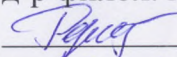


Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
САЕ «Институт человека цифровой эпохи»
Автономная магистерская программа «Компьютерная лингвистика»

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

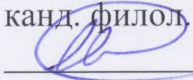
Руководитель ООП
д-р филол. наук, профессор
 3. И. Резанова
« 17 » июня 2019 г.

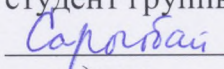
МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

АВТОМАТИЧЕСКАЯ БИНАРНАЯ КЛАССИФИКАЦИЯ ОБЗОРОВ НА
УНИВЕРСИТЕТЫ РОССИИ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ
МАШИННОГО ОБУЧЕНИЯ

по основной образовательной программе подготовки магистров
направление подготовки 45.04.03 – Фундаментальная и прикладная
лингвистика

Сарыгбай Долана Владимировна

Научный руководитель,
канд. филол. наук, доцент
 К.С. Шильяев
подпись
« 17 » июня 2019 г.

Автор работы
студент группы № 13785
 Д.В. Сарыгбай
подпись

Томск-2019

Оглавление

Введение	3
Глава I. Сентимент-анализ в контексте науки о данных и обработки естественного языка	7
1.1 Сентимент-анализ в контексте обработки естественного языка	9
1.1.1. Корпус для сентимент-анализа	12
1.1.2. Уровни сентимент-анализа	13
1.1.3. Особенности сентимент-анализа на уровне документа	15
1.1.4. Семантические тезаурусы для сентимент-анализа	17
1.1.5. Лингвистические основания сентимент-анализа	21
1.2 Сентимент-анализ и наука о данных	24
1.2.1 Подходы к классификации текста	25
1.2.2. Подходы к машинному обучению	27
1.2.3. Глубокое обучение	31
1.2.4. Алгоритмы машинного обучения с учителем	32
1.2.5 Предварительная обработка текста	37
1.2.6 Перекрестная проверка для данных	42
Глава II Описание скрипта алгоритмов машинного обучения для классификации отзывов на университеты	44
Выводы по главе II	57
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ	61

Введение

С увеличением количества информации в интернете и с развитием социальных сетей у пользователей появилась возможность обмениваться информацией. Чаще всего эта информация носит субъективный характер, то есть содержит мнение автора. К примеру, в социальной сети Twitter может авторизоваться любой человек и написать текстовое сообщение на любую тему. В социальных сетях Facebook, Вконтакте, Instagram пользователь может выложить фотографию и под фотографией написать пост на любую тему. Кроме социальных сетей интернет дает возможность делиться опытом, впечатлениями, давать советы, вести блоги. Для этого существуют специальные платформы как LiveJournal, Кинопоиск, TripAdvisor, Amazon, Booking, Livelib.

Объем неструктурированной информации увеличивается в интернет-пространстве. Её обработка требует значительных усилий и затрат. Для ее обработки используют различные инструменты больших данных. «Большие данные (Big Data) — общее название для структурированных и неструктурированных данных огромных объемов, которые эффективно обрабатываются с помощью масштабируемых программных инструментов. Такие инструменты появились в конце 2000 годов и стали альтернативой традиционным базам данных и решениям Business Intelligence» [Большие данные... 2019]. Сентимент-анализ в качестве одного из инструментов больших данных анализирует, классифицирует и предсказывает на основе данных информацию [Pagolu 2016, 1345-1350]. Значимым ресурсом для сентимент-анализа становится субъективная информация, которая содержит положительные или негативные мнения пользователей. Сентимент-анализ как инструмент аналитики применяется, во-первых, в сфере бизнеса, а именно в сфере продаж (marketing): знание мнения о продукте дает построить правильную стратегию продаж, а также проводить мониторинг бренда на рынке. Например, аналитика позволяет узнать, среди какой возрастной группы

популярен продукт и на кого ориентироваться, их пол, в какой стране больший спрос на продукт [Introduction to Sentiment... 2019]. Во-вторых, инструменты сентимент-анализа, применяются в политических исследованиях. В качестве популярного примера приводят выборы в Президенты США 2016 года, где кандидатами были Дональд Трамп и Хиллари Клинтон. Пользователи социальной сети Twitter писали огромное количество постов на тему выборов, так как выборы – это одна из широко обсуждаемых новостей. На основе текстовой информации от пользователей аналитики могли предсказать, у кого больше шансов на победу [Ramteke 2016, 1-5]. В-третьих, приложения сентимент-анализа применяются в социологических исследованиях. На основе записей пользователей в социальных сетях можно сделать выводы об отношении людей к аспектам общества: религии, образованию, здравоохранению, местным властям, частному предпринимательству и т. д. Тексты в постах социальных сетей имеют личный, искренний характер, потому что они, как правило, непосредственно от самих авторов, поэтому результаты социологических исследований чаще всего достоверны [Sentiment Analysis... 2019].

На основе вышесказанного можно сделать вывод, что сентимент-анализ – это современный инструмент аналитики, который применяется в маркетинге, в политических и социальных исследованиях. Это обосновывает **актуальность** темы дипломной работы.

Выявление мнений является основной задачей анализа тональности текстов. В данной дипломной работе в качестве обучающей и тестовой выборки текстов были отобраны русскоязычные отзывы на университеты с сайта <https://www.iagora.com/studies/Russia/>. Сайт предназначен, чтобы студенты могли оставить свои мнения о высшем учебном заведении. С сайта были собраны тексты, чтобы автоматически классифицировать отзывы на университеты на положительные или негативные классы. Мы рассматриваем популярную задачу бинарной классификации отзывов, но **новизна** данной

работы заключается в его обучающей и тестовой выборке с сайта <https://www.iagora.com/studies/Russia/>.

Следовательно, **целью** дипломной работы выступает создание автоматического классификатора с применением методов машинного обучения на основе русскоязычных отзывов на университеты. Для достижения цели поставлены следующие **задачи**:

- 1) провести обзор литературы по обработке естественного языка, сентимент-анализу, машинному обучению;
- 2) собрать обучающие тексты и тестовые тексты для классификации на негативные и положительные отзывы;
- 3) написать код классификатора на языке программирования Python с применением методов машинного обучения.

Объектом дипломной работы является бинарный автоматический классификатор тональности текстов. **Предметом работы** выступает автоматический классификатор текстов с применением методов машинного обучения на положительные и негативные отзывы на университеты.

Теоретическая значимость работы заключается в применении результатов исследования для создания не только бинарного классификатора, но и для создания классификатора по базовым эмоциям или для решения задач обработки естественного языка: автореферирование отзывов, распознавание именованных сущностей, создание корпуса отзывов на университеты и т. д. **Практическая значимость** состоит в учебной отработке классической задачи классификации отзывов на университеты. В дальнейшем к отработанным алгоритмам можно будет применять обучающую и тестовую выборки с различной тематикой. Например, это могут быть отзывы на университет, на товары и услуги, отзывы путешественников и т. д.

В качестве **материала** дипломной работы использованы положительные и негативные русскоязычные отзывы с сайта <https://www.iagora.com/studies/Russia/>.

Структура работы. Дипломная работа состоит из введения, теоретической и практической глав, заключения и списка использованных источников и литературы.

Глава I. Сентимент-анализ в контексте науки о данных и обработки естественного языка

В русскоязычных работах сентимент-анализ чаще всего обозначают как «анализ тональности текстов», тогда как в англоязычной среде – sentiment analysis. В англоязычных статьях, книгах объектом исследования становится эмоция, чувство, мнение (sentiment), то есть психологический аспект человека. Также термин sentiment analysis синонимичен термину opinion mining («извлечение мнения»). Это значит, что в англоязычных работах имеет значение субъективность слова – «содержит ли данный текст мнение, а если содержит, то положительное, негативное или нейтральное» [Pang 2008, 2-17].

В русскоязычной литературе встречается слово «тональность» [Большакова 2017, 125-177, Батура 2016, 157-165]. Под тональностью имеется в виду полярность слова. Полярность выражается в виде положительного или отрицательного числа. Также полярность может быть нулевой. Таким образом, задача сентимент-анализа в русскоязычном контексте сводится к классификации текста, предложения или слов по полярностям, но без упоминания, содержится ли мнение автора, то есть субъективность текста.

Задача сентимент-анализа состоит в классификации текстов по положительному и негативному классам. Классами могут быть эмоция или полярность. Задача решается в рамках компьютерной лингвистики и/или науки о данных. В рамках компьютерной лингвистики (обработка естественного языка) лингвист вручную размечает по полярностям тексты или слова и затем классифицирует. В рамках науки о данных решение задачи сводится к автоматизации, а именно к применению алгоритмов машинного обучения, программирования, статистики [Liu 2015, 47-49].

В первой главе данной дипломной работы раскрывается теоретическая основа сентимент-анализа в контексте компьютерной лингвистики. Во второй главе рассматривается классический пример классификации текстов (отзывы

на университеты): на языке программирования Python с применением алгоритмов машинного обучения для их классификации.

1.1 Сентимент-анализ в контексте обработки естественного языка

Задача сентимент-анализа в качестве классификации текста есть одна из задач обработки естественного языка. Обработка естественного языка (natural language processing, NLP) – междисциплинарная область на стыке лингвистики и компьютерных наук, объектом изучения которой является текст и/или речь и способы его/ее автоматической обработки [Jurafsky 2009, 1-2].

Обработка естественного языка (компьютерная лингвистика) зародилась в 50-ые годы XX-ого века вместе с развитием ЭВМ. Первой задачей обработки естественного языка был машинный перевод. В 1954 году IBM совместно с Джорджтаунским университетом провёл эксперимент: перевести с русского на английский более 60 предложений. Эксперимент прошел удачно, но так как предложения для перевода были заранее подготовлены и были достаточно просты, то последующие попытки перевода не были столь же удачными, и проект далее не получил финансовой поддержки. Таким образом, развитие машинного перевода и обработки естественного языка на несколько лет остановилось [Нелюбин 2006].

Вторым значимым событием обработки естественного языка считается эксперимент машины Тьюринга в 50-ые годы XX-века. Сейчас данный эксперимент назвали бы задачей создания диалоговых систем. Эксперимент заключался в том, что человек должен был понять, с кем он переписывается: с живым человеком или с машиной. Задачей машины было выводить настолько естественные предложения, что собеседник подумал, что перед ним человек, а не машина. С 1990 года проводится международный ежегодный конкурс Лёбнера, где участники конкурса представляют свои так называемые чат-боты, которые соревнуются в прохождении теста Тьюринга. Побеждает тот конкурсант, чей чат-бот ведет диалог наиболее естественно [Тест Тьюринга...2019].

