Санкт-Петербургский государственный университет

Безрукова Дарья Михайловна

Выпускная квалификационная работа

Анализ тональности текстов новостных источников по отношению к заданному объекту

Уровень образования:

Направление *03.04.01 «Прикладные математика и физика»*

Основная образовательная программа *ВМ.5521.2017 «Математические и информационные технологии»*

Научный руководитель:
доцент кафедры ТСУЭФА,
к.ф.-м.н.,
Головкина А. Г.

Рецензент:
руководитель DevOps
подразделения
ООО "Ф-Лайн Софтвер",
к.ф.-м.н.,
Райк А. В.

Санкт-Петербург

2019

Оглавление

Введение

В настоящее время, чтобы стать лидером в своей отрасли, компаниям необходимо не просто производить качественные товары, оказывать большой спектр услуг, но и постоянно собирать обратную связь с потребителей, знать, что они думают о товаре, какие отзывы оставляют на специализированных ресурсах, оценивать общую удовлетворенность людей от продукта. В свою очередь любой человек, совершающий покупку или делающий выбор в пользу коммерческих предложений, сталкивается с необходимостью получения независимой оценки этих товаров. Он хочет узнать опыт других людей, понять какое впечатление на них произвел тот или иной производитель услуг. Важность этой информации повышается в разы, когда речь заходит о выборе банковских продуктов, в частности, кредита, ипотеки или ведения счетов ИП. В условиях стремительного роста пользовательских текстов в Интернете автоматическое извлечение полезной информации из многочисленных документов вызывает интерес у исследователей во многих областях, в частности в области *обработки естественного языка* (*Natural Language Processing*).

Анализ мнений (*opinion mining*) или *анализ тональности текстов* (*Sentiment Analysis*)– это область компьютерной лингвистики, которая занимается автоматизированным выявлением и изучением эмоционально окрашенной лексики и эмоциональной оценки авторов по отношению к объектам, о которых идет речь в тексте. Данное научное направление зародилось в начале этого столетия и постепенно стало активно развиваться в связи большим количеством практических применений в различных областях, например, анализ ценообразования [1] ,мониторинг бренда [2], прогнозирование рынка [3] и др. Анализ тональности позволяет извлечь из текста мнение автора в отношении заданного объекта. Отношение может выражать суждение, мнение или оценку автора, его эмоциональное состояние.

В настоящее время рост популярности социальных сетей, интернет-магазинов и ресурсов с онлайн-обзорами различных продуктов и услуг

предоставляет большое количество материалов, которые могут быть использованы для принятия решения в пользу того или иного объекта.

Целью данной работы является сравнение методов анализа тональности текста и их применение по отношению к текстовым комментариям, оставленным на новостных и информационных порталах, посвященных сравнению банковских продуктов, а также разработка технологии автоматического выявления и оценки мнений. Решение этой проблемы позволит понять, когда клиенты банков удовлетворены или недовольны, в чем заключаются проблемы обслуживания, понять отношение клиентов к банку. Для банковского сектора извлечение такой информации является критичным, в силу высокой конкуренции в этой сфере. В связи с этим тема выпускной работы является **актуальной** и находит практическое применение.

Задача анализа тональности текста состоит из трех этапов: предварительной обработки текста, перевода текста в вещественное пространство признаков и использования методов машинного обучения для последующей классификации тональности. Предобработка текста – ключевой момент данного процесса, включающий в себя удаление стоп-слов, сегментацию и приведение слов к одной грамматической форме, маркировку частей речи и анализ. Современные алгоритмы машинного обучения, используемые при решении подобных задач, ориентированы на признаковое описание объектов [4]. В связи с этим после предобработки анализируемый текст переводится в вещественное пространство признаков. Для этого чаще всего используются методы, основанные на статистической информации о словах, например, «мешок слов» (bag of words) [5] или Word2Vec [6]. В этом случае каждому объекту соответствует вектор, длина которого равна количеству используемых слов во всех текстах выборки.

Заключительным шагом при анализе тональности текста является выбор подходящих для решения данной задачи алгоритмов машинного обучения. Как правило, анализ мнений на уровне документа может быть сформулирован как проблема классификации, которая определяет, выражается ли

положительное, отрицательное или нейтральное мнение. Классификаторы обучаются определять полярности рассматриваемых текстов. Наивный байесовский классификатор [7], энтропийный классификатор [8], метод опорных векторов (SVM) [9], градиентного бустинга [10] являются наиболее часто используемыми моделями.

В представленной работе исследуются существующие в настоящее время методы обработки естественного языка для анализа мнений клиентов банков. В разделе, посвященному обзору литературы, рассматриваются основные подходы и алгоритмы, описанные в литературе и применяемые на каждом из трех этапов решения поставленной задачи. В первой главе описываются общие методы обработки естественного языка, используемые для предварительной обработки текстов. Вторая глава посвящена сравнению двух наиболее популярных методик представления слова в виде вектора фиксированной длины: мешок слов и Word2Vec. В третьей главе исследуются алгоритмы классификации, использованные в данной работе. В четвертой главе представлено описание практической реализации рассмотренных алгоритмов и результаты определения тональности неразмеченных текстов мнений. В разделах выводы и заключение содержатся основные результаты выполненной работы, а также предлагаются возможные пути улучшения качества работы созданной системы.

Постановка задачи

Целью данной работы является разработка программного обеспечения для определения и анализа тональности текстов комментариев новостных и информационных источников по отношению к заданному объекту для последующей оптимизации процесса принятия решений. Для достижения поставленной цели необходимо решить следующие *задачи*:

1) разработать программное обеспечение для выгрузки релевантных поставленной задаче данных с web-ресурсов, на основе которых обучается машинный классификатор;

2) выполнить предварительную обработку выгруженных текстовых данных;

3) представить каждый блок текстовых данных в виде векторов признаков, по которым он будет анализироваться;

4) выбрать и реализовать подходящий алгоритм классификации и метод обучения классификатора;

5) провести валидацию модели на неразмеченных текстовых данных;

6) сравнить результаты системы при использовании методов векторизации Word2Vec и «мешок слов».

Обзор литературы

Анализ настроений или тональности сталкивается с тем же набором проблем, что и распознавание эмоций – прежде чем решить, какое настроение имеет данное предложение, необходимо установить в первую очередь, что такое «настроение». Можно ли определять настроение точно, это счастье, грусть, злость или скука? Или можно только как-то оценивать чувства, например по шкале от 1 до 10?

В дополнение к проблеме определения, в каждом предложенном человеком предложении есть несколько уровней значения. Люди выражают свое мнение сложными способами; риторические устройства, такие как сарказм, ирония и подразумеваемое значение, могут ввести в заблуждение анализ настроений. Единственный способ по-настоящему понять эти устройства - через контекст: знание того, как начинается абзац, может сильно повлиять на настроение последующих внутренних предложений.

Большая часть нынешнего мышления в анализе настроений происходит в категориальной структуре: настроения анализируются как принадлежащие к определенной группе, в определенной степени. Например, данное предложение может выражать «счастье» на 45%, «печаль» – на 23%, «возбуждение» – на 89% и «надежду» – на 55%. Эти цифры не суммируют до 100% – они являются индивидуальными признаками того, насколько «X» относится к предложению.

Создание входных данных для модели, которая распознает контекст, тон и другие признаки настроения, может помочь повысить точность и лучше понять то, что автор пытается сказать.

Существуют разные типы классификации найденных в тексте тональностей, самым базовым из них является стандартное деление на «позитивные» и «негативные» настроения. Однако часто такого деления недостаточно и используются более детализированные подходы. Основными видами классификации в настоящее время являются:

1)Классификация по бинарной шкале [11].

Это самый распространенный подход к оценке тональности. Есть всего два исхода, либо текст написан в положительном ключе, либо в отрицательном. Проблема в том, что часто нельзя однозначно определить настроение автора, в одно время он может дать положительную оценку одной части услуг/товар и отрицательную другой.

2) Классификация по многополосной шкале [8, 9].

Логическим продолжением бинарной классификации является многополосная шкала или система шкалирования. В шкалу оценивания добавляются пункты: отзывы делятся уже не просто на положительные и отрицательные, а на резко положительные и просто положительные, удовлетворительные, негативные и резко отрицательные. Этот метод позволяет точнее понять ситуацию.

3) Субъективность/объективность [11].

Важным видом классификации является определение субъективности слов и фраз, имеется в виду сильная зависимость от контекста.

В данной работе будет использоваться бинарная классификация, как наиболее базовый метод деления входящего потока информации. В дальнейшем можно усложнять алгоритм, переходя к системе шкалирования.

Началом широкой осведомленности о проблемах и возможностях исследований, которые открывают анализ настроений и анализ мнений можно считать 2001 год, и впоследствии было опубликовано буквально сотни статей на эту тему. Впервые термин «тональность» появился в работах [12, 13], но тогда исследования в данной области носили достаточно наивный характер, например, в [12] считались упоминания продукта в Интернете, и от количества упоминаний зависел уровень репутации продукта.

На сегодняшний день существует три подхода к анализу тональности текстовых сообщений:

1) Метод, основанный на правилах и словарях [14].

Данный подход интуитивно понятен, из последовательностей слов или отдельных терминов, встречающихся в тексте, создается словарь. Далее

каждому элементу словаря присваивается оценка согласно выбранному типу классификации. Таким образом, разработчик самостоятельно генерирует правила деления слов и ищет полученные шаблоны в новых текстах. Явным минусом данного метода является серьезная подготовка исходных данных и нединамичное обновление словаря.

2) Машинное обучение без учителя [15].

