто встречающиеся ключевые фразы и в тестовой выборке пометить их также как ключевые), однако на финише обучения данная модель превзошла результаты, показанные во всех остальных пунктах данного исследования. При этом показатель точности(Precision) такой модели вполне сравним с аналогичным показателем для модели, представленной в [12], что, несмотря на некоторую разницу в методах подсчёта для оценок качества, описанную в п.6.6, является высоким результатом, учитывая более сложную и менее упорядоченную структуру предложений(а соответственно и твитов) в русском языке в сравнении с английским языком. Таким образом оба улучшения также могут быть признаны удачными и пригодными к использованию при решении данной задачи и смежных с ней.

Затем была предпринята попытка сравнить результаты работы метода с использованием Joint-layer neural network и некоторых других популярных методов решения данной задачи. При сравнении результатов работы основного метода и метода, использующего меру TF-IDF, было зафиксировано, что точность работы метода, использующего меру TF-IDF значительно ниже, чем при использованиии нейронных сетей. В то же время при сравнении результатов, получающихся при использовании основного подхода и подохода, основанного на обыкновенных Recurrent Neural Networks, было зафиксировано, что хоть обозначенный метод и даёт точность большую, чем при использовании меры TF-IDF, всё же качество его работы по всем показателям уступает основному методу, рассматриваемому в данной работе. Таким образом можно заключить, что результаты

сравнительного анализа метода, использующего Joint-layer neural networks и некоторых других популярных методов, применяющихся в данной области, получились схожими с результатами, полученными в ходе исследований китайскими учёными при работе с англоязычными твитами и эти результаты также говорят о том, что описанная методика извлечения ключевых слов и фраз из коротких текстов пригодна для применения в решении задачи и, помимо этого, показывает и более высокие результаты при сравнении с другими популярными подходами в данной предметной области.

Опираясь все представленные выше факты можно с полной уверенностью заявить, что поставленная в п.2 данной работы задача была полностью и успешно выполнена, а полученные результаты могут быть использованы как для дальнейших исследований в этом направлении, так и для решения различных бизнес-задач.

10. Заключение

Как было заявлено во введении, задача извлечения ключевых слов и фраз из коротких текстов(наиболее распространённым примером которых являются записи, размещаемые в социальной сети ”Твиттер”) становится с течением времени всё более актуальной. Данная исследование было проведено с целью изучения возможности применения наиболее передовых подходов, используемых для извлечения ключевых фраз из текстов на английском языке, для решения аналогичной задачи для русскоязычного контента.

По итогам выполнения работы были решены следующие задачи:

- Проанализирована литература в данной предметной области
- Исходя из условий выбраны наиболее подходящие инструменты для работы
- Разработана русскоязычная обучающая коллекция
- Реализованы алгоритмы извлечения ключевых фраз
- Алгоритмы протестированы и проведена оценка качества их работы
- Проанализированы полученные результаты
- По итогам проекта подготовлены соответствующие выводы

Результаты данной работы говорят о том, что описанный в [12] подход демонстрирует высокие результаты не только для английского, но и для русского языка. В связи с чем можно признать, что цели, поставленные в п.2 данной работы были выполнены в полном объёме, получены результаты, позволяющие опираться на них в дальнейших исследованиях данного вопроса.

11. Список литературы

Список литературы

[1] Брадис Н.В. Применение статистических методов выявления устойчивых словосочетаний в текстах на русском языке для извлечения ключевых фраз/ С.В Брадис, Д.А. Сытник // Глобальный научный потенциал. Информационные технологии в экономике, 2016, №11(68).

[2] Ванюшкин А.С. Методы и алгоритмы извлечения ключевых слов / А.С. Ванюшкин, Л.А. Гращенко // Новые информационные технологии в автоматизированных системах, 2016

[3] Попова С.В. Извлечение ключевых словосочетаний/ С.В. Попова, И.А. Ходырев // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики, 2012, № 1 (77).

[4] Соколова. Е.В. Автоматическое извлечение ключевых слов и словосочетаний из русскоязычных текстов с помощью алгоритма KEA./ Е.В. Соколова, О.А. Митрофанова// Компьютерная лингвистика и вычислительные онтологии. Выпуск 1 (Труды XX Международной объединенной научной конференции «Интернет и современное общество», IMS-2017, Санкт-Петербург, 21 – 23 июня 2017 г. Сборник научных статей).

[5] Ю. В. Рубцова, Построение корпуса текстов для настройки тонового классификатора // Программные продукты и системы, 2015, №1(109), С.72-78.

[6] Шереметьева, С.О. Методы и модели автоматического извлечения ключевых слов / С.О. Шереметьева, П.Г. Осминин // Вестник ЮУрГУ. Серия «Лингвистика». – 2015. – Т. 12, № 1. – С. 76–81.

[7] Mandic, D., Chambers, J. Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability.

[8] Tomas Mikolov, Distributed Representations of Words and Phrases and their Compositionality / T. Mikolov, I.Sutskever, K.Chen, G.Corrado, J.Dean // In Proceedings of NIPS.

[9] Salton, G., Buckley, C. Term-weighting approaches in automatic text retrieval. Information Processing & Management, 1988, 24(5): 513—523

[10] Ilya Segalovich, A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine // Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications. MLMTA’03, June 23 - 26, 2003, Las Vegas, Nevada, USA

[11] Nils Schaetti, UniNE at CLEF 2017: TF-IDF and Deep-Learning for Author Profiling, Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017.

[12] Qi Zhang, Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter / Qi Zhang, Yang Wang, Yeyun Gong, Xuanjing Huang // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 836–845, Austin, Texas, November 1-5, 2016.

[13] Yabin Zheng, Automatic Keyphrase Extraction via Topic Decomposition / Zhiyuan Liu, Wenyi Huang, Yabin Zheng, Maosong Sun // Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 366–376, MIT, Massachusetts, USA, 9-11 October 2010.

[14] Python 3.7.3 documentation - https://docs.python.org/3/ - 3.7.3 Documentation

12. Приложение

Приложение 1 Интерфейс приложения, которое было разработано для упрощения процесса оценки датасета волонтёрами.