Помимо сентимент-анализа в область проблем естественного языка включают машинный перевод, распознавание речи, распознавание сущностей,

оптимизация поисковых систем, диалоговых систем и многое другое. Таким образом, можно заключить, что основная цель обработки естественного языка – это вычленять определенную информацию по заданным параметрам из неструктурированных данных. В случае с sentiment-анализом в задачу NLP входит вычленение 2 типов информации из текста: во-первых, субъективен или объективен текст, и, во-вторых, если текст субъективен, то какова его полярность: негативная, положительная или нейтральная [Liu 2015, 16-17].

Интерес к sentiment-анализу появился с возрастанием количества субъективного текста в интернет-пространстве. Субъективная информация появляется в местах общения людей: форумы, социальные сети, рекомендательные сайты [Liu 2015, 47-48].

Также возрастает интерес к приложениям и инструментам sentiment-анализа со стороны бизнес-сектора. Компании, которые специализируются на аналитике, добавляют sentiment-анализ в качестве одной из их услуг. К таким компаниям можно причислить SAS, Cogito, Concordus, USA Today.

Дисциплина обработки естественного языка находится на стыке гуманитарных и технических наук. На сегодняшний день наблюдается тенденция перехода в сторону науки о данных (data science). Для лингвиста без знания программирования, машинного обучения, статистики становится трудновыполнимой классическая задача классификации текстов. На практике задача классификации текстов становится задачей специалиста по данным (data scientist) [Hart 2013, 1-2].

Если верно написать на одном из языков программирования алгоритмы классификаторов машинного обучения, то задача классификации текстов сводится к минимальным ресурсным затратам. Специалисты в области компьютерных наук и специалисты по данным становятся основными фигурами в решении задач в области компьютерной лингвистики. Задача лингвиста сводится к созданию корпуса текстов для sentiment-анализа,

подбору лексики для обучающей и тестовой выборки, а также к составлению теоретической базы для сентимент-анализа.

Теоретическая база сентимент-анализа является предметом исследования лингвистов. Лингвисты выделяют несколько аспектов сентимент-анализа [Liu 2015, 9-10]:

1. Создание корпуса, создание словарей тональности. Подготовка обучающей и тренировочной выборки.
2. Выбор уровня: документ, предложение, аспекты
3. Субъективность текста: субъективен ли текст или объективен
4. Полярность: положительная, негативная, нейтральная
5. Предварительная обработка (pre-processing)
6. NLP основы: n-граммы, tf-idf, bag-of-words, feature extraction.

1.1.1. Корпус для sentiment-анализа

Корпус текстов для sentiment-анализа должен содержать положительное или негативное мнение по поводу какого-либо объекта, то есть текст должен быть субъективным. Обычно язык интернет-общения достаточно субъективен и эмоционален. Важными источниками материала для sentiment-анализа могут быть социальные сети (ВКонтакте, Facebook, Twitter), рекомендательные сайты (TripAdvisor, Internet Movie Database, Amazon). Субъективный текст можно найти на форумах, где обсуждаются социальные и политические проблемы.

Язык корпуса для sentiment-анализа отклоняется от литературного языка: он содержит сленг, более неформальный, с опечатками и с ошибками. В разных медиа-платформах стиль языка отличается, поэтому обучающая и тестовая выборка должны составляться из одного текстового материала для точности классификации [Hart 2013, 7-8].

1.1.2. Уровни сентимент-анализа

Бинг Лиу [Liu 2015, 9-10] выделяет 3 уровня сентимент-анализа: документа, предложения, аспекта. Уровень документа предполагает, что целый текст (документ) содержит одно мнение и одну полярность, хотя в действительности документ может содержать несколько мнений от разных лиц и несколько полярностей: например, «моему другу очень нравится Хонор, хотя меня раздражает его камера». В приведенном примере мы видим два субъекта (мой друг и я) и 2 полярности (положительная, негативная). Таким образом, уровень документа не отражает точное мнение об объекте. Особенно это важно, если пользователь ищет информацию о конкретном объекте (например, камера телефона). Тем не менее уровень документа считается очень популярным для отработки навыков классификации текстов по полярностям, так как это самый базовый, и соответственно, самый легкий уровень.

Следующим уровнем сентимент-анализа выступает уровень предложения. Задача на этом уровне – определить, выражает ли каждое предложение мнение. Данный уровень различает объективные предложения, выражающие какой-либо факт, и субъективные предложения, выражающие мнения. Таким образом, мы видим две ступени: во-первых, распознать, является ли предложение субъективным, и, во-вторых, узнать его полярность.

Третьим, наиболее детальным уровнем является уровень аспекта. На данном уровне нужно рассматривать полярность ключевых слов (features). В зависимости от полярности слова устанавливается положительное или отрицательное мнение. Далее нужно определить к чему относится мнение: к сущности (какой-либо объект) или к аспекту сущности. Например, сущностью может быть камера, а аспектом выступает батарея камеры.

В данной дипломной работе рассматривается наиболее базовый и наиболее достижимый уровень – уровень документа, так как практическая

значимость работы состоит в учебной отработке базовой задачи классификации отзывов на университеты.

1.1.3. Особенности sentiment-анализа на уровне документа

Сентимент-анализ на уровне документа [Liu 2015, 47-69]. – это наиболее широко изученная тема в области СА, особенно в ранние годы его исследования. Задачей СА на уровне документа является классифицировать документы по положительным и негативным полярностям (эмоциям, сентиментам). Задача классификации относится к уровню документа, потому что документ обрабатывается целиком, то есть не рассматриваются сущности или аспекты, а также не определяются полярности по отношению к ним.

Документы для классификации должны быть субъективными, содержащими полярное мнение. Мнение должно принадлежать только одному автору и содержать одну тему.

Мы можем видеть, что проблема уровня документа достаточно ограничена, так как в документе может идти речь о нескольких объектах, и к каждому объекту может быть различное эмоциональное отношение. Автор может быть позитивно настроен к одной сущности и негативно – к другой. В таком случае задача СА уровня документа становится менее значимой, потому что не имеет смысла определять одним сентиментом целый документ. Подобным образом задача не имеет значения, если несколько авторов выражают мнение в одном документе, так как их мнения и полярности мнений могут отличаться друг от друга.

Предположение уровня документа СА о том, что документ содержит одну полярность от одного автора, подходит для онлайн-отзывов на товары и услуги, так как в отзыве обычно идёт речь об одном товаре или услуге от одного автора. Однако это предположение может быть неверным для форумных дискуссий или для постов в блогах, потому что в таких записях автор может выражать мнение по нескольким темам. Поэтому многие исследователи используют отзывы для решения задачи классификации или регрессии.

Для онлайн-отзывов не нужна классификация по полярностям, потому что почти каждый отзыв сопровождается рейтинговыми звездами для пользователей. На практике классификация по полярностям нужна для дискуссий на форумах и постах в блогах, чтобы определить мнения людей.

Сентимент-анализ на уровне документа – одна из базовых задач, которая решается как задача текстовой классификации, где в качестве классов – полярности. Следовательно, любой алгоритм машинного обучения с учителем может быть непосредственно применен для решения задачи классификации. Также часто бывает, что признаки (features), выделенные для классификации по полярностям, совпадают с признаками для текстовой классификации.

Существуют две популярные постановки задачи, основанные на качественном типе сентимента на уровне документа. Если сентимент рассматривается как категориальная величина, например, классы позитивный и негативный, то это задача классификации. Если сентимент рассматривается как числовая величина или порядковый счет в рейтинге, например, в виде звезд от 1 до 5 то задача СА становится задачей регрессии.

1.1.4. Семантические тезаурусы для сентимент-анализа

Существует ряд тезаурусов, специально размеченных с учётом эмоциональной составляющей. Такие словари, описанные далее, необходимы компьютерным программам при анализе тональности текста.

WordNet-Affect

Примером для разработки WordNet-Affect послужило многоязычное расширение WordNet, названное WordNet Domain [Carlo Strapparava 2004, 1083-1086]. В расширении WordNet Domain каждому синсету приписано не менее одной пометы предметной области (англ. «domain label»), например: спорт, политика, медицина. Всего в иерархически организованную структуру было включено около двухсот предметных помет [Bernardo Magnini 2000].

WordNet-Affect – это семантический тезаурус, в котором понятия, связанные с эмоциями («эмоциональные концепты», англ. «affective concepts»), представлены с помощью слов, обладающих эмоциональной составляющей («эмоциональные слова», англ. «affective words»). WordNet-Affect состоит из такого подмножества синсетов WordNet, где каждый синсет, соответствующий «эмоциональному концепту», может быть представлен с помощью «эмоциональных слов».

Таким образом, WordNet-Affect был создан на основе WordNet для английского языка (также существуют версии WordNet-Affect и для других языков [Victoria Bobicev 2010]) путём выбора и отнесения наборов синонимов (синсетов) к различным эмоциональным понятиям. В частности, синсеты глаголов, существительных, прилагательных, наречий, которые представляют собой описание эмоций, были вручную размечены с помощью специальных эмоциональных меток (affective labels, A-labels). Эти эмоциональные метки характеризуют различные состояния, выражающие настроения,

эмоциональные отклики или ситуации, которые вызывают эмоции [Wordnet-Affect 2009].

Также в WordNet-Affect используются дополнительные эмоциональные метки для того, чтобы разделять синсеты в соответствии с их эмоциональной валентностью. Для этого определяются четыре дополнительные эмоциональные метки: позитивная, негативная, неоднозначная и нейтральная [Wordnet-Affect 2009]. Первая соответствует положительным эмоциям, которые определяются как эмоциональные состояния, характеризующиеся наличием положительных гедонистических сигналов (или удовольствия). Она включает в себя такие синсеты как радость#1 или увлечение#1. Аналогично негативная метка идентифицирует негативные эмоции, характеризующиеся отрицательными гедонистическими сигналами (или болью), например, гнев#1 или печаль#1. Синсеты, представляющие эмоциональные состояния, валентность которых зависит от семантического контекста (например, удивление#1) помечаются как неоднозначные. Наконец, синсеты, определяющие психологические состояния и всегда рассматривающиеся как неоднозначные, но при этом не характеризующиеся валентностью, являются нейтральными [Wordnet-Affect 2009].