При использовании такого способа работы с текстом за основу берется принцип, что наибольший вес имеют слова, которые часто встречаются в конкретном блоке текста и редко встречаются в остальном тексте. Выделив эти ключевые слова, можно определить тональность всего текста. Однако такой подход скорее лучше работает для структурированных текстов, чем для естественного языка.

3) Машинное обучение с учителем [16].

Этот метод можно в некотором смысле считать комбинацией первых двух: он также требует предварительной подготовки исходных данных, но для принятия решения использует алгоритмы машинного обучения. На вход подаются размеченные данные, на которых обучается классификатор, а после этого проходит его валидация на совершенно новых текстах. Очевидным плюсом является то, что он не привязан к статичному словарю и обучение происходит не на одном ключевом слове, а на совокупности. При правильном выборе технологии подготовки данных можно учесть все значимые зависимости между словами и определить тональность более точно.

Большой вклад в развитие анализа тональности текстовых сообщений внесли исследователи из Корнельского университета Б. Пэнг и Л. Ли. В 2008 году они выпустили книгу [17], посвященную современным методам и подходам к анализу тональности в текстовых сообщениях. В их работе [18] рассматривается классификация тональности с использованием машинного обучения и показывается, что такой подход превосходит простые техники, основанные на составлении словарей часто употребляемых позитивных и

негативных слов. В связи с этим в данной работе был использован такой подход.

Тем не менее, при использовании любого из перечисленных выше методов необходимо выполнить предварительную обработку анализируемого текста для выделения слов и сочетаний, несущих эмоциональную нагрузку, и приведения их к одинаковой грамматической форме. В работе [19] подробно изложены сложности обработки естественного языка, нужно понимать, что мы работаем с данными, состоящими из морфем, соединяющихся на синтаксическом уровне, что у каждого слова или словоформы есть свой лексический смысл. Интересный подход к предварительной обработке данных был рассмотрен в статье [4] авторы уделили внимание не только самим словам, но и эмоциональной пунктуации, они использовали в работе метод основанный на словарях и добавили в него пунктуационное представление положительных и отрицательных смайлов.

При использовании при анализе тональности текста подходов на основе алгоритмов машинного обучения после предобработки текста требуется выполнить его перевод в вещественное пространство признаков. Для этого, например, авторы [4] используют TF-IDF классификатор, где глобальный вес рассчитывается по релевантной частоте (RF). В методе RF (Relevance Frequency — релевантная частота) для вычисления глобального веса термина используется информация о распределении этого термина по документам обучающей коллекции с учетом принадлежности документов к классам.

Другим популярным методом представления слов в векторном виде является технология Word2Vec. Ей посвящено достаточно много литературы и различной документации, этот подход взят в основу практически всех исследований по обработке естественного языка. В работе [20] технология Word2Vec сравнивается с другими менее популярными способами обработки текстовой информации, такими как LSA и Glove. Метод GloVe (Global Vectors)[21] также основан на частоте встречаемости слова в тексте, а в то время как латентно-семантический метод (LSA) принимает параметрами

функции взвешивания и размерность семантического пространства.[22] это статистический алгоритм, который также как TF-IDF классификатор основан на частоте встречаемости слова.

Непосредственно задача определения тональности сводится к задаче классификации, для решения которой используются методы машинного обучения. В работе [20] сравниваются разные методы, например, метод Rocchio [23] и метод k ближайших соседей [24], которые в сравнении с наивным байесовским классификатором [25] и методом опорных векторов [26] дали существенно худшие результаты.

В статье [27] выявлена интересная особенность технологии Word2Vec, там описывается эксперимент по поиску синонимов медицинских терминов. И алгоритм показал хорошую точность работы в предметной области.

Стоит отметить, что методы машинного обучения для решения задачи классификации тональности текстовых сообщений в основном активно развиваются за рубежом, т.к. подавляющее большинство статей принадлежит иностранным ученым.

К настоящему времени создан ряд автоматизированных продуктов, разработанных зарубежными и отечественными специалистами, которыми может воспользоваться любой желающий. Они помогают подготовить предобработку данных для последующего применения на них классификаторов. Это предварительно обученные модели чаще всего реализованные на языках Python или R:

1) Stanford NLP [28] – это полностью доступная демо-модель Стэнфордского университета для определения тональности рецензий на фильмы. В основе системы лежат рекурсивные нейронные сети. Из минусов то, что пока данная модель доступна только для англоязычного материала.
2) Sentiment140 [29] – еще одно решение, разработанное выпускниками Стэнфорда. Данная модель применяется к микроблогам Twitter и позволяет получить любому пользователю в ответ на свой запрос

выборку позитивных/негативных/нейтральных твиттеров. Выгодное отличие продукта к том, что он визуализирует результат с помощью инфографики, но работает также с ограниченным количеством языков: английским, испанским.

3) 30dB [30] – открытая платформа. Принцип работы аналогичен предыдущим инструментам, пользователь генерирует запрос и получает в ответ эмоциональную окраску относительно заданной темы. В качестве источников для анализа может использовать Twitter, Facebook, Google+.Преимуществом данного сервиса является возможность провести сравнение двух введенных тем сразу. К сожалению, поддерживает только англоязычную лексику.
4) ВААЛ [31] – это разработка русского происхождения. С помощью этой системы можно анализировать тексты и четче выявлять психологические качества авторов текстов, плюс эта платформа производит автоматическую категоризацию текста.

Глава 1. Предварительная обработка текста

Существует несколько этапов обработки естественного языка. И для каждого этапа есть свой набор функций (библиотек, например, на языке Python), которые позволяют машине понять текст, также как мы его понимаем.

Интересно изложены этапы анализа текста:

- Графематический анализ [19]- это деление всего текста на предложения, словосочетания или словоформы (их еще называют *токены*). Суть данного процесса в том, чтобы перестать воспринимать буквы, как отдельные символы, а группировать их по словам, к которым они относятся.
- Морфологический анализ [19] позволяет из каждой словоформы или морфемы выделить её коренную часть и
- Синтаксический анализ [19]- это один из важнейших этапов работы с текстом. Он помогает выявить связи между словами и их роль в предложении.
- Семантический анализ [19] отвечает за смысл фраз. Только на этом этапы можно определить, какую эмоциональную окраску несет в себе текст.

Глава 2. Построение векторной модели текста

В настоящей работе для анализа тональности текстов, как было отмечено выше, используется метод обучения с учителем. Однако в реальности нельзя просто передать текст классификатору, необходимо предварительно подготовить эти данные, представив их в виде векторной совокупности признаков.

В настоящее время самыми популярными методами подготовки текстовых данных являются технология Word2Vec, созданная Google в 2016 году, и BagOf Words, в основе которой лежит принцип TFIDF анализа. Данная глава посвящена описанию и сравнению особенностей каждого из представленных выше методов.

2.1. Метод Word2Vec

Слова можно кодировать различными способами, самый простой из них – сделать словарь из слов, встречающихся в предложении, и присвоить им порядковый номер. Однако такой способ не несет никакой смысловой нагрузки. Необходимо понимать, насколько близки друг к другу лексические значения слов.

Выход из этой ситуации был найден в 2013 году аспирантом Томашем Миколовым и назвал он эту технологию word2vec. В основе его подхода лежит, так называемая, гипотеза локальности, которая утверждает, что на объект влияет только его близкое окружение, здесь конечно, есть оговорка, можно ли с уверенностью сказать, что рядом всегда стоят только влияющие слова?

Модель, которую предложил Миколов, может рассчитать вектор слова по окружающим словам. Здесь не выделяется одно ключевое слово, отвечающее за настроение всего текста, а берется вместе с контекстом. Таким образом, векторное представление близких друг к другу слов будем мало отличаться. Математически это записывается следующим образом:

$$P(w_0|w_1) = \frac{e^{s(w_0,w_c)}}{\sum_{w_1 \in V} e^{s(w_i,w_c)}} \ . \quad (1)$$

Здесь w_0 — вектор ключевого слова, w_1 — это вектор слов, окружающих целевое слово, вычисленный одним из возможных методов, например усреднением или суммой векторов контекста. А $s(w_0|w_1)$ — это функция, которая паре векторов ставит в соответствие число. В основе этой функции может лежать мера сходства между двумя ненулевыми векторами внутреннего пространства произведений, которое измеряет косинус угла между ними. Формулу (1) можно дифференцировать и обучить методом обратного распространения ошибки. В основе процесса тренировки алгоритма лежит следующий подход: есть слово, вектор которого мы хотим вычислить, для этого берем (2k+1) слов, окружающих целевое, последовательно друг за другом. Они будут отвечать за контекст, длиной k в каждую сторону от центра. Каждому из этих слов будет сопоставлен уникальный вектор, полученный путем усреднения.

Такой подход по сути аналогичен BoW (bag of words) за одним исключением: алгоритм получает наборы слов последовательно, то есть важным является порядок, с которым обучается инструмент, но вот расположение самих слов внутри последовательности роли не играет. Если останавливаться на модели CBOW подробнее, то важно сказать, что в ее основе лежит минимизация векторов слов, окружающих ключевое слово, с помощью дивергенции Кульбака-Лейблера:

$$KL(p||q) = \int p(x) log \frac{p(x)}{q(x)} dx \quad (2)$$

Здесь $p(x)$ — это распределение векторов, которые мы берем из контекста, $q(x)$— распределение, которое дает модель. Дивергенция — это разность между распределениями. В нашем случае мы работаем со словами, а значит с дискретными величинами, тогда мы можем использовать сумму вместо интеграла:

$$KL(p||q) = \sum_{v \in V} p(x) log \frac{p(x)}{q(x)} dx \quad (3)$$

Минимизировать данную формулу сложно, в первую очередь из-за того, что минимум рассчитывается по всему объему исходных данных. А учитывая то, что большая часть слов из всего набора не встречается в конкретной последовательности, то часть вычислений излишне. Выходом из данной ситуации служит метод Negative Sampling. Он усложняет формулу (3), но помогает ограничить область данных, с которой мы работаем. Мы с одной стороны вычисляем максимум вероятности того, что мы встретим ключевое слово в типичном для него контексте, а с другой стороны находим минимум вероятности столкнуться с этим же словом в совершенно нетипичном для него окружении.