Синсеты, помеченные эмоциональными метками, дополнительно переразмечаются шестью эмоциональными категориями: радость, страх, гнев, печаль, отвращение, удивление. Таким образом, физическая структура WordNet-Affect состоит из шести файлов: anger.txt, disgust.txt, fear.txt, joy.txt, sadness.txt, surprise.txt, где каждый файл представляет собой описание какой-либо категории [Румынский и русский WordNet-Affect...2019]. На данный момент WordNet-Affect содержит 2874 синсетов и 4787 слов.

SentiWordNet

Данный лексический семантический тезаурус является результатом процесса автоматического аннотирования каждого WordNet синсета (набора синонимов) в соответствии с его степенью позитивности, негативности и объективности [SentiWordNet... 2019]. Таким образом, каждому синонимическому ряду из WordNet присваивается три численных оценки, где каждая из этих оценок соответственно определяет объективную, позитивную или негативную составляющую синсета. Каждая из этих оценок принимает значения в интервале от 0 до 1, и в сумме они дают 1 (единицу), то есть каждая из этих оценок может иметь ненулевое значение. Термы, которые могут иметь различные значения, могут иметь и различные значения оценок [Stefano Baccianella 2010, 2200-2204].

SenticNet

SenticNet представляет собой еще один семантический тезаурус для работы с наборами эмоциональных понятий. SenticNet является проектом, запущенным в медиа-лаборатории Массачусетского технологического института в 2010 году [About SenticNet... 2019]. С тех пор проект SenticNet получил дальнейшее развитие и применяется для проектирования интеллектуальных приложений, предназначенных для анализа эмоциональной составляющей текста и охватывающих спектр задач от data mining до организации взаимодействия человека с компьютером. Главным назначением SenticNet является упрощение процедуры машинного распознавания концептуальной и эмоциональной информации, передаваемой с помощью естественного языка [About SenticNet... 2019]. Если сравнить другие лексические тезаурусы, такие как SentiWordNet и WordNet-Affect с SenticNet, то их главным различием будет то, что SentiWordNet и WordNet-Affect обеспечивают связывание слов и эмоциональных понятий на синтаксическом

уровне, не позволяя выявлять смысловую составляющую, например, «достижение цели», «нехорошее чувство», «отпраздновать особый случай», «потерять самообладание» или «быть на седьмом небе от счастья», в то время как SenticNet связывает понятия на семантическом уровне [Erik Cambria... 2019].

1.1.5. Лингвистические основания сентимент-анализа

Полярность и тональность

Так, согласно определению, взятому из Логического словаря справочника Н.И. Кондакова, “**полярность**” характеризуется присутствием двух противоположных полюсов в отдельно взятом объекте; к тому же, эти полюса должны представлять четко-выраженные оппозиции [Кондаков 1975].

Тональность – это эмоциональное отношение автора высказывания к некоторому объекту (объекту реального мира, событию, процессу или их свойствам/атрибутам), выраженное в тексте. Эмоциональная составляющая, выраженная на уровне лексемы или коммуникативного фрагмента, называется лексической тональностью (или лексическим сентиментом). Тональность всего текста в целом можно определить как функцию (в простейшем случае сумму) лексических тональностей составляющих его единиц и правил их сочетания [Liu B. 2013].

Категория тональности неразрывно связана с эмоциональным тоном произведения, но находится в отношении к нему как частное к общему. Эмоциональный тон включает в себя весь спектр эмоций в тексте, а категория тональности связана с эмоциональными доминантами, выраженными в тексте в виде различных языковых и речевых средств. В рамках одного текста могут взаимодействовать несколько тональностей, которые вместе и будут создавать общий эмоциональный тон произведения [Liu B. 2013].

Субъективность и объективность

Другое исследовательское направление – это идентификация субъективности/объективности. Эта задача обычно определяется как отнесение данного текста в один из двух классов: субъективный или объективный. Эта проблема иногда может быть более сложной, чем классификация полярности: субъективность слов и фраз

может зависеть от их контекста, а объективный документ может содержать в себе субъективные предложения (например, новостная статья, цитирующая мнения людей). Более того, как упоминал Су [Fangzhong Su 2008], результаты в большей степени зависят от определения субъективности, употребляющейся в рамках аннотации текстов. Как бы то ни было, Панг [Bo Pang 2004, 271-278] показал, что удаление объективных предложений из документа перед классификацией полярности помогло повысить точность результатов.

Денотат и коннотат

Денотат – это объект мысли, отражающий предмет или класс предметов действительности и обозначаемый языковым выражением или языковой единицей – именем. Денотат представляет собой предметное значение объекта, устанавливаемое в процессе его обозначения в имени и тем самым образующее его понятийное содержание. Указывая на объект его именем, денотат в то же время представляет его как объект мысли; при этом денотат – это именно представление об объекте, а не сам объект. Денотативный процесс есть отношение имени к предмету безотносительно к его свойствам – «означение», в отличие от коннотативного процесса – «соозначения», сопровождающегося не только указанием на предмет, но и обозначением его отличительных свойств [Апресян Ю. 1974].

Коннотация – сопутствующее значение языковой единицы. Коннотация включает дополнительные семантические или стилистические элементы, устойчиво связанные с основным значением в сознании носителей языка. Коннотация предназначена для выражения эмоциональных или оценочных оттенков высказывания и отображает культурные традиции общества. Коннотации представляют собой разновидность прагматической информации, отражающей не сами предметы и явления, а определённое отношение к ним [Апресян Ю. 1995].

Например, если человек пишет: «Мы хотим быстрых решений. А он же просто черепаха», то автор поста, скорее всего, имел в виду медлительного человека, а не представителя класса пресмыкающихся. В этом примере у слова «черепаха» денотативное значение – это «представители рептилий, тело которых покрыто панцирем». А коннотативное значение — это «медлительный человек». Конечно, для человека намного легче распознать коннотативное значение слова или фразы, чем для машины. В этом и состоит задача: научить машину понимать, где и когда употребляется коннотативное значение слова.

Или, например, предложения «Подскажите, пожалуйста, где можно купить чудесный / замечательный / прекрасный / красивый подарок на Новый год». В этом предложении употреблены прилагательные «чудесный / замечательный / прекрасный / красивый», но все они имеют лишь денотативное значение «хороший». Здесь какого-либо дополнительного коннотативного значения нет. Эти прилагательные лишь описывают хороший подарок от лица автора, так как он выбирает употребление какой-либо лексической единицы.

1.2 Сентимент-анализ и наука о данных

Специалист по данным решает задачу сентимент-анализа со сбора и чистки данных. В зависимости от задачи собираются конкретные данные. Например, задача может быть представлена в классификации мнений о новой модели айфонов или о кандидате на выборах Президента. Ресурсами для сбора данных могут быть сайты, где люди оставляют свои негативные или положительные мнения. Это могут быть Amazon, Rotten Tomatoes, Trip Advisor, социальные сети (Twitter, Instagram, Facebook), то есть данные представляют собой размеченные тексты.

Для сбора данных с сайтов специалист (data scientist) применяет инструмент crawler и/или scraper. Задача этих инструментов заключается в сканировании определенных сайтов в интернете для получения заданных данных. После сбора необходима чистка данных. Этот процесс называется pre-processing (предварительная обработка данных). Процесс включает в себя удаление стоп-слов, токенизацию (tokenizing), стематизацию (stemming), частеречную разметку (part-of-speech tagging).

Собранные данные нужно правильно классифицировать. Для бинарной классификации сентимент-анализа применяются инструменты машинного обучения. Машинное обучение – это метод решения задач, опирающийся на обучающие данные (training data), для прогнозирования, классификации или кластеризации объектов. Теоретической основой машинного обучения является статистический анализ. После сбора данных нужно обозначить, какой тип машинного обучения нужен для классификации текстов. Есть два типа машинного обучения, применяемых для решения задачи сентимент-анализа: обучение с учителем (supervised), обучение без учителя (unsupervised). Помимо машинного обучения существует метод со словарем [Text Mining...2019].

1.2.1 Подходы к классификации текста

1) **Машинное обучение с учителем** – наиболее распространенный тип машинного обучения, применяемый в области обработки естественного языка, но для которого необходима обширная размеченная выборка. Часто выборка составляется вручную. Обучение с учителем основано на наличии известного набора классов, примеры каждого из которых содержится в корпусе. В обучающей выборке каждый документ помечен одним из классов. Далее система строит модель классификации, основанную на извлеченных признаках (ключевых словах, features). Алгоритмы машинного обучения с учителем включают наивный байесовский классификатор, дерево решений, логистическую регрессию [Bird 2009, 221-257].

2) **Машинное обучение без учителя** – другой тип машинного обучения без применения размеченной выборки. Это значит, что обучение без учителя может обработать «сырой» текст. Но данные должны, как и в обучении с учителем, быть подготовлены (pre-processing), и признаки (features) должны быть извлечены. Обучение без учителя применимо, если определяющие характеристики наборов для классификации неизвестны. Предполагается, что машина сама должна обучиться, найти закономерности в неразмеченных данных.

Несмотря на преимущество в виде отсутствия необходимости размеченных данных для обучения, машинное обучение без учителя не часто используется в задаче сентимент-анализа, так как высока вероятность ошибок. Но стоит отметить, что данный метод широко применяется в других задачах обработки естественного языка и искусственного интеллекта [Mueller 2017, 133-134].

3) **Классификация с использованием словарей тональности.** В качестве выборки используются не размеченные тексты, а словарь, где у

каждого слова своя полярность. Для конкретной категории объектов (ноутбуки, фильмы, фирмы) составляются свои словари, где у каждого слова своя полярность. Метод считается ресурсозатратным, поэтому редко применяется на практике [Батура 2016, 157-165].

1.2.2. Подходы к машинному обучению

Эти подходы требуют заранее размеченный корпус (позитивный, негативный, нейтральный) [Zhang et al. 2007, 500-274]. В основном используются такие признаки (features) как: слова, биграммы, триграммы, частеречная разметка и полярность. Применяются много техник обучения с учителем, но двое из которых позволяют получать лучшие результаты: метод опорных векторов (SVM) и наивные байесовские классификаторы [Sindhu 2013, Wang 2012, Nilesh 2012].