Эта идея математически записывается следующей формулой:

$$NegS(w_0) = \sum_{i=1,x_i\sim D}^{i=k} -\log\left(1 + e^{s(x_i,w_0)}\right) + \sum_{i=1,x_i\sim D,}^{i=l} -\log\left(1 + e^{-s(x_i,w_0)}\right) \quad (4)$$

Здесь $s(x, w)$ выполняет все ту же роль функции соответствия, но в остальном есть изменения. Теперь в формуле присутствуют две составляющие: позитивная ($s(x, w)$) со знаком плюс и негативная с отрицательным знаком. Как ясно из определения выше, позитивная стороны отвечает за максимум вероятности в типичном окружении, а негативная, за атипичность. Пространство D здесь — это распределение вероятность встречаемости слова и остальных слов корпуса. Под негативной составляющей мы понимаем совокупность слов, которые очень редко или никогда не встречаются с нашим ключевым словом. Собрать такие слова сложная задача, в реальности Негативная часть — это, пожалуй, самое интересное — это набор слов, которые с нашим целевым словом встречаются редко. В исследованиях [9] данный метод дал хорошую точность.

В своей технологии Word2Vec Миколов соединил два подхода, к уже изученному CBOW он добавил противоположный метод skip-gram, если переводить дословно, то “словосочетание с пропуском”. Суть этого метода в том, что мы наоборот концентрируемся на ключевом слове и по нему угадываем, в каком контексте и с какими словосочетаниями оно может быть употреблено. Здесь важно отметить самое главное качество модели Word2Vec,

которое выгодным образом отличает ее от всех других, это семантика. Хотя она и не заложена напрямую, но хорошо обученная модель может улавливать смысловую зависимость близких по лексическому значению слов. Классический пример из работы автора [6]:

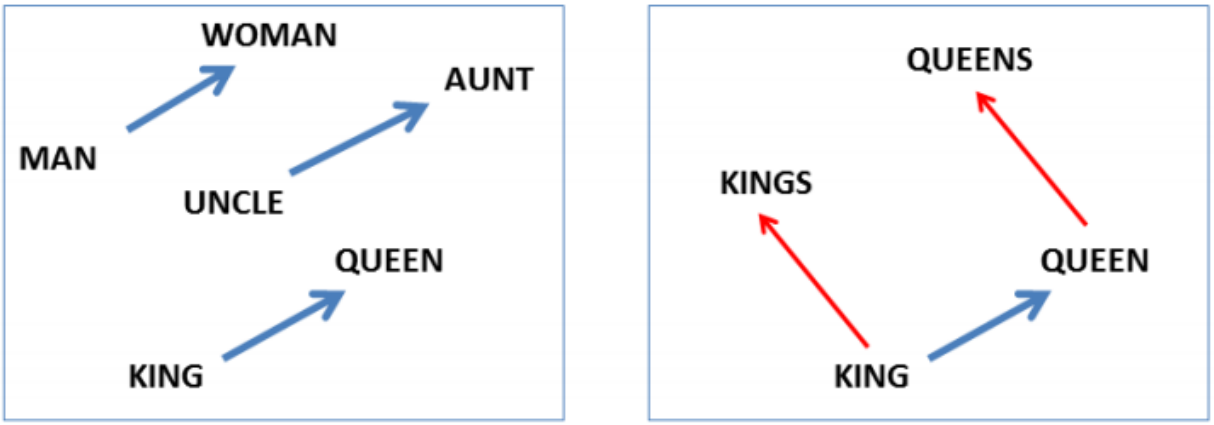


Рис.1.Визуализация связей между словами.

На Рис.1 видно, что благодаря тому, что стоящие рядом слова становятся близкими по векторному представлению, между ними получается обнаружить связи. Самый популярный пример работы этого модуля: если от вектора слова «king» отнять вектор «man» и прибавить «woman» получится «queen».

2.2. Метод Bag Of Words

В основе данного метода лежит TF-IDF (term frequency-inverse document frequency) алгоритм. Смысл метода в том, что если термин встречается с высокой частотой в одном тексте и практически не встречается во всех остальных текстах, то он имеет большую значимость для этого текста. Из явных плюсов данного метода хорошее выявление не значимых слов, таких как предлог или вводные слова, они участвуют во всех текстах, а значит имеют маленький вес.

Рассмотрим формулу расчета подробно.

TF (дословно: частота слова) рассчитывается как количество появлений слова в тексте к ко всему числу слов в текстах, таким образом вычисляется важность слова t_i в тексте:

$$tf(t,d) = \frac{n_t}{\sum_k n_k}, \qquad (5)$$

где n_t - количество появлений слова t в тексте, а в знаменателе — общее число слов в текстах.

IDF — обратная частота документа, она измеряет важность данного слова для всех документов. Он рассчитывается, как логарифм (причем можно брать любой логарифм, для простоты часто берут десятичный или натуральный, в связи с тем, что TF-IDF выражается относительно друг друга мерой). С его помощь удается избавиться от часто встречающихся слов, например, предлогов. Важно отметить, что для каждого уникального слова есть только один IDF.

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

где |D| — количество всех текстов;

$|\{d_i \in D | t \in d_i\}|$ — число документов из D, в которых появляется слово t (когда $n_t \neq 0$).

Важно заметить, что в основании логарифма может стоять любое число, так как происходит домножение веса слова на константу, значение дроби это не изменяет.

Теперь, если подставить значения TF и IDF соответственно:

$$tf - idf(t, f, D) = tf(t, d) \times idf(t, D)\ . \qquad (7)$$

Суть данного метода в том, что будут иметь больший вес слова, которые часто встречаются в конкретном тексте и редко во всех остальных.

Глава 3. Определение тональности текста с помощью алгоритмов машинного обучения

Данная глава посвящена рассмотрению основных методов машинного обучения, используемых для анализа тональности текстов, а именно: наивный байесовский классификатор , метод опорных векторов (SVM) и метод градиентного бустинга.

3.1. Наивный байесовский классификатор.

В основе данного метода лежит теорема Байеса для расчета вероятности какого-либо события при условии, что произошло другое связанное с ним событие.[7]

То есть, используя формулу Байеса, можно рассчитать вероятность события, беря в расчет, как уже произошедшие события так и грядущие. Она выглядит следующим образом:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (8)$$

Где

- P(A) — безусловная вероятность, того, что событие A произошло, за неимением обратной информации, также это называют априорной вероятностью;
- P(A|B) — вероятность того, что событие A произошло с условием, что также произошло событие B (апостериорная вероятность);
- P(B|A) — вероятность наступления события B при истинности гипотезы A ;
- P(B) — полная вероятность наступления события B .

В машинном обучении существует целое семейство классификаторов, основанных на теореме Байеса и допущении, что признаки объектов, которые мы классифицируем, не зависят друг от друга. Впервые данные метод нашел свое применение еще в 1950 годах, его использовали для классификации документов, где признаками были частоты слов. Плюс этого алгоритма в том, что мы можем практически неограниченно увеличивать количество признаков, что будет приближать точность прогноза к таким популярным методам, как метод опорных векторов.

Принцип работы байесовского метода ничем не отличается от других классификаторов, мы относим наблюдения к тому или иному классу по векторам признаков. Но делаем это с важным допущением, считая, что каждый признак не связан с другими и влияет на классификацию. Если рассматривать простой байесовский классификатор, то в его основе лежит обучение с учителем. Также преимуществом является то, что ему нужен небольшой набор входных данных для обучения. Поэтому несмотря на то, что его неспроста называют «наивным», благодаря маловероятному утверждению о независимости признаков, он уже долгое время удерживает место среди самых точных классификаторов.

Рассмотрим вероятностную модель, лежащую в основе алгоритма. Есть множество событий (наблюдений) $x = (x_1, x_2, \dots, x_n)$. Алгоритм соотносит каждому наблюдению условную вероятность $p(C_k|x_1, x_2, \dots, x_n)$, где C_k-это класс.

Воспользовавшись теоремой Байеса:

$$p(C_k|x_1, x_2, \dots, x_n) = \frac{p(C_k)p(x|C_k)}{p(x)} \quad (9)$$

Здесь наибольший интерес вызывает числитель, так как знаменатель никак не участвует в задаче классификации и будет константой. Благодаря предположению, что все признаки независимы:

$$p(C_k|x_1, x_2, \dots, x_n) = p(C_k)p(x_1|C_k)p(x_2|C_k) \dots p(x_n|C_k) = \prod_n p(x_i|C_k) \quad (10)$$

Получается простой байесовский классификатор присваивает каждому значению метку принадлежности к классу, т.е. $y = C_k$ выглядит следующим образом:

$$y = arg_k max_{1\dots k} \prod_n p(x_i|C_k) \quad (11)$$

Класс y выбирается таким образом, чтобы максимизировать функцию правдоподобия, которая представляет собой произведение условных вероятностей признака x_i.

Наивный Байес предсказывает класс с наибольшей условной вероятностью для вектора признаков *x*.

3.2. Метод опорных векторов.

Классификация данных – важная задача машинного обучения и сейчас метод опорных векторов один из наиболее популярных алгоритмов, применяющихся для решения.

Предположим, что все объекты можно отнести к одному из двух классов. Проблема состоит в том, что необходимо определить, к какому классу будут относится новые объекты. Именно в такой ситуации применим метод SVM. Точка в пространстве будет рассматриваться как вектор размерности p и нужно понять, получится ли разделить данные гиперплоскостью размерности (p-1). По сути эта гиперплоскость будет выполнять роль классификатора и она может быть не единственной, но метод будет давать хорошую точность только в тех случаях, если деление между 2 классами будет максимально. Предположим, что наш вектор представим в таком виде (см. рисунок 1) и гиперплоскостью будет прямая. Можно провести любую прямую и она разделит эти точки, но можно провести одну линию, расстояние до которой будет максимально большим от всех точек:

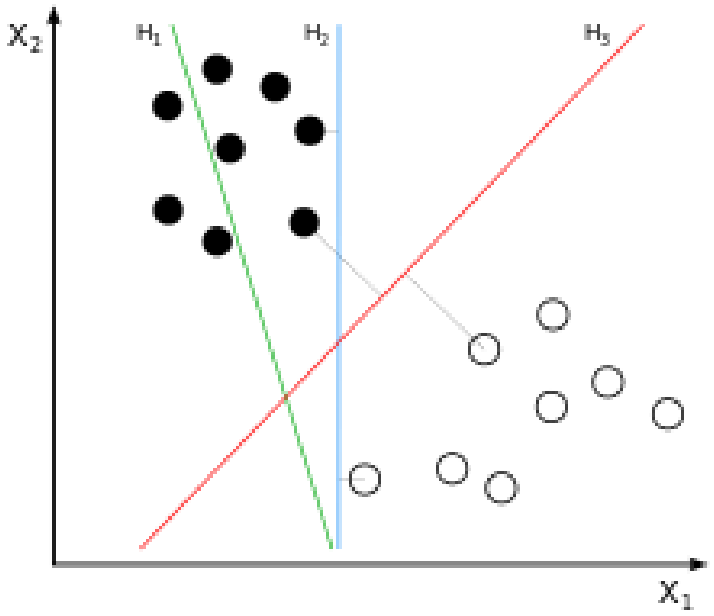


Рис.1.H1,H2,H3–гиперплоскости. H3 – гиперплоскость максимального расстояния.

D-это обучение, а (x_i, y_i) набор из n объектов:

$$D = \{(x_i, y_i) | x_i \in R^p, y_i \in \{-1,1\}\}_{i=1}^{n} \quad (12)$$

y будет определять, к какому из двух классов относится точка x_i, а каждая точка представляет собой вектор размерности p.

То есть задача состоит в том, чтобы найти такую гиперплоскость максимальной разности, разделяющую объекты.

Уравнением такой прямой является вид:

$$w * x - b = 0 \quad (13)$$

Где под * имеется в виду скалярное произведение нормали к гиперплоскости.$\frac{b}{||w||}$ определяет смещение гиперплоскости относительно начала координат.

В ситуации, когда данные разделимы неединственным способом, можно провести две гиперплоскости, а дальше пытаться максимизировать расстояние между ними. Их можно описать такими уравнениями:

$$w * x - b = 1$$

$$w * x - b = -1 \quad (14)$$

Аналогичным способом можно определить смещение между ними: $\frac{2}{||w||}$.Логично, что сделать эту дробь максимальной можно только в случае, если знаменатель минимален. Накладываем условие на первый блок уравнений:

$$w * x_i - b \geq 1$$

$$w * x_i - b \leq -1 \quad (15)$$

Это эквивалентно:

$$y_i(w * x_i - b) \geq 1 \quad (16)$$

где $0 \leq i \leq n$.

Мы приходим к задаче поиска минимума:

$$||w|| \rightarrow min \quad (17)$$

Однако данная задача сложно решается в условиях, если нам неизвестна норма w, которая будет включать в себя квадратный корень. Если мы упростим задачу, заменив в формуле (17) на $\frac{1}{2}||w||^2$. Тогда эта задача модифицируется в квадратичную задачу поиска минимизации:

$$\frac{1}{2}||w||^2 \rightarrow min \quad (18)$$

При этом еще накладываются ограничения (16). Если мы введем множители Лагранжа в уравнение, то оно получит следующий вид, таким образом сняв с себя ограничения:

$$\min_{w,b} \max_{\alpha \geq 0} \left\{ \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i * [y_i(w * x_i - b) - 1] \right\} \quad (19)$$

Теперь мы ищем седловую точку. Можно убрать и условия 16 знак больше, так как нам достаточно будет равенства в уравнении.

Используя теорему Куна-Такеру решение может быть представлено линейной комбинацией обучающих векторов:

$$w = \sum_{i=1}^{n} a_i y_i x_i \qquad (20)$$

Из этого множество только часть векторов а больше нуля, а x_i принадлежит уравнение (16), решенному при условии равенства левой и правой части.

Из этого следует:

$$w * x_i - b = \frac{1}{y_i} = y_i \leftrightarrow b = w * x_i - y_i \qquad (21)$$

На практике применяют усреднение по всем опорным векторам и конечная формула выглядит следующим образом:

$$a(x) = sign\left(\sum_{i=1}^{n} a_i y_i x_i * x - b\right) \qquad (22)$$

При чем суммирование идёт не по всей значениям, а только по опорным векторам, для которых $a_i \neq 0$.

3.3. Алгоритм градиентного бустинга.

На сегодняшний день алгоритм градиентного бустинга – один из самых популярных методов машинного обучения, он направлен на последовательное построение композиций алгоритмов.

Композицией T алгоритмов $a_t(x) = C(b_t(x))$, t=1, . . . , T является суперпозиция алгоритмов b_t : X → R, корректирующей операции F : R^T → R и решающего правила C : R → Y:

$$a(x) = C\left(F\left(b_1(x), \ldots, b_T(x)\right)\right) \qquad (23)$$

Итак, построим композицию:

$$a_N(x) = \sum_{n=1}^{N} b_n(x) \qquad (24)$$

В нашем случае композиция представляется в виде суммы N алгоритмов $b_n(x)$. В случае с градиентным бустингом мы не берем среднее из всех значений базовых алгоритмов, а складываем их, потому что каждый

последующий алгоритм помогает скорректировать предыдущий. Также при работе нельзя забывать о функции потерь, которая измеряет ошибку для каждого из объектов, ее соответственно нужно привести к минимуму:

$$L(y, z)\,, \tag{25}$$

где y- это истинное значение на данном объекте, а z- прогноз алгоритма. Функция L может быть вычислена по-разному в зависимости от типа задачи, для задач регрессии обычно используется среднеквадратичная ошибка, а для задач классификации используется логистическая регрессия вида:

$$L(y, z) = \log(1 + e^{-yz})\ . \tag{26}$$

Начнем с того, что надо построить первый базовый алгоритм $b_0(x)$.Его можно взять даже просто константным значением, равным 0 например, но это применимо только к задачам регрессии, а если, как в нашем случае, мы работаем с задачей классификации, то необходимо решать задачу максимизации или минимизации функции:

$$b_0(x) = argmax_{y \in Y} \sum_{i=1}^{l} [y_i = y] \tag{27}$$

Дальше следуем по индукции. Предположим, что базовый алгоритм N-1 построен

$$a_{N-1}(x) = \textstyle\sum_{n=0}^{N-1} b_n(x)\,. \tag{28}$$

Теперь предполагаем, как будет выглядеть алгоритм b_n, чтобы ошибок на обучающей выборке была как можно меньше. Для этого просуммируем потери суммы алгоритмов a_{N-1} и последнего алгоритма $b(x)$ и минимизируем полученный функционал:

$$\sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + b(x_i)) \to min_b \tag{29}$$

Чтобы упростить для понимания задачу, попробуем представить, какие конкретно значения может принимать функция в каждой точки x_i, чтобы уменьшить ошибку. Перепишем функционал в другом виде:

$$\sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + s_i) \to min_{s_1,\dots,s_l} \qquad (30)$$

где s_i -это корректирующий сдвиг прогноза на i-том объекте, значит надо нам найти последовательность таких s_i , которые максимально уменьшат функцию потерь. Таким образом, задача сводится к оптимизационной: нужно найти вектор s минимизирующий функцию

$$F(s) = \sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + s_i) \to min_s \qquad (31)$$

Известно, что направление вектора градиента совпадает с направлением наибольшего возрастания функции. В нашем случае решается задача минимизации, потому рассматривается вектор антиградиента:

$$s = -\nabla F = (-L'_z(y_1, a_{N-1}(x_1), \dots, -L'_z\big(y_l, a_{N-1}(x_l)\big) \qquad (32)$$

где каждый элемент равен сдвигу на соответствующем объекте, то есть частной производной функции L по следующему элементу. В случае с первым элементом берется прогноз на объекте x_1, взятый с отрицательным знаком, в случае же с последним элементом берется также частная производная функции L с отрицательным знаком, прогноз вычисляется на объекте x_l.