В Пак и Парубек [Pak, A., Paroubek, P. 2011] разработали новое представление субграфа из дерева синтаксической зависимости. Они представляют собой текст как коллекцию субграфов, где узлы – слова (или классы слов), и дуги – синтаксические зависимости между ними. Такое представление позволяет избежать потери информации как с моделью мешка слов (bag of words model), который базируется только на коллекции n-граммов слов. Подходы, основанные на n-граммах, не могут правильно распознать совокупность выражений сентимента. Авторы применяют постепенный парсер (Incremental Parser XIP), чтобы построить дерево зависимостей. Они протестировали модель на выборке французских отзывов на видеоигры по анализу мнений (opinion mining). Им удалось продемонстрировать SVM-классификатор, использующий признаки, (features) построенные из субграфов деревьев зависимостей, дают лучшие результаты, чем традиционные системы, основанные на униграммах.

Рафрари и соавторы [Rafrafi, A. et 2011] предлагают использовать нейронные сети для обучения эффективной модели сентимент-классификации. Они сравнили свою работу с SVM-моделью, используя многотематический корпус Амазона. Результаты эксперимента показывают одинаковую работу.

В [Zhang, L. et al. 2014] целью работы является классификация отзывов китайских мобильных телефонов классификаторами методов опорных

векторов (SVM) и наивного байеса (NB). Оценка, использованная в этой работе, является звездами в iTunes. В iTunes оценка идет от 1 до 5 звезд, где отзывы с 1 или 2 звездами отмечены как отрицательные, отзывы с 4 и 5 звездами отмечены как положительные, а отзывы с 3 звездами – нейтральные. Результаты показывают, что наивный байесовский классификатор работает лучше.

Винодхини и Чандрасекаран [Vinodhini, G., Chandrasekaran, R.M. 2013] оценили эффект метода главных компонент (PCA) с помощью двух методов классификации сентиментов: метод опорных векторов и наивный байес. Эксперименты проводятся по отзывам продукции. Производительность улучшается с использованием метода главных компонент в качестве метода выделения признаков (features).

Существует несколько работ, посвященных сентимент-анализу на уровне документов, обрабатывающих отзывы на фильмы. Пан [Pang, B. 2002] стал первым, кто применил к этому подходу машинное обучение. Предложенный метод, который оказался успешным в категоризации текста, не достиг хорошей производительности для классификации сентиментов. Пан также продемонстрировал, что бинарное представление (0 – негативный, 1 – положительный) является более значимым, чем частотное представление (сколько раз встречается слово в документе).

Авторы в [Zhang, Q et al. 2008] использовали Классификацию уменьшения ошибки (Classification by minimizing the error, CME), чтобы приписать оценку каждому предложению. Затем они использовали классификатор SVM, чтобы присвоить оценку каждому документу на основе значений некоторых признаков (эти признаки определяются на основе субъективности и релевантности во всех предложениях). Таким образом, блоги классифицируются в соответствии с их окончательной оценкой на основе оценки релевантности, умноженной на оценку мнения.

Первый свободно доступный корпус арабского языка (Opinion corpus of Arabic, OCA) для sentiment-анализа предложен [Rushdi Saleh et al. 2011, 2045-2054]. Корпус арабского языка состоит из 500 отзывов на фильмы, собранных на различных арабских веб-страницах, 250 из которых определены как положительные, а остальные – отрицательные. Кроме того, были проведены различные эксперименты на этом корпусе с использованием классификаторов NB и SVM.

Говиндарааян [Govindarajan, M. 2013, 139-146] предложил гибридный метод классификации, основанный на сочетании NB и генетического алгоритма (genetic algorithm, GA). В этом методе сначала строятся два основных классификатора NB и GA, чтобы присвоить оценку мнению, а затем проводится классификация нового отзыва путем объединения прогнозов двух основных классификаторов большинством голосов. Автор использовал набор 2000 отзыва на фильмы (они были извлечены из корпуса Бо Панга). Гибридный метод сравнивается с двумя базовыми классификаторами NB и GA. Результаты показали, что гибридный метод улучшил производительность.

Нгуен и соавт. [Nguyen, D.Q., et al. 2014, 128-135] предложили новый тип признака под названием «признак на основе рейтинга». Признак на основе рейтинга основан на утверждении, что балловые оценки (которых пользователи используют для категоризации сущностей в отзывах) могут предоставить полезную информацию для повышения эффективности классификации мнений. Для отзыва без какой-либо балловой оценки авторы используют регрессионную модель (regression model) для прогнозирования оценки. Они комбинируют рейтинговый признак с униграммами, биграммами и триграммами.

В [Tripathi, G., Naganna, S. 2015] авторы предлагают модель классификации сентиментов. Во-первых, различные схемы предварительной обработки применяются к набору данных (data set). Во-вторых, поведение классификаторов NB и SVM изучается в сочетании различных схем выбора

признаков. Результаты классификации ясно показывают, что линейные SVM дают большую точность, чем классификатор NB.

В [Duwairi, R.M. 2015, 166-170] авторы предлагают метод анализа настроений в арабских твитах с наличием диалектальных слов. Эти слова были заменены соответствующими словами из современного стандартного арабского языка (Modern Standard Arabic, MSA) с использованием диалектной лексики. Классификаторы NB и SVM использовались для определения полярности твитов. Используются две версии одного и того же набора данных. Первая версия состоит из твитов, содержащих диалектальные слова, а вторая версия состоит из твитов, содержащих переведенные слова. Результаты показывают, что замена диалектальных слов повышает точность классификаторов (3%).

1.2.3. Глубокое обучение

Глубокое обучение – это часть процесса машинного обучения, относящаяся к глубокой нейронной сети (Deep Neural Network). К ним относятся сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN), рекурсивные нейронные сети, глубокая сеть доверия (Deep belief Network, DBN).

В последние годы подходы глубокого обучения привлекли внимание исследователей, поскольку они значительно превосходили традиционные методы. Например, Паредес-Вальверде и др. [Paredes-Valverde 2017] предложили подход глубокого обучения для построения классификатора обнаружения сентиментов. Этот подход разделен на три основных модуля: (1) модуль предварительной обработки, (2) векторное представление слов и (3) модель CNN. Результаты показывают, что CNN превзошел традиционные модели, такие как SVM и NB.

В [Tang et al. 2015, 1422-1432] представляют модели нейронной сети Conv GRNN (General regression neural network, сверточная обобщенно-регрессионная нейронная сеть) и LSTM-GRNN (Long short-term memory долгая краткосрочная память, General regression neural network) для сентимент-классификации на уровне документа. Модель сначала изучает представление предложений с помощью сверточной нейронной сети или долгой краткосрочной памяти. После этого семантика предложений и их отношения адаптивно кодируются в представлении документа с помощью, закрытой рекуррентной нейронной сети. Они провели сентимент-классификацию на уровне документов по четырем наборам данных (отзывы) из IMDb и Yelp Dataset Challenge. Экспериментальные результаты показывают, что LSTM работает лучше, чем мультифильтрованная CNN при моделировании представления предложений.

1.2.4. Алгоритмы машинного обучения с учителем

Обозначим необходимые алгоритмы машинного обучения с учителем. Алгоритмами могут быть 2 вида наивного байесовского классификатора (Bernoulli, Multinomial), логистическая регрессия, дерево решений. Рассмотрим особенности каждого алгоритма в контексте решения задачи классификации текстов.

1) **Наивный байесовский алгоритм** – это алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков. Другими словами, НБА предполагает, что наличие какого-либо признака в классе не связано с наличием какого-либо другого признака. Например, фрукт может считаться яблоком, если он красный, круглый и его диаметр составляет порядка 8 сантиметров. Даже если эти признаки зависят друг от друга или от других признаков, в любом случае они вносят независимый вклад в вероятность того, что этот фрукт является яблоком. В связи с таким допущением алгоритм называется «наивным» [6 простых шагов...2015].

Модели на основе НБА достаточно просты и крайне полезны при работе с очень большими наборами данных.

Теорема Байеса позволяет рассчитать апостериорную вероятность $P(c/x)$ на основе $P(c)$, $P(x)$ и $P(x/c)$.

The diagram illustrates the Bernoulli formula for calculating the posterior probability $P(c|x)$. It shows the formula $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$ with arrows pointing from each term to its label: $P(x|c)$ is labeled 'Likelihood', $P(c)$ is labeled 'Class Prior Probability', $P(c|x)$ is labeled 'Posterior Probability', and $P(x)$ is labeled 'Predictor Prior Probability'. Below the formula, the expanded version is given: $P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$
$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Рисунок 1. Формула Бернулли

На рисунке выше:

- $P(c/x)$ – апостериорная вероятность данного класса c (т.е. данного значения целевой переменной) при данном значении признака x .

- $P(c)$ – априорная вероятность данного класса.
- $P(x/c)$ – правдоподобие, т.е. вероятность данного значения признака при данном классе.
- $P(x)$ – априорная вероятность данного значения признака [6 простых шагов...2015].

MNB_classifier

- Multinomial (мультиномиальное распределение). Используется в случае дискретных признаков. Например, в задаче классификации текстов признаки могут показывать, сколько раз каждое слово встречается в данном тексте [6 простых шагов...2015].

BernoulliNB_classifier

- Bernoulli (распределение Бернулли). Используется в случае двоичных дискретных признаков (могут принимать только два значения: 0 и 1). Например, в задаче классификации текстов с применением подхода «мешок слов» (bag-of-words) бинарный признак определяет присутствие (1) или отсутствие (0) данного слова в тексте [6 простых шагов...2015].

2) Дерево решений

Метод деревьев решений (decision tree) для задачи классификации состоит в том, чтобы осуществлять процесс деления исходных данных на группы, пока не будут получены однородные (или почти однородные) их множества. Совокупность правил, которые дают такое разбиение (partition), позволят затем делать прогноз (т.е. определять наиболее вероятный номер класса) для новых данных.

Метод деревьев решений применим для решения задач классификации, возникающих в самых разных областях, и считается одним из самых эффективных.