В ходе данной работы были получены значения s_i, которые должна принимать функция b(x) на элементах обучающей выборки и выдавать прогноз в любой точке. Получается мы сталкиваемся с задачей обучения на размеченных данных, теперь если двигаться по алгоритму, представленному выше: нужно подобрать функционал $b_n(x)$ так, чтобы он стал максимально близок к сдвигам и представим его в виде среднеквадратичного отклонения.

$$b_N(x) = argmin_b \frac{1}{l} \sum_{i=1}^{l} (b(x_i) - s_i)^2 \qquad (33)$$

Значения алгоритма b (x) в точках сдвига s_i это сумма квадратов отклонений. Важно отметить, что на первый взгляд функция потерь здесь не присутствует, но на самом деле она уже содержится в s_i, таким образом у нас пропадает необходимость искать функцию L на каждом шаге алгоритма, все

что нужно это уменьшать функцию (33). Большинство подобных задач решаются таким образом. Вот это суть метода, использующегося при градиентном спуске.

Если подводить итог, то весь метод градиентного спуска можно свести к небольшому понятному алгоритму, который помогает решить до 90 процентов задач классификации, связанных с обучением на размеченных данных. В первую очередь мы выбираем первый базовый алгоритм b_0, в задача классификации это обычно функция, усредняющая значения или возвращающая максимальное/минимальное значение. Далее мы циклично будем строить новые базовые алгоритмы b_n,для этого мы работаем с вектором s, который отвечает за сдвиг алгоритма и помогает корректировать прогнозные значения уже построенной модели, так как он представляет собой частные производные функции L в точках выборки, то нам удается значительно уменьшить функцию потерь на этапе обучения. После этого вы возвращаемся к базовым алгоритмам и строим b_n так, чтобы максимально приблизить его значение в точках к сдвигам s_i. После того как алгоритм построен возвращаемся к композиции и добавляем его туда:

$$a_n(x) = \sum_{m=1}^{n} b_m(x) \qquad (34)$$

После этого повторяем все перечисленные шаги пока не получим сходимость или критерий остановки. Таким образом, градиентный бустинг строит a_n последовательно и каждый алгоритм корректирует предыдущий, приближая вектор антиградиента к выборке и уменьшая ошибку уже реализованной композиции во время обучения. Градиентный бустинг можно также назвать градиентным спуском по всем возможным алгоритмам и функциям, где каждая итерация отправляет нас к функции b_n.

Существенным минусом данного метода является его предрасположенность к переобучению.

Переобучение – это негативное явление, когда модель очень хорошо работает на тестовых данных, на которых он обучилась, и выдает очень низких результаты при валидации на новых данных.

Глава 4. Практическая реализация и результаты

4.1. Формулировка задачи

В качестве объекта исследования было выбрано отношение к банковским продуктам у нас в стране. За источник данных для обучения и последующей валидации был взят сайт banki.ru, на котором собрано большое количество информации, новостей и комментариев о работе всех банков, существующих на данный момент в России. Основное внимание уделяется страницам, на которых люди оставляют отзывы об обслуживании в офисах банка, о банковских продуктах, таких как кредиты, ипотеки, овердрафтные системы и вклады. Эти отзывы написаны максимально «живым» языком и, безусловно, передают эмоции автора: злость, благодарность, недовольство или радость.

Общий план работы для достижения поставленной в работе цели выглядит следующим образом:

1) автоматический сбор данных с веб ресурса;
2) чистка этих данных (удаление предлогов, вводных конструкций) и приведение к виду dataset формата: пары {оценка (то, что мы должны будем в дальнейшем предсказать); отзыв};
3) создать из имеющихся отзывов набор слов, который можно было бы передать в алгоритмы преобразования текстовой информации в векторную.
4) применить алгоритмы Word2Vec и Bag Of Words;
5) передать полученные векторные представления в классификатор градиентного бустинга;
6) сравнить полученные значения точности корректного определения положительного/отрицательного настроя отзыва при использовании методологии Word2Vec и Bag Of Words;
7) провести валидацию на неразмеченных данных.

4.2. Сбор данных с web-ресурса.

Для работы с классификаторами необходимо подготовить базу, на которой будет обучаться алгоритма. Собранные данные с сайта *banki.ru* были разделены на обучающую выборку и тестовую, на которой в последствии проводилась валидация классификаторов.

Изначально данные на сайте выглядят следующим образом (*Рис.2*):

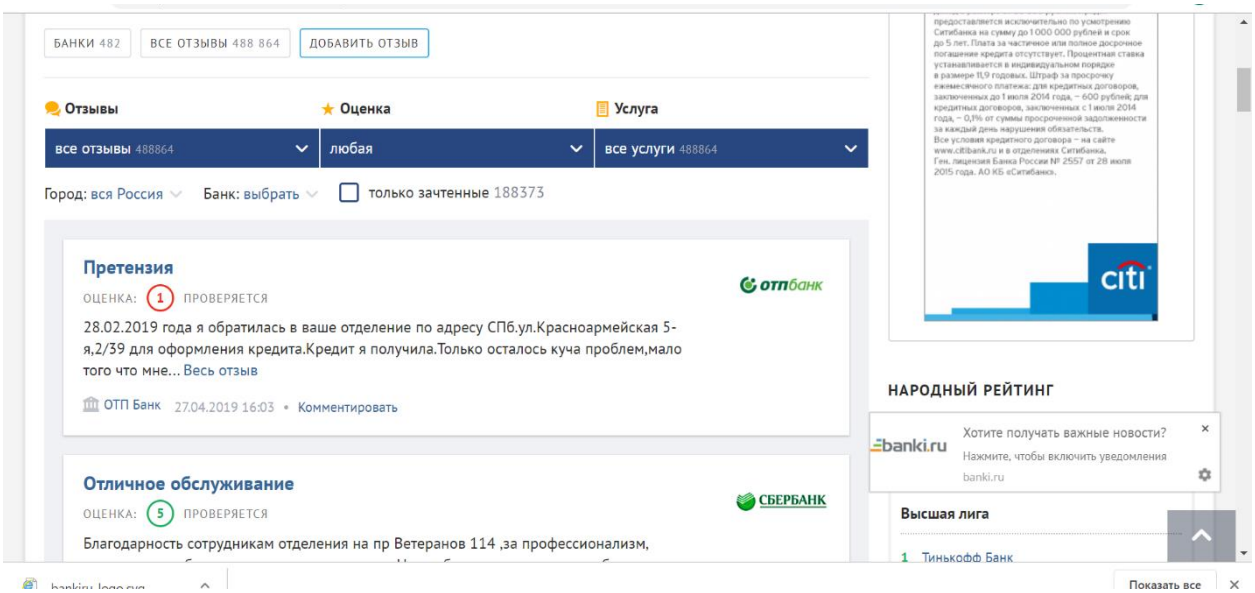


Рис.2.Данные с сайта banki.ru

Для работы использовался язык программирования Python, он был выбран исходя из того, что со все классификаторами машинного обучения и алгоритмами работы с текстом и веб- источниками удобно работать на этом языке.

Для сбора данных была использована библиотека Request, с помощью функции get был получен объект типа Response, с помощью которого далее считывался ответ сервера. С помощью функции text, производится автоматическое декодирование html-кода. Библиотека Request работает особым образом: ее работа с кодировкой основана на HTTP-заголовках, строках в HTTP-сообщении, которые содержат пару имя-значение, разделенную двоеточием. Таким образом, с помощью функций данной библиотеки был выделен html- код страницы, содержащей всю необходимую нам информацию для дальнейшей работы. Благодаря универсальному виду

ссылок был реализован цикл, с помощью которого была пройдена сотня страниц с отзывами и осуществлен переход на каждый из них, чтобы забрать весь текст.

Следующим важным этапом необходимо «распарсить» полученный код, иными словами, нужно извлечь важную нам информацию из большого количества полученных данных. Для этого на языке Python был изобретен синтаксический парсер- BeautifulSoup. Он специализируется на HTML/XML-документах и может работать даже с неправильной разметкой, преобразуя ее в дерево синтаксического разбора. Поддерживает простые и естественные способы навигации, поиска и модификации дерева синтаксического разбора. Для работы конструктору Beautiful Soup требуется документ XML или HTML в виде строки (или открытого файлоподобного объекта). Он произведет синтаксический разбор и создаст в памяти структуры данных, соответствующие документу.

Если обработать с помощью Beautiful Soup хорошо оформленный документ, то разобранная структура будет выглядеть также как и исходный документ. Но если его разметка будет содержать ошибки, то Beautiful Soup использует эвристические методы для построения наиболее подходящей структуры данных. Обратите внимание на то, что BeautifulSoup вычисляет наиболее вероятные места для закрывающих тегов, даже если они отсутствуют в исходном документе.