Итак, дерево решений – это модель, представляющая собой совокупность правил для принятия решений. Графически её можно представить в виде древовидной структуры, где моменты принятия решений соответствуют так называемым узлам (decision nodes). В узлах происходит ветвление процесса (branching), т.е. деление его на так называемые ветви (branches) в зависимости от сделанного выбора. Конечные (терминальные) узлы называют листьями (leafs, leaf nodes) – каждый лист – это конечный результат последовательного принятия решений.

Данные, подлежащие классификации, находятся в так называемом «корне» дерева. В зависимости от решения, принимаемого в узлах, процесс в конце концов останавливается в одном из листьев, где переменной отклика (искомому номеру класса) присваивается то или иное значение.

Идея метода

Метод деревьев решений реализует принцип так называемого «рекурсивного деления» (recursive partitioning). Эта стратегия также называется «Разделяй и властвуй». В узлах, начиная с корневого, выбирается признак, значение которого используется для разбиения всех данных на 2 класса. Процесс продолжается до тех пор, пока не выполнится критерий остановки. Это возможно в следующих ситуациях:

- Все (или почти все) данные данного узла принадлежат одному и тому же классу;
- Не осталось признаков, по которым можно построить новое разбиение;
- Дерево превысило заранее заданный «лимит роста» (если таковой был заранее установлен) [Lantz 2013, 119-123].

tree_0

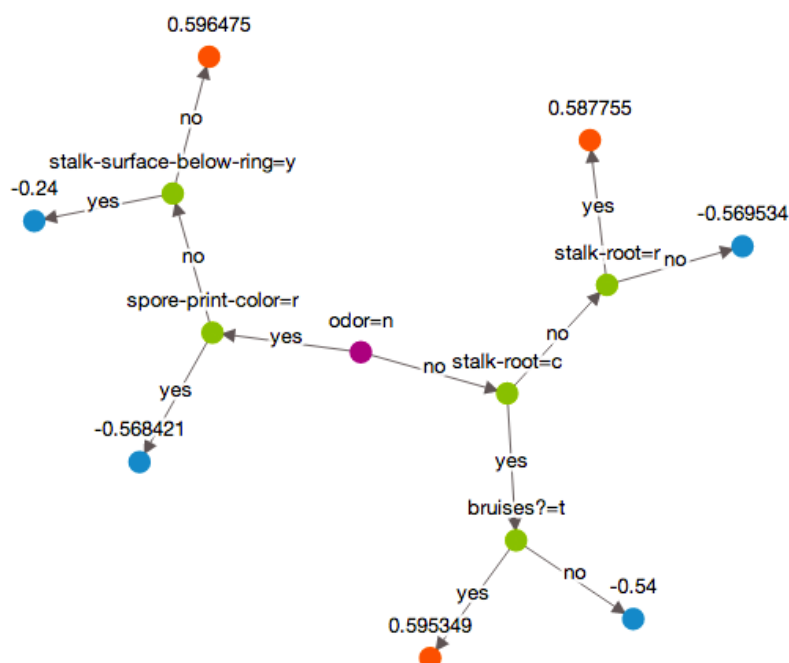


Рисунок 2. Схематичная иллюстрация алгоритма дерева решений

3) **Логистическая регрессия** – это не регрессия в обычном статистическом понимании, а один из методов классификации. Логистическая регрессия применяется для задач, где нужно соотнести данные по двум классам. В данной работе для сентимент-анализа с помощью логистической регрессии можно построить бинарный классификатор мнений.

Основная идея логистической регрессии заключается в том, что пространство исходных значений может быть разделено линейной границей (т.е. прямой) на две соответствующих классам области. Итак, что же имеется ввиду под линейной границей? В случае двух измерений — это просто прямая линия без изгибов. В случае трех — плоскость, и так далее. Эта граница задается в зависимости от имеющихся исходных данных и обучающего алгоритма. Чтобы все работало, точки исходных данных должны разделяться линейной границей на две вышеупомянутых области. Если точки исходных

данных удовлетворяют этому требованию, то их можно назвать линейно разделяемыми [Как легко...2019].

Чтобы задать логистическую регрессию мы используем библиотеку машинного обучения Sci-kit learn и пакет линейной модели в языке программирования Python и импортируем классификатор logistic regression:
`from sklearn.linear_model import LogisticRegression.`

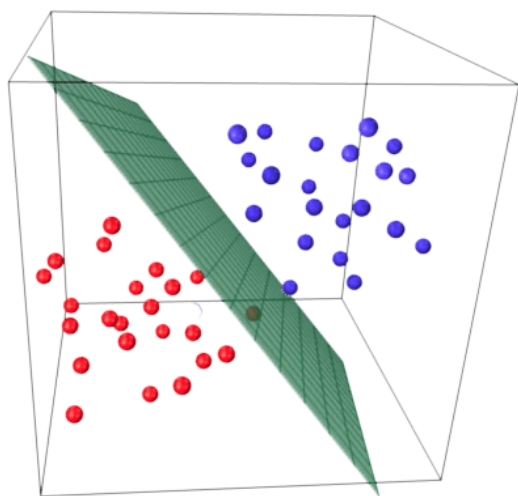


Рисунок 3. Схематичная иллюстрация алгоритма логистической регрессия

1.2.5 Предварительная обработка текста

Токенизация – это процесс разбиения последовательности строк на части: на слова, ключевые слова, фразы, символы и другие элементы, которые называются токенами [Обработка языковых данных... 2019].

```
In [3]: from nltk.tokenize import sent_tokenize, word_tokenize

example_text = "It's a good film."

print (sent_tokenize(example_text))
print(word_tokenize(example_text))

["It's a good film."]
['It', "'s", 'a', 'good', 'film', '.']
```

Рисунок 4. Токенизация

Лемматизация – это преобразование слов в лемму, то есть в их первоначальную словарную форму [Лемматизация... 2019]. Например:

running -> run

cats -> cat

При лемматизации части речи преобразуют по такому принципу:

1. Существительное — единственное число, именительный падеж.
2. Глагол — неопределенная форма (инфинитив).

```
In [9]: from nltk.stem import WordNetLemmatizer

        lemmatizer = WordNetLemmatizer()

        print(lemmatizer.lemmatize("better", pos = "a"))
        print(lemmatizer.lemmatize("running", pos = "v"))
        print(lemmatizer.lemmatize("running", pos = "n"))

good
run
running
```

Рисунок 5. Лемматизация

Стемминг (англ. stemming) или морфологический поиск - процесс поиска основы слова с учетом морфологии исходного слова. Стемминг подразумевает морфологический разбор слова с нахождением общей для всех его грамматических форм основы, отбрасывая суффиксы и окончания.

Поисковые машины, применяя в алгоритмах работы принцип стемминга, позволяют производить поиск с учетом морфологии слова. То есть, при вводе ключевого слова, поисковик учитывает все словоформы этого слова и отражает это в поисковой выдаче.

Например, при вводе поискового запроса "write" в поисковой выдаче так же будут присутствовать словоформы с основой исходного слова, такие как, "writing", "written", "wrote" и т.д. [Стемминг... 2019].

```
In [1]: from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()

new_text = "They write an improbably essaistic romance for the competitionaly. Yesterday they were wrting novel for the poetry"
words = word_tokenize(new_text)

for w in words:
    print (ps.stem(w))
```

they
write
an
improb
essaist
romanc
for
the
competitiionali
.
yesterday
they
were
write
novel
for
the
poetri
meet

Рисунок 6. Стемминг

Стоп-слова (иначе называемые шумовыми) – это слова, знаки, символы, которые самостоятельно не несут никакой смысловой нагрузки и просто игнорируются поисковыми системами при осуществлении ранжирования или индексации сайтов. Но которые, тем не менее, необходимы для нормального восприятия текста, его целостности, читабельности. Без использования стоп-слов невозможно создать полноценный контент, хорошо воспринимаемый не только поисковиками, но и людьми.

Удаляют стоп-слова с целью уменьшения размеров индекса, снижения нагрузок на сервер, рационального использования пространства баз данных. Кроме того, вычеркивание стоп-слов из запросов позволяет сократить количество операций по поиску каждого элемента ключевой фразы, а значит, повысить скорость, эффективность поиска нужной информации, сохранив релевантность [Стоп-слова...2019].

Частеречная разметка (автоматическая морфологическая разметка, POS tagging, part-of-speech tagging) – этап автоматической обработки текста, задачей которого является определение части речи и грамматических характеристик слов в тексте (корпусе) с приписыванием им соответствующих

тегов. POS tagging является одним из первых этапов компьютерного анализа текста [Daniel Jurafsky Speech... 2019].

CC	Coordinating conjunction	NNS	Noun, plural	UH	Interjection
CD	Cardinal number	NNP	Proper noun, singular	VB	Verb, base form
DT	Determiner	NNPS	Proper noun, plural	VBD	Verb, past tense
EX	Existential there	PDT	Predeterminer	VBG	Verb, gerund or present participle
FW	Foreign word	POS	Possessive ending	VBN	Verb, past participle
IN	Preposition or subordinating conjunction	PRP	Personal pronoun	VBP	Verb, non-3rd person singular present
JJ	Adjective	PRP\$	Possessive pronoun	VBZ	Verb, 3rd person singular present
JJR	Adjective, comparative	RB	Adverb	WDT	Wh-determiner
JJS	Adjective, superlative	RBR	Adverb, comparative	WP	Wh-pronoun
LS	List item marker	RBS	Adverb, superlative	WP\$	Possessive wh-pronoun
MD	Modal	RP	Particle	WRB	Wh-adverb
NN	Noun, singular or mass	SYM	Symbol		
		TO	to		

Рисунок 8. Части речи в the Penn Treebank

Распознавание именованных сущностей

Одной из разновидностей информационного поиска является задача извлечения информации, т.е. извлечение структурированных данных из неструктурированных документов. Одной из задач извлечение информации является задача распознавания именованных сущностей (named entity recognition, NER) состоит. Задача распознавания именованных сущностей – это выделение в тексте последовательностей слов, являющихся именованными сущностями, и классификация выделенных именованных сущностей. Примерами классов именованных сущностей являются имена людей, названий организаций, географических названий, прочие типы имен собственных, а также выражения специального вида, такие, как обозначения моментов времени, дат, денежные суммы и процентные выражения [Nadeau, D. 2007, 3-26].