Благодаря функциям из этой библиотеки удалось преобразовать исходные данные в вид (Рис. 3):

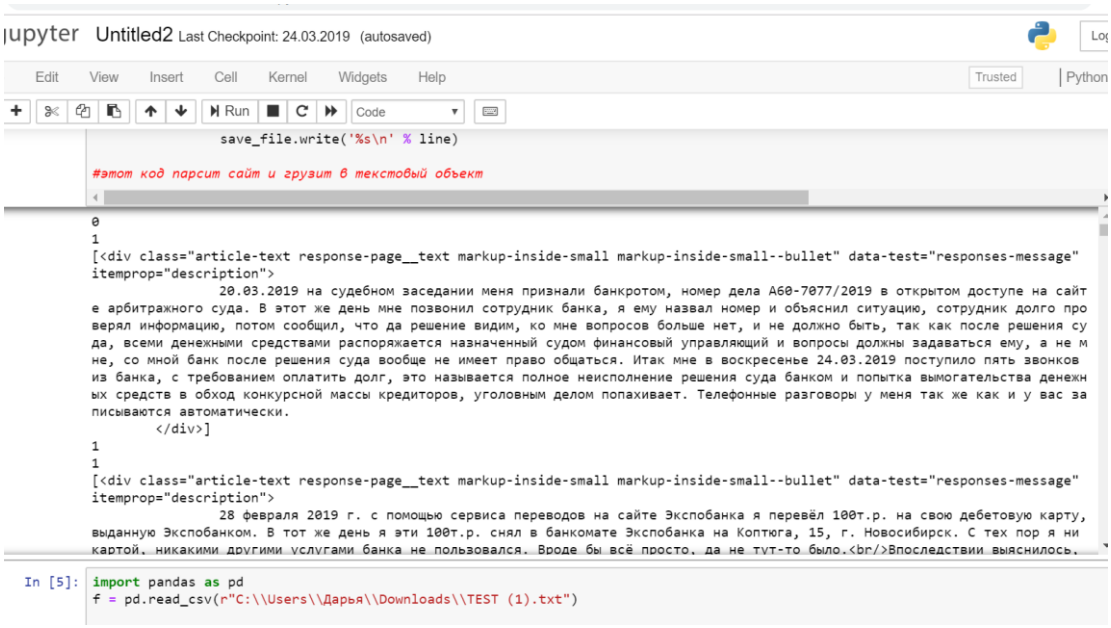
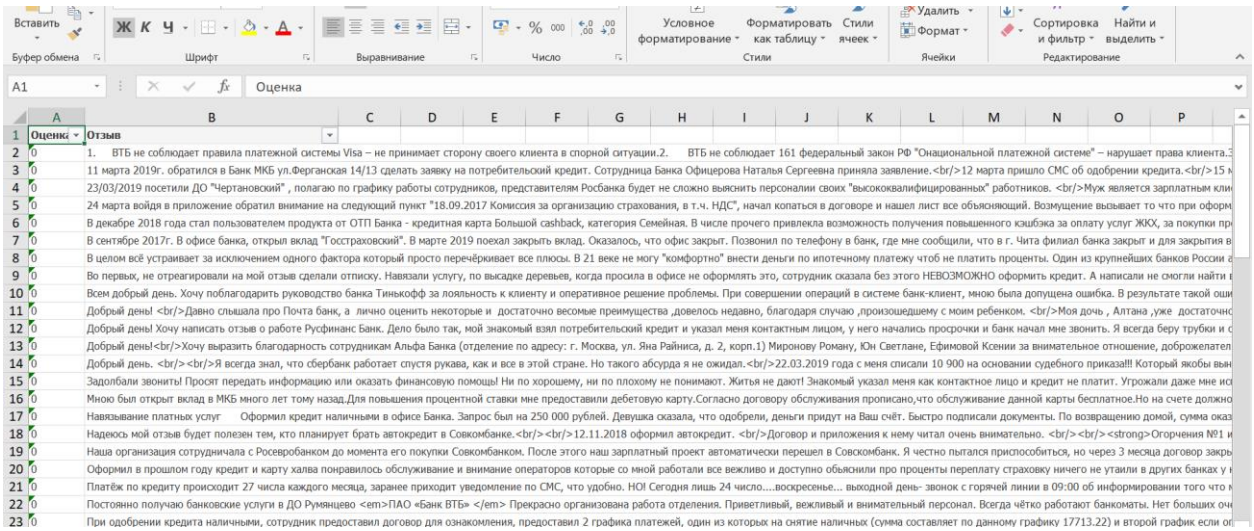


Рис.3. Вид данных, полученный после парсинга html-страницы.

Как видно из рисунка, мы забираем не только текст отзыва, но оценку, которую поставил человек. Это нужно для того, чтобы представить алгоритму размеченные данные, на которых он будет обучаться.

Сразу хочется сказать, в ходе работы стало заметно частое несоответствие оценки и отзыва, люди исходя из собственных взглядов могут написать резко негативный отзыв, поставив при этом недостаточно низкую оценку, верна и обратная ситуация. Все это может значительно ухудшить точность прогноза. В подобных условиях практически невозможно корректно предсказать результаты по системе шкалирования, то есть отличить резко негативный отзыв от негативного или удовлетворительного сложно, во многом из-за этого в работе была выбрана бинарная классификация, чтобы уменьшить процент ошибки.

Далее эти данные были собраны в csv. файл и преобразованы в формат dataset: текст отзыва и оценка, используя приложение для работы с базами данных Qlikview. Они приобрели следующий вид:



	A	B
1	Оценка	Отзыв
2	0	1. ВТБ не соблюдает правила платежной системы Visa – не принимает сторону своего клиента в спорной ситуации.2. ВТБ не соблюдает 161 федеральный закон РФ "Онациональной платежной системе" – нарушает права клиента.3
3	0	11 марта 2019г. обратился в Банк МКБ ул.Ферганская 14/13 сделать заявку на потребительский кредит. Сотрудница Банка Офицерова Наталья Сергеевна приняла заявление. 12 марта пришло СМС об одобрении кредита. 15 к
4	0	23/03/2019 посетили ДО "Чертановский" , полагаю по графику работы сотрудников, представителям Росбанка будет не сложно выяснить персоналии своих "высококвалифицированных" работников. Муж является зарплатным кли
5	0	24 марта войдя в приложение обратил внимание на следующий пункт "18.09.2017 Комиссия за организацию страхования, в т.ч. НДС", начал копаться в договоре и нашел лист все объясняющий. Возмущение вызывает то что при оформ
6	0	В декабре 2018 года стал пользователем продукта от ОТП Банка - кредитная карта Большой cashback, категория Семейная. В числе прочего привлекла возможность получения повышенного кэшбэка за оплату услуг ЖКХ, за покупки пр
7	0	В сентябре 2017г. В офисе банка, открыл вклад "Госстраховский". В марте 2019 поехал закрыть вклад. Оказалось, что офис закрыт. Позвонил по телефону в банк, где мне сообщили, что в г. Чита филиал банка закрыт и для закрытия в
8	0	В целом всё устраивает за исключением одного фактора который просто перечёркивает все плюсы. В 21 веке не могу "комфортно" внести деньги по ипотечному платежу чтоб не платить проценты. Один из крупнейших банков России а
9	0	Во первых, не отреагировали на мой отзыв сделали отписку. Навязали услугу, по высадке деревьев, когда просила в офисе не оформлять это, сотрудник сказала без этого НЕВОЗМОЖНО оформить кредит. А написали не смогли найти
10	0	Всем добрый день. Хочу поблагодарить руководство банка Тинькофф за лояльность к клиенту и оперативное решение проблемы. При совершении операций в системе банк-клиент, мною была допущена ошибка. В результате такой оши
11	0	Добрый день! Давно слышала про Почта банк, а лично оценить некоторые и достаточно весомые преимущества ,довелось недавно, благодаря случаю ,произошедшему с моим ребенком. Моя дочь , Алтана ,уже достаточно
12	0	Добрый день! Хочу написать отзыв о работе Русфинанс Банк. Дело было так, мой знакомый взял потребительский кредит и указал меня контактным лицом, у него начались просрочки и банк начал мне звонить. Я всегда беру трубки и с
13	0	Добрый день! Хочу выразить благодарность сотрудникам Альфа Банка (отделение по адресу: г. Москва, ул. Яна Райниса, д. 2, корп.1) Миронову Роману, Юн Светлане, Ефимовой Ксении за внимательное отношение, доброжелател
14	0	Добрый день. Я всегда знал, что сбербанк работает спустя рукава, как и все в этой стране. Но такого абсурда я не ожидал. 22.03.2019 года с меня списали 10 900 на основании судебного приказа!!! Который якобы вын
15	0	Задолбали звонить! Просят передать информацию или оказать финансовую помощь! Ни по хорошему, ни по плохому не понимают. Житья не дают! Знакомый указал меня как контактное лицо и кредит не платит. Угрожали даже мне ис
16	0	Мною был открыт вклад в МКБ много лет тому назад.Для повышения процентной ставки мне предоставили дебетовую карту.Согласно договору обслуживания прописано,что обслуживание данной карты бесплатное.Но на счете должно
17	0	Навязывание платных услуг Оформил кредит наличными в офисе Банка. Запрос был на 250 000 рублей. Девушка сказала, что одобрели, деньги придут на Ваш счёт. Быстро подписали документы. По возвращению домой, сумма оказ
18	0	Надеюсь мой отзыв будет полезен тем, кто планирует брать автокредит в Совкомбанке. 12.11.2018 оформил автокредит. Договор и приложения к нему читал очень внимательно. Огорчения №1 и
19	0	Наша организация сотрудничала с Росевробанком до момента его покупки Совкомбанком. После этого наш зарплатный проект автоматически перешел в Совкомбанк. Я честно пытался приспособиться, но через 3 месяца договор закры
20	0	Оформил в прошлом году кредит и карту халва понравилось обслуживание и внимание операторов которые со мной работали все вежливо и доступно объяснили про проценты переплату страховку ничего не утаили в других банках у к
21	0	Платёж по кредиту происходит 27 числа каждого месяца, заранее приходит уведомление по СМС, что удобно. НО! Сегодня лишь 24 число....воскресенье... выходной день- звонок с горячей линии в 09:00 об информировании того что к
22	0	Постоянно получаю банковские услуги в ДО Румянцево ПАО «Банк ВТБ» Прекрасно организована работа отделения. Приветливый, вежливый и внимательный персонал. Всегда чётко работают банкоматы. Нет больших оч
23	0	При одобрении кредита наличными, сотрудник предоставил договор для ознакомления, предоставил 2 графика платежей, один из которых на снятие наличных (сумма составляет по данному графику 17713.22) и второй график если он

Рис.4.Вид данных после их обработки в базе данных.

На этом этап обработки данных завершен. Приступаем к формированию словаря и работе с классификаторами векторного представления слов.

4.3.Чистка данных и создание словаря.

Чтобы классификатор векторного представления слов сработал максимально корректно необходимо удалить те части речи, которые не несут для нас никакой важной информации. Для этого будем использовать NLTK (Natural Language Toolkit) – пакет библиотек и программ для символьной и статистической обработки естественного языка, написанных на языке программирования Python.[32] Используя функцию tokenize, разделяем строку на подстроки и относит к определенным частям речи согласно кодовым названиям {'CONJ', 'ADV-PRO', 'CONJ', 'PART'}.

Теперь мы получаем словарь, состоящий из всех слов в наших отзывах. Получаем следующий формат записи:

Рис. 5.Словарь, полученный в Python

4.4. Алгоритмы преобразования текстовой информации в векторную.

Теперь необходимо обучить модель классификатора Word2Vec. Он принимает несколько параметров, которые влияют как на скорость обучения, так и на качество.

Word2Vec(size=100, min_count=1)

где size-это размерность векторов слов,min_count- игнорирует позволяет игнорировать все слова с меньшей частотой встречаемости. Вызов Word2Vec(sentences, iter=1) обычно проходит два прохода по всему тексту. То есть iter+1, можно задать большее количество проходов. Первый проход собирает слова и их частоты для построения внутренней древовидной структуры словаря. Разумное значение для min_count составляет от 0 до 100, в зависимости от размера вашего набора данных.Второй проход тренирует нейронную модель.

model.train(sentences, total_examples=model.corpus_count, epochs=model.epochs)

Обучение потоковое, то есть предложения могут быть генератором, считывающим входные данные из источника, без загрузки всего корпуса в RAM, epochs- количество итераций по корпусу.

А дальше мы сохраняем обученные векторы слов в экземпляре KeyedVectors в model.wv:

w2v = dict(zip(model.wv.index2word, model.wv.vectors))

```
{'поощрялась': array([-3.5042786e-03,  5.2169146e-04,  2.1134880e-03,  3.4255565e-03,
         2.9535536e-03, -2.5336174e-04,  3.5182014e-03,  4.7578118e-03,
    ..........................................................
        -2.3358944e-03,  3.9440244e-03,  7.6886819e-05, -3.4618229e-04],
       dtype=float32),
```

Рис.6.Векторное представление слов.

4.5. Применение градиентного бустинга.

И на полученных данных применяем XGBClassifier, в основе которого лежит алгоритм градиентного бустинга, описанного выше. Мы спускаемся по всем возможным алгоритмам и функциям, где каждая итерация отправляет нас к функции базовых алгоритмов. Здесь может заметить интересную корреляцию, представленную на рисунке 7 и 8:

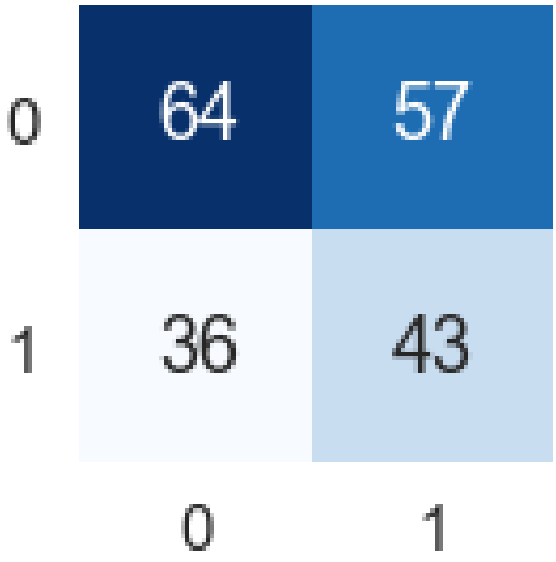


Рис. 7 Диаграмма

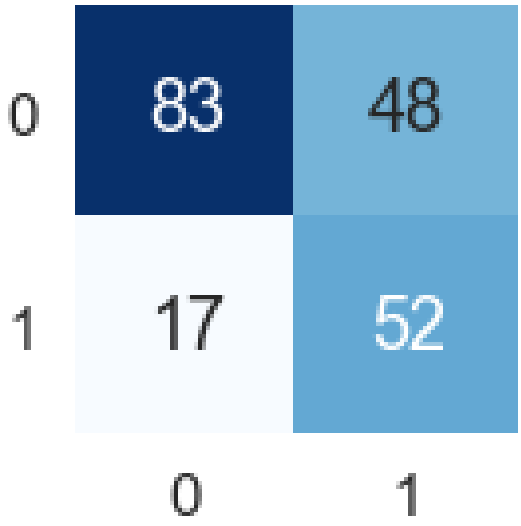


Рис. 8.Диаграмма

Данные диаграммы показывают точность, которую дает алгоритм справа, а слева то, в каких пропорциях были взяты положительные и отрицательные данные из исходной выборки.

Можно заметить, что в случае, когда данные взяты в близкой пропорции, точность определения отрицательных отзывов выше статистической. В тоже время на малом количестве положительных отзывов процент правильного определения максимален.

Чтобы сделать вывод о точности алгоритма, давайте построим ROC-кривую. Она отражает взаимную зависимость ложноположительных и истинно положительных результатов. Полное название таких кривых — «операционные характеристические кривые наблюдателя» - Receiver Operating Characteristic curve или, сокращенно, ROC-curve. Поэтому часто такие кривые называют ROC-кривыми, а выполняемые для их построения действия — ROC-анализом. При анализе ROC-кривых придерживаются следующего принципа: чем ближе к левому верхнему углу координатной сетки расположена кривая, тем выше информативность исследуемого метода диагностики или лучше качество системы отображения данных. Если кривая прилежит к диагонали (или совпадает с ней), то информативность метода ничтожна. Необходимо отметить, что в качестве истинно положительных решений может выступать критерий «чувствительность», а в качестве ложно положительных - критерии

«1 - специфичность». Принято считать, что коэффициент площади кривой, находящийся в интервале 0,9-1,0 следует рассматривать как показатель наивысшей информативности диагностического метода, 0,8-0,9 - хорошей, 0,7-0,8 - приемлемой, 0,6-0,7 - слабой, 0.5-0,6 - чрезвычайно слабой. Диагональ на приведенном рисунке отображает полное отсутствие информативности диагностического метода. Чтобы ее построить необходимо вычислить значение метрики accuracy (AUC) — доля правильных ответов алгоритма:

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FT+FN} \quad (35)$$

где TP- True Positive (положительные отзывы, верно определенные алгоритмом) ,TN- True Negative (негативные отзывы, верно определенные алгоритмом),FP- False Positive (положительные отзывы,неверно определенные алгоритмом), FN- False Negative (негативные отзывы, определенные алгоритмом).

Рассчитываем данный коэффициент для нашего случая:

$$\text{Accuracy}=\frac{260+116}{260+116+246+124}=0.5186 \quad (36)$$

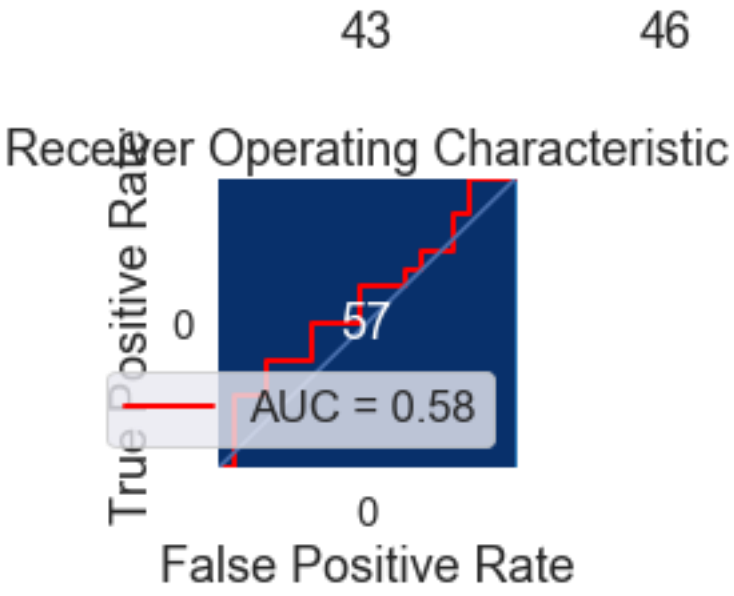


Рис.9.ROC- кривая классификатора Word2Vec

Как видим из графика точность алгоритма 57%, что не является достаточно высоким значением.

По аналогии применяем алгоритм TfidfVectorizer, основанный на методе TF-IDF анализа. Смысл метода в том, что если термин встречается с высокой частотой в одном тексте и практически не встречается во всех остальных текстах, то он имеет большую значимость для этого текста.