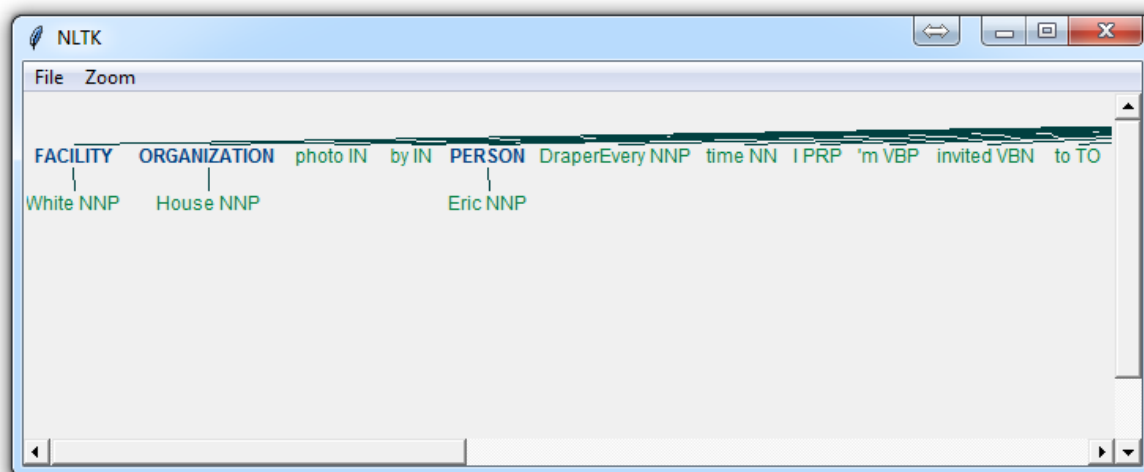


Рисунок 9. Представление сущностей в NLTK

Стоп-слова (иначе называемые шумовыми) – это слова, знаки, символы, которые самостоятельно не несут никакой смысловой нагрузки и просто игнорируются поисковыми системами при осуществлении ранжирования или индексации сайтов. Но которые, тем не менее, необходимы для нормального восприятия текста, его целостности, читабельности. Без использования стоп-слов невозможно создать полноценный контент, хорошо воспринимаемый не только поисковиками, но и людьми.

Удаляют стоп-слова с целью уменьшения размеров индекса, снижения нагрузок на сервер, рационального использования пространства баз данных. Кроме того, вычеркивание стоп-слов из запросов позволяет сократить количество операций по поиску каждого элемента ключевой фразы, а значит, повысить скорость, эффективность поиска нужной информации, сохранив релевантность [Стоп-слова... 2012-2019].

1.2.6 Перекрестная проверка для данных

Полученные данные делятся на обучающую выборку (**training set**) и тестовую выборку (**test set**). Они делятся по соотношению примерно 70% и 30%. Алгоритм учится на обучающей выборке и выделяет ключевые слова (features). На обучающей выборке проверяют алгоритмы машинного обучения несколько раз. Также тестовая выборка должна быть обработана на алгоритмах несколько раз. Результаты классификаторов и признаков (features) каждый раз должны быть разными. Этот процесс называется метод перекрестной проверки (**cross-validation**). Если результаты выходят одинаковыми, то возникает проблема переобученного алгоритма (overfitting). Это значит, что тестовые и обучающие выборки почти одинаковы. И если заменить тему тестовой выборки, то алгоритмы не смогут правильно классифицировать, так как они приспособились к одной тематике [Cross-validation... 2019].

Выводы по главе I

Задача сентимент-анализа состоит в классификации текстов по положительным и негативным полярностям. Задача решается в рамках обработки естественного языка и науки о данных.

Одной из задач обработки естественного языка является извлечение информации по определённым параметрам из неструктурированных данных. С развитием интернета в сети появилось большое количество субъективных текстовых данных. С помощью сентимент-анализа можно извлечь и классифицировать эти текстовые данные по определённым параметрам (по базовым эмоциям, положительная и негативная полярности). Задача компьютерной лингвистики заключается в создании обучающего текстового корпуса и в подготовке тестовой выборки, а также в создании теоретической лингвистической базы сентимент-анализа. Лингвисты соглашались в выделении 3 уровней сентимент-анализа для классификации: уровень документа, предложения, аспекта. В данной работе проводится классификация на уровне документа, так как это базовый уровень. В качестве документов отобраны отзывы на университеты. Особенностью уровня документа является предположение о том, что, во-первых, в документе текст субъективен, во-вторых, речь идет об одной сущности, в-третьих, и к этой сущности выражено негативное или позитивное мнение. Для классификации на уровне документа нужно создать обучающий корпус, где предварительно размечаются тексты на негативный и позитивный по пользовательским рейтинговым звездам.

Далее для классификации текстов применяются инструменты науки о данных. Наука о данных занимается сбором данных и их классификацией с применением методов машинного обучения, статистики и программирования. Для классификации текстовой информации в науке о данных как правило используют метод машинного обучения с учителем. Классификаторами метода машинного обучения с учителем являются наивный байесовский классификатор, логистическая регрессия и дерево решений.

Глава II Описание скрипта алгоритмов машинного обучения для классификации отзывов на университеты

Это задание из области обработки естественного языка – сентимент-анализ отзывов на российские университеты. Данные извлекаем из отзывов на Google. Отзывы классифицируем на негативные и положительные. Негативные – 1, 2 звезды. Положительные – 3, 4, 5 звёзд.

Во второй главе рассматривается машинный скрипт, где показаны этапы написания классификаторов. В основные этапы включают:

- 1) загрузки всех необходимых библиотек
- 2) сбор отзывов университетов
- 3) предварительная обработка: привести к нижнему регистру и удалить знаки препинания
- 4) Вывод на экран таблицы отзывов с помощью библиотеки pandas датафрейм и подсчет количества отзывов
- 5) обучение классификатора на обучающей выборке
- 6) применить классификаторы – метод опорных векторов (svm), логистическая регрессия, случайный лес
- 7) оценить точность (accuracy) и полноту (recall) классификаторов

```
In [1]: import pandas as pd
import numpy as np
import re

!pip install stop-words

import stop_words

Requirement already satisfied: stop-words in c:\programdata\ana...
```

Рисунок 10. Импорт библиотек

Скачиваем библиотеки `pandas`, `numpy`. Импортируем регулярные выражения, стоп-слова.

`pandas` – программная библиотека на языке Python для обработки и анализа данных. [pandas... 2019] Применяем библиотеку `pandas` для того, чтобы вывести пакет данных (data frame).

`NumPy` – это библиотека языка Python, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых (и очень быстрых) математических функций для операций с этими массивами [NumPy... 2012-2017]. Применяем библиотеку `NumPy` для обработки массивов данных.

Регулярные выражения (англ. *regular expressions*) – формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов (символов-джокеров, англ. *wildcard characters*) [Регулярные выражения... 2019]. Применяем регулярные выражения для поиска и обработки текстовых данных.

Устанавливаем через `pip` [pip... 2012-2017] стоп-слова и импортируем их. Рисунок 10.

```
In [7]: import nltk
```

Рисунок 11. Импорт nltk

Импортируем библиотеку `nltk` – пакет библиотек и программ для символьной и статистической обработки естественного языка, написанных на языке программирования Python [Natural Language, 2019]. Рисунок 11.

```
In [9]: from nltk.stem import SnowballStemmer
```

Рисунок 12. Импорт SnowballStemmer

Из пакета `nltk.stem` импортируем `Snowball Stemmer`. Пакет `stem` применяется, чтобы удалить морфологические аффиксы из слов, оставляя только основу слова (`stem`) [Stemmers... 2019]. Рисунок 12.

```
In [10]: stemmer_ru = SnowballStemmer('russian')
         stemmer_eng = SnowballStemmer('english')
```

Рисунок 13. Языки для стемминга

`Snowball` – это небольшой язык обработки строк, предназначенный для создания алгоритмов стемминга для использования в поиске информации [Snowball 2019]. Модуль `SnowballStemmer` применяется, чтобы указать языки для стемминга [snowballstemmer... 2019]. В данном случае это русский и английский языки. Рисунок 13.

```
In [11]: from nltk import word_tokenize
```

Рисунок 14. Импорт word_tokenize для токенизации

Из библиотеки `nltk` импортируем пакет `word_tokenize`, чтобы токенизировать знаки.

```
In [12]: from nltk.corpus import stopwords
stop_ru = set(stopwords.words('russian'))
stop = stop_ru | set(stop_words.get_stop_words('ru')) - set('год')
```

Рисунок 15. Импорт stopwords

Из пакета `nltk.corpus` импортируем стоп-слова из русского языка. Применяем `nltk.corpus`

Модули в этом пакете предоставляют функции, которые могут быть использованы для чтения файлов корпусов в различных форматах [Source code... 2019].

```
from sklearn.linear_model import LogisticRegression
```

Рисунок 16. Импорт логистической регрессии

Применяем библиотеку машинного обучения `sklearn`. Эту библиотеку применяют для извлечения и анализа данных [Scikit-learn... 2019].

Применяем пакет `sklearn.linear model`, так как в нем содержится набор методов для регрессии [Generalized Linear Models... 2019].

Импортируем из библиотеки логистическую регрессию для классификации данных.

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

Рисунок 17. Импорт Tf-idf Vectorizer

Из `sklearn.feature_extraction.text` импортируем `TfidfVectorizer`. Субмодуль `sklearn.feature_extraction.text` собирает утилиты, чтобы построить векторы признаков из текстовых документов [API Reference...2019]. `TfidfVectorizer` преобразовывает коллекцию необработанных документов в матрицу функций TF-IDF [`sklearn.feature_extraction.text.TfidfVectorizer...` 2019].

```
from sklearn.model_selection import cross_val_score, train_test_split, StratifiedKFold, GridSearchCV
```

Рисунок 18. Разделение обучающей и тестовой выборок

Из `sklearn.model_selection` импортируем перекрестную проверку (`cross_val_score`), разделитель обучающей и тестовой выборки (`train_test_split`), `StratifiedKFold` (перекрестная проверка `KFold`), `GridSearchCV`.

`Cross_val_score` оценивает счет путем перекрестной проверки [`sklearn.model_selection.cross_val_score...2019`].

`train_test_split` разбивает массивы или матрицы на неупорядоченные обучающую и тестовую выборки [`sklearn.model_selection.train_test_split...2019`].