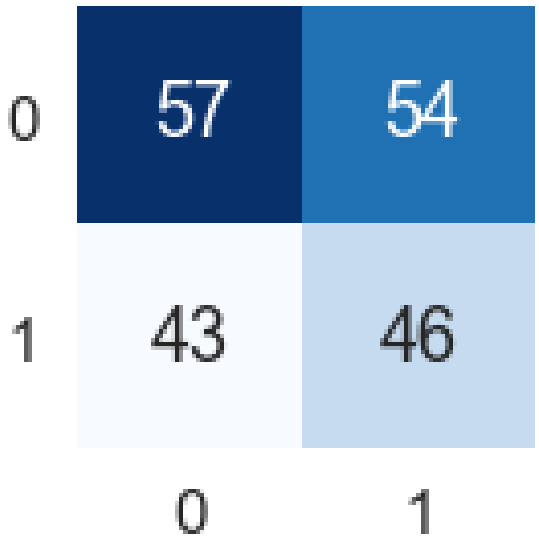


Рис.10.Диаграмма TF-IDF

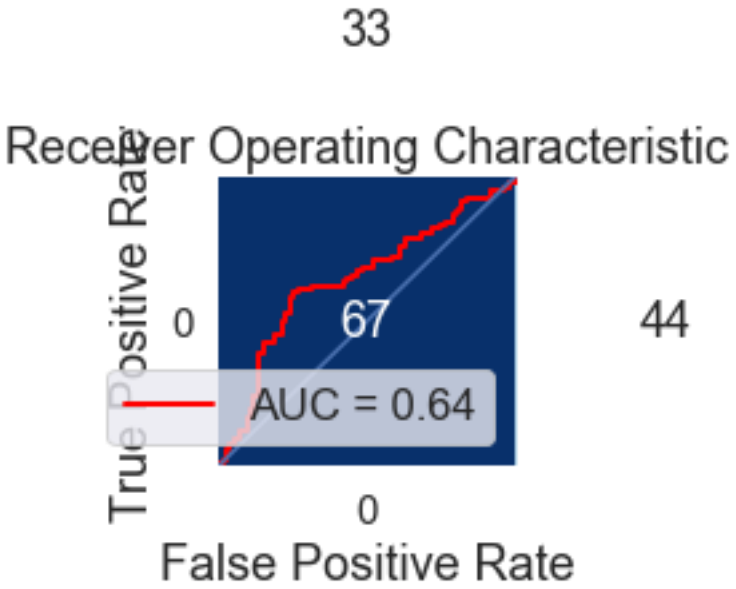


Рис.11 ROC-кривая TF-IDF классификатора.

Как видим точность алгоритма, основанного на TF-IDF анализе, больше, что связано с объемом входных данных. Метод Word2Vec лучше подходит для небольших текстов. Чем меньше объем входных данных тем быстрее получается обучить классифицирующую нейронную сеть.

Выводы

В рамках данной работы было разработано программное обеспечение для определения и анализа тональности текстов комментариев новостных и информационных источников по отношению к заданному объекту для последующей оптимизации процесса принятия решений. Для достижения поставленной цели были решены следующие задачи:

1) разработано программное обеспечение для выгрузки релевантных поставленной задаче данных с web-ресурсов, на основе которых обучался машинный классификатор;

2) выполнена предварительная обработка выгруженных текстовых данных;

3) каждый блок текстовых данных был представлен в виде векторов признаков, по которым он далее анализировался;

4) в качества алгоритм классификации и метод обучения классификатора был выбран метод градиентного бустинга, который был реализован с использованием библиотек и средств языка Python;

5) произведена валидация модели на неразмеченных текстовых данных;

6) были сравнены результаты работы системы при использовании методов векторизации Word2Vec и «мешок слов», в результате показатель точности работы классификатора при использовании Word2Vec составил 57%, а при использовании BagOfWords – 67%.

Заключение

В ходе данной работы был создан программный продукт, позволяющий проводить анализ тональности отзывов с новостных или информационных веб-ресурсов. Анализ проводился с использованием алгоритма машинного обучения с учителем, в качестве которого был выбран метод градиентного бустинга. Векторная модель текста строилась двумя разными способами: с помощью нейронной сети, лежащей в основе технологии Word2Vec, и с помощью алгоритма TF-IDF (BagOfWords). В итоге были получены следующие результаты по точности определения тональности текста: для модели Word2Vec – 57%, для TF-IDF – 67%. Полученные показатели наглядно демонстрирует, что в случае анализа «живого» языка, при большем количестве исходных данных лучшие результаты показывает классификатор TfidfVectorizer.

Также важно понимать, что на точность прогноза алгоритмов повлияло качество, загруженных с веб-ресурса данных. На сайте не ведется мониторинг соответствия текста отзыва оценке, поставленной автором. Были замечены отзывы, имеющие ярко выраженную позитивную окраску, но оценка стояла самая низкая.

Таким образом, можно сделать вывод, что метод машинного обучения с учителем всецело зависит от качества размеченных данных, а использование неочищенных текстовых источников ведет к получению низкой точности прогноза.

Результатом, полученным в 4 главе, является программный код, который проводит семантический анализ отзывов с веб-ресурса. Анализ был проведен двумя разными способами, мы сделали векторную модель слов с помощью нейронной сети и с помощью алгоритма TF-IDF, полученные результаты (57% и 67% соответственно) очевидно доказывают, что в нашей ситуации, когда мы работает с «живым» языком, большим количеством исходных данных лучше показывает результаты TfidfVectorizer классификатор. Также важно понимать,

что на точность прогноза алгоритмов повлияло качество, загруженных с веб-ресурса данных. На сайте не ведется мониторинг соответствия текста отзыва оценке, поставленной автором. Были замечены отзывы, имеющие ярко выраженную позитивную окраску, но оценка стояла самая низкая. Как видим, метод машинного обучения с учителем всецело зависит от размеченных данных и использование неочищенных источников ведет к получению низкой точности прогноза.

Список литературы:

1) Лысенко В. Д. Анализ тональности текста для прогнозирования цен на фондовом рынке // Молодой ученый. — 2018. — №22. — С. 420-423. — URL https://moluch.ru/archive/208/51025/
2) Андреева А. Н. Сентимент-анализ брендов в российской блогосфере как инструмент маркетинговых исследований. // "Бренд-менеджмент", #4, 2012 г.с.44-45
3) Kalyani Joshi1, Prof. Bharathi H. N.2, Prof. Jyothi Rao3, STOCK TREND PREDICTION USING NEWS SENTIMENT ANALYSIS // International Journal of Computer Science & Information Technology (IJCSIT) Vol 8, No 3, June 2016
4) Котельников Е. В. ,Клековкина М. В. Автоматический анализ тональности текстов на основе методов машинного обучения.// РОМИП-2011.
5) Нугуманова А.Б. ,Бессмертный И.А. Обогащение модели Bag of words семантическими связями для повышения качества классификации текстов предметной области.// журнал «Программные продукты и системы» № 2 за 2016 год. с. 89-99
6) Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 3111–3119, 2013.
7) Воронцов К.В. Простые классификаторы. Курс лекций. http://lpcs.math.msu.su/~pentus/opm/simple_classifiers.pdf
8) Е. А. Соколов Решающие деревья //ФКН ВШЭ https://www.hse.ru/mirror/pubs/share/215285956

9) К. В. Воронцов Лекции по методу опорных векторов. Курс лекций. http://www.ccas.ru/voron/download/SVM.pdf

10) А. Дьяконов «Введение в анализ данных и машинное обучение». [стр. 1-14]

11) Е.Р. Горяинова, Т.И. Слепнёва «Методы бинарной классификации объектов с номинальными показателями»// Журнал Новой экономической ассоциации №2 (14), С. 27–49

12) Morinaga, S., Yamanishi, K., Tateishi, K., Fukushima, T. Mining Product Reputations on the Web // In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). 2002. P. 341-349.

13) Pang B., Lee L., Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques // In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2002). 2002.

14) Liu H. MontyLingua: An end-to-end natural language processor with common sense, 2004.Available at <http://web.media.mit.edu/hugo/montylingua> (accessed 1 February 2005).

15) Turney, P. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews // ACL‘02. 2002.

16) Joachims T. Making large-scale SVM learning practical // In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), The MIT Press, 1999.

17) Pang B., Lee L. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval. 2. No. 1–2 (2008). P. 1–135.

18) Pang B., Lee L. Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the Conference on Empirical Methods in Natural.Language Processing (EMNLP). Philadelphia. 2002.P. 79–86.

19) Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышински Э.С., Лукашевич Н.В., Сапин А.С. Автоматическая обработка текстов на естественном языке и анализ данных// Изд-во НИУ ВШЭ, 2017. — с.17-18

20) И.В. Бондарева, Д.Г. Лагерев Исследование методов векторного представления текстовой информации для решения задачи анализа тональности//Всероссийская конференция "Информационные технологии интеллектуальной поддержки принятия решений",2018,с.4

21) Pennington, J., Socher R., Manning C.D. Global Vectors for Word Representation. // Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing, P. 1532–1543.

22) Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K., Harshman R. Indexing by Latent Semantic Analysis // The American Society for Information Science. 1990. Vol. 41. P. 391-407

23) Joachims T. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. Proceedings of 14th International Conference on Machine Learning, Nashville, TN, 1997, pp. 143–151

24) Masand B., Linoff G., Waltz D. Classifying news stories using memory-based reasoning. Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, 1992, pp. 59–65.

25) Lewis D. D. Naive (Bayes) at forty: The independence assumption in information retrieval. Proceedings of 10th European Conference on Machine Learning, Chemnitz, Germany, 1998, pp. 4–15.

26) Joachims T. Text categorization with support vector machines: learning with many relevant features. Proceedings of 10th European Conference on Machine Learning, Chemnitz, Germany, 1998, pp. 137–142.

27) Wang C., Cao L., Zhou B., “Medical Synonym Extraction with Concept Space Models”, 2015, https://arxiv.org/pdf/1506.00528.pdf.

28) Stanford Demo for predicting sentiment of movies reviews. http://nlp.stanford.edu/sentiment/

29) Sentiment140 – sentiment analysis platform. http://www.sentiment140.com/

30) 30dp – opinion search platform. https://www.30db.com/

31) ВААЛ – система контекст-анализа текста. http://www.vaal.ru/

32) https://ru.wikipedia.org/wiki/Natural_Language_Toolkit