`StratifiedKFold` предоставляет показатели обучающей/тестовой выборок для разделения данных в обучающих/тестовых наборах [`StratifiedKFold...2019`].

`GridSearchCV` реализует метод «подгонки» и «оценки». Он также реализует «прогнозирование», «прогнозирование_процесса», «решение_функции», «преобразование» и «обратное преобразование», если они реализованы в используемой оценочной функции [`sklearn.model_selection.GridSearchCV...2019`]. Рисунок 18.

```
: from sklearn.ensemble import RandomForestClassifier
```

Рисунок 19. Импорт классификатора случайного леса

Из `sklearn.ensemble` импортируем классификатор случайного леса (`RandomForestClassifier`).

Цель методов ансамбля состоит в том, чтобы объединить предсказания нескольких базовых оценок, построенных с данным алгоритмом обучения для улучшения обобщаемости / устойчивости по одной оценке [`Ensemble methods`].

Случайный лес – это мета-оценка, которая соответствует ряду классификаторов дерева решений для различных подвыборок набора данных и использует усреднение для повышения точности прогнозирования и контроля переобучения [sklearn.ensemble.RandomForestClassifier]. Рисунок 19.

```
from sklearn.svm import SVC
```

Рисунок 20. Импорт классификатора опорных векторов

sklearn.svm – набор методов машинного обучения с учителем, используемые для классификации, регрессии и обнаружения выбросов [Support Vector Machines...2019]. Из машины опорных векторов (sklearn.svm) импортируем классификатор опорных векторов (SVC). Рисунок 20.

```
: from sklearn.metrics import f1_score, confusion_matrix, accuracy_score, recall_score, precision_score
```

Рисунок 21. Импорт метрик машинного обучения

Модуль sklearn.metrics включает функции оценки, показатели производительности и попарные показатели, а также вычисления расстояний [API Reference... 2019].

Мера F1 (f1_score) может быть интерпретирована как взвешенное среднее значение точности и полноты, где мера F1 достигает своего лучшего значения при 1 и худшего значения при 0. Относительный вклад точности и полноты в мере F1 равны. Формула для оценки F1: $F1 = 2 * (\text{точность} * \text{полнота}) / (\text{точность} + \text{полнота})$ [sklearn.metrics.f1_score...2019]

Цель матрицы неточностей (confusion_matrix) – оценка точности классификации. По определению матрица неточностей C – это матрица C_{ij} , которая находится в группе i, но согласно прогнозам, относится к группе j. Таким образом, в двоичной классификации количество истинных негативов равно $C_{0,0}$, ложных негативов – $C_{1,0}$, истинных позитивов – $C_{1,1}$, а ложных срабатываний – $C_{0,1}$ [sklearn.metrics.confusion_matrix...2019].

Accuracy (accuracy_score) – доля правильных ответов алгоритма [Метрики в задачах... 2006-2019].

Для оценки качества работы алгоритма на каждом из классов по отдельности введем метрики precision (точность) и recall (полнота). Precision можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм [Метрики в задачах...2006-2019]. Рисунок 21.

```
from scipy.sparse import csr_matrix, hstack
```

Рисунок 22. Построение разреженной матрицы

Применяем библиотеку SciPy – библиотека для языка программирования Python с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчётов [SciPy...2018].

Импортируем csr_matrix, hstack для построения разреженной матрицы [scipy.sparse.hstack 2008-2019]. Рисунок 22.

```
df1 = pd.read_csv('data_universities.csv', engine='python', sep=';', names=['text', 'class'])
#df2 = pd.read_csv('clear_texts_01.csv', sep=';', encoding='cp1251')

#df = pd.concat([df1, df2])
df = df1.copy()
df.drop_duplicates(inplace=True)
df['text'] = df['text'].str.replace('<br>', ' ')
```

Рисунок 23. Создание дата фрейма

Создаем дата фрейм. Прописываем путь к файлу csv. Применяем библиотеку pandas. В дата фрейме два столбца: text и class.

```
# Разметим датафрейм
# 1,2 - 0
# 3,4,5 - 1
df['sentiment'] = (df['class'] > 3).astype(int)
```

Рисунок 24. Разметка датафрейма

Идентифицируем столбик class. Если пользовательских звезд в отзывах университетов 1, 2 – значит отзыв негативен. Если звезд 3, 4, 5 – отзыв положителен. Рисунок 24.

```
: # drop class
df.drop('class', axis=1, inplace=True)
```

Рисунок 25. Drop

Drop указывает метки (labels) для строк или столбцов [pandas.DataFrame.drop]. Рисунок 25

```
In [16]: df.head()
```

```
Out[16]:
```

	text	sentiment
0	Даже если ты уже давно не студент двери универ...	1
1	Основание университету в Томске в составе 4 фа...	1
2	НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ГОСУДАР...	1
3	Моя жизнь была долго связана с этим университе...	0
4	Кто такой Тура Партхаяна? Я слышал о нем по те...	1

Рисунок 26. Начало дата фрейма

Выводим на экран первые строки датафрейма с помощью команды df.head(). Рисунок 26.

```
In [17]: df.sentiment.replace(2, 1, inplace = True)
```

Рисунок 27. Sentiment

Вводим в датафрейм новый столбец sentiment, который будет обозначать положительный отзыв как 1, а негативный как 0. Рисунок 27.

```
In [18]: df.sentiment.value_counts()
Out[18]: 1    200
         0     42
         Name: sentiment, dtype: int64
```

Рисунок 28. Количество отзывов

Применяем метод value_counts(), который считает количество положительных (1) и негативных (0) отзывов. На экран выводится, что в датафрейме 200 положительных и 42 негативных отзыва.

```
In [19]: df
```

```
Out[19]:
```

	text	sentiment
0	Даже если ты уже давно не студент двери универ...	1
1	Основание университету в Томске в составе 4 фа...	1
2	НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ГОСУДАР...	1
3	Моя жизнь была долго связана с этим университе...	0
4	Кто такой Тура Партхаяна? Я слышал о нем по те...	1
5	Ставлю 3 балла и то с натяжкой. Практический з...	0
6	ТГУ ГГФ дал мне настоящую профессию как по тео...	0
7	Роца классная, преподаватели добрые, деканат о...	1

Рисунок 29 а. Датафрейм

235	Уровень образования оставляет желать лучшего	1
236	Одно из красивейших зданий Москвы. Особенно по...	1
237	Лучшие преподаватели, халявишь когда хочешь ха...	1
238	Лучший инженерный ВУЗ страны	1
239	Учусь пока что первый год, и нравится	1
240	Лучше чем моя шарага	1
241	Красивое здание, красивые виды, прекрасные люди	1

242 rows x 2 columns

Рисунок 29 б. Датафрейм

Выводим на экран все отзывы.

```
df['text'] = df['text'].str.replace(r'\[[a-zA-Za-яА-Я\.\ \/:\_\\-0-9\\>\<\?\\+\\,]+\] shared a link.', '')
df['text'] = df['text'].str.replace(r'\[[a-zA-Za-яА-Я\.\ \/:\_\\-0-9\\>\<\?\\+\\,]+\]', '')
df['text'] = df['text'].str.replace(r'(?:(?:https?|ftp):\\/\\)?[\\w\\/\\-?=%.&]+\\. [\\w\\/\\-?=%.&]+', '')
df['text'] = df['text'].str.replace('shared a ', '')
df['text'] = df['text'].str.replace(r'\\w', ' ')
```

Рисунок 30. Предобработка текста

Проводим предварительную обработку текста: убираем знаки препинания, числа и возводим все к единому регистру. Рисунок 30

```
def string_transform(string):
    string = re.sub('[\\W0-9]', ' ', string)
    string = string.split()
    string = [stemmer_eng.stem(stemmer_ru.stem(i)) for i in string if i not in stop]
    return ' '.join(string)

df['new_text'] = df.text.astype(str).apply(string_transform)
```

Рисунок 31. Стемминг

Стематизируем слова. Рисунок 31.

```
words = (' '.join(df['new_text'])).split()
all_words = nltk.FreqDist(w.lower() for w in words)
word_features = [w for (w, ct) in all_words.most_common(20)]
```

Рисунок 32 а. Вывод на экран наиболее частотных слов

Создаем кортеж `words`, где используем метод `join`, чтобы разделить пробелом данные в колонке `text`. Также используем метод `split`, чтобы разделить колонку `text` от других колонок (`ratings`, `sentiment`). Создаем строку `all_words`, где `nltk.FreqDist` – частота повторения слова в тексте, `w.lower()` – это метод, который переводит символы на нижний регистр. Выделяем принаковые слова `word features`. `word_features` – это список, указывающий на наиболее частотные 20 слов. Рисунок 32 а.

```
In [49]: word_features
```

```
Out[49]: ['в',  
          'и',  
          'не',  
          'на',  
          'что',  
          'с',  
          'университет',  
          'по',  
          'из',  
          'для',  
          'это',  
          'я',  
          'но',  
          'вуз',  
          'а',  
          'как',  
          'за',  
          'здесь',  
          'очень',  
          'россии']
```

Рисунок 32 б. Вывод на экран наиболее частотных слов

Выводим на экран наиболее 20 частотных слов с помощью `word_features`. Рисунок 32 б.

```
In [24]: df.head()
```

```
Out[24]:
```

	text	sentiment	new_text
0	Даже если ты уже давно не студент двери универ...	1	даж студент двер университет открыт прогуля па...
1	Основание университету в Томске в составе 4 фа...	1	основан университет томск состав факультет ист...
2	НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ГОСУДАР...	1	национальн исследовательск томск государствен ...
3	Моя жизнь была долго связана с этим университе...	0	мо связа университет мог обычн провинциальн ву...
4	Кто такой Тура Партхаяна Я слышал о нем по те...	1	кто тур партхая я слыша телевиден влогер индон...

Рисунок 33. Результат стемминга

Данные после стемминга: слова без аффиксов и сведены к нижнему регистру. Рисунок 33.

```
data = df['new_text']
tf_idf = TfidfVectorizer(max_features = 10000, min_df=5, ngram_range = (1,2))

X_train, X_test, y_train, y_test = train_test_split(data, df['sentiment'], test_size = 0.20, random_state = 42)
#model.fit(X_train, y_train, plot = False, eval_set = (X_test, y_test))

train_corpus = tf_idf.fit_transform(X_train)
test_corpus = tf_idf.transform(X_test)

#model = CatBoostClassifier(verbose = False, max_depth = 3, learning_rate=0.3, loss_function = 'MultiClass', iter
model1 = LogisticRegression(solver = 'sag', max_iter = 1000)

model1.fit(train_corpus, y_train)
prediction1 = model1.predict(test_corpus)
print('acc = {}, recall = {}'.format(accuracy_score(y_test, prediction1), recall_score(y_test, prediction1)))

acc = 0.8775510204081632, recall = 1.0
```

Рисунок 34. Алгоритм логистической регрессии

Векторизуем данные по частотности: если слово встречается 5 и более раз, то это слово является признаковым словом.

Разделяем на обучающую и тестовую выборки. Обучающая выборка составляет 80 %, тестовая выборка – 20 %.

Для классификации применяем алгоритм логистической регрессии. Для сравнения алгоритмов применяем 2 метрики: точность (accuracy) и полнота (recall). Получаем accuracy – 0.877, recall – 1.0. Рисунок 34

```
In [26]: confusion_matrix(y_test, prediction1)
Out[26]: array([[ 0,  6],
               [ 0, 43]], dtype=int64)
```

Рисунок 35. Матрица неопределенности

```

from sklearn.svm import LinearSVC

model2 = LinearSVC(loss = 'squared_hinge',
                  class_weight = {1:7},
                  C = 0.2)

model2.fit(train_corpus, y_train)
prediction2 = model2.predict(test_corpus)
print('acc = {}, recall = {}'.
      format(accuracy_score(y_test, prediction2), recall_score(y_test, prediction2)))

acc = 0.8775510204081632, recall = 1.0

```

Рисунок 36. Классификатор опорных векторов.

Применяем алгоритм классификации линейного метода опорных векторов. Рисунок 36 все так же как у логистической регрессии

Выводы по главе II

Основная задача построенного алгоритма – автоматическая классификация отзывов о российских университетах. Все упоминания о высших учебных заведениях собраны из отзывов в Google. Было собрано 242 упоминаний, относящихся к ТГУ, НГУ, ТПУ, ВШЭ, КФУ, МГИМО, СибГМУ, СПбГУ, МФТИ, РГГУ, РНИМУ, МГТУ, ИТМО. Массив упоминаний очищен от повторов и коротких сообщений длиной менее 220 знаков.

Для составления обучающей выборки вручную отобраны и размечены упоминаний. Использовались следующие понятия:

“Мусор” – упоминания, которые не содержат информации о качестве предоставления образовательных услуг, в основном это информационные сообщения, реклама, ошибочно собранные сообщения, не относящиеся к теме университета;

“Отзыв” – упоминания, которые будут использованы для описания качества предоставления образовательных услуг.

Для создания автоматического алгоритма классификации текстов использованы следующие стандартные библиотеки машинного обучения – Scikit Learn (<https://scikit-learn.org/stable/>), Pandas (<https://pandas.pydata.org/>), Numpy (<https://www.numpy.org/>) и набор инструментов для анализа естественного языка NLTK (Natural Language Toolkit, <https://www.nltk.org/>). Алгоритм реализован на языке программирования Python 3.

Сначала были загружены все упоминания из обучающей выборки, тексты очищены от ссылок, хеш-тегов, цифр, специальных символов и знаков препинания, удалены повторяющиеся пробелы. Далее, тексты были разбиты на отдельные слова и очищены от стоп-слов (слова, которые не несут смысловой нагрузки – местоимения, предлоги, числительные и т.д.).

Для применения разнообразных методов классификации, необходимо представить тексты упоминаний в векторном виде. В рамках работы

использован подход TF-IDF (TF — term frequency, IDF — inverse document frequency), в котором используются веса слов пропорциональные частоте употребления этих слова в документе и обратно пропорциональных частоте употребления слов во всех документах коллекции. Мера TF-IDF часто используется для представления документов коллекции в виде числовых векторов, отражающих важность использования каждого слова из некоторого набора слов (количество слов набора определяет размерность вектора) в каждом документе.

После того, как все упоминания были преобразованы в вектора, обучающая выборка упоминаний была перемешана в случайном порядке и разбита на две части — train, для обучения алгоритма, и test — для валидации алгоритма на незнакомых ему текстах, но имеющих размеченный вручную класс. Тестовая выборка включала 20 процентов всей обучающей выборки.

Для классификации полученных векторов использованы два алгоритма — логистическая регрессия и классификатор опорных векторов. Логистическая регрессия и классификатор опорных векторов показали одинаковые результаты. Была достигнута точность определения классов в 87,7%, но в то же время полнота (удельная доля всех найденных отзывов) 100%. Классификатор работает, но не совсем корректно. Причиной тому служит малый объем выборки.

Заключение

Целью дипломной работы являлась создание автоматического классификатора с применением методов машинного обучения на основе русскоязычных отзывов на университеты. Для достижения цели решены следующие задачи:

- 1) проведен обзор литературы по обработке естественного языка, sentiment-анализу, машинному обучению;
- 2) собраны обучающие тексты и тестовые тексты для классификации на негативные и положительные отзывы;
- 3) написан код классификатора на языке программирования Python с применением методов машинного обучения.

Построение классификатора sentiment-анализа – комплексная задача. Задачу можно решить с помощью инструментов обработки естественного языка, науки о данных и навыков программирования. Таким образом, классификатор положительных и негативных текстов можно рассматривать как результат междисциплинарных наук, потому что с одной стороны, материалом для классификатора выступает текст, написанный на естественном языке, с другой стороны, в качестве материально-технической базы для классификатора применяется машинное обучение – современный алгоритм решения многих наукоёмких задач. Теория машинного обучения включает линейную алгебру, математический анализ, статистику, аналитическую геометрию.

Сентимент-анализ находит свое применение в бизнес-сфере: например, чтобы знать отношение покупателей к товару, услуге, компании, бренду. Также алгоритм находит свое применение в сфере политики и социальных исследованиях: например, для оценки заинтересованности людей к массовым событиям (Чемпионат мира по футболу, Олимпийские игры, выборы Президента страны).

В качестве результата работы представлен машинный код классификатор, написанный на языке программирования Python.

В качестве продолжения работы можно сформулировать и решить задачу классификации текстовых документов по базовым эмоциям, задача классификации не только на уровне документов, а на уровне предложений и аспектов или задача классификации текстов с различной темой.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. 6 простых шагов для освоения наивного байесовского алгоритма (с примером кода на Python) [Электронный ресурс] // Data Review. Ваш проводник в мире анализа данных . – Электрон. дан. – [Б. м.], 2015. – URL: <http://datareview.info/article/6-prostyih-shagov-dlya-osvoeniya-naivnogo-bayesovskogo-algoritma-s-primerom-koda-na-python/> (дата обращения 31.05.2019).
2. About SenticNet [Electronic resource] // SenticNet. – Electronic data. – [S. l.], URL: <http://sentic.net/about/> (access date: 17.06.2019).
3. An Introduction to Machine Learning with Python (O'Reilly) by Andreas C. Mueller and Sarah Guido. Copyright 2017 Sarah Guido and Andreas Mueller
4. API Reference [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: <https://scikit-learn.org/stable/modules/classes.html> (дата обращения: 19.06.2019).
5. Bernardo Magnini, Gabriela Cavaglia [Integrating subject field codes into WordNet](#) (англ.) : [LREC](#). — 2000 (access date: 17.06.2019).
6. Bing Liu. Sentiment Analysis and Subjectivity // [Handbook of Natural Language Processing](#) / под ред. N. Indurkha и F. J. Damerau. – 2010 (access date: 17.06.2019).
7. Bird S., Klein E., Loper E. Natural language processing with Python: analyzing text with the natural language toolkit. – " O'Reilly Media, Inc.", 2009.
8. Bo Pang, Lillian Lee [A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts](#) (англ.) // Proceedings of the Association for Computational Linguistics (ACL): журнал. – 2004. – P. 271–278 (access date: 17.06.2019).
9. Carlo Strapparava, Alessandro Valitutti [WordNet-Affect: an Affective Extension of WordNet](#) (англ.) : [LREC](#). — 2004. — Vol. 4. — P. 1083–1086 (access date: 17.06.2019).

10. Cross-validation (statistics) [Electronic resource] // Wikipedia. – Electronic data. – [S. l.], 2019. – URL: [https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics)) (access date: 29.05.2019).
11. D., Martin J. H. Speech and language processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. – London : Pearson Prentice Hall, 2009. — 988 p.
12. Daniel Jurafsky Speech and Language Processing / Daniel Jurafsky, James H. Martin. – New Jersey: Prentice Hall; 2nd edition, 2008. – 950 p.
13. Duwairi, R.M. (2015) Sentiment Analysis for Dialectical Arabic. In Proceedings 6th ICICS International Conference on Information and Communication Systems, pp. 166 - 170
14. Ensemble methods [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: <https://scikit-learn.org/stable/modules/ensemble.html> (дата обращения: 19.06.2019).
15. Erik Cambria [SenticNet 2: A semantic and affective resource for opinion mining and sentiment analysis](#) (англ.) // Proceedings of AAAI FLAIRS : конференция. — 2012. —Р. 202–207 (access date: 17.06.2019).
16. Fangzhong Su, Katja Markert [From Words to Senses: a Case Study in Subjectivity Recognition](#) (англ.) // Proceedings of Coling, Manchester, UK. – 2008 (access date: 17.06.2019).
17. Generalized Linear Models [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: https://scikit-learn.org/stable/modules/linear_model.html (дата обращения: 19.06.2019).
18. Govindarajan, M. (2013) Sentiment Analysis of Movie Reviews using Hybrid Method of Naive Bayes and Genetic Algorithm. International Journal of Advanced Computer Research, 3 (4), pp. 139-146
19. Hart Laurel, "The Linguistics of Sentiment Analysis" (2013).University Honors Theses.Paper 20, 30 p.

20. Introduction to Sentiment Analysis [Electronic resource] // Algorithmia. – Electronic data. – [S. l.], 2019. – URL: <https://blog.algorithmia.com/introduction-sentiment-analysis/> (access date: 28.05.2019).
21. J. Ramteke, S. Shah, D. Godhia and A. Shaikh, "Election result prediction using Twitter sentiment analysis," *2016 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, 2016, pp. 1-5.
22. Lantz Brett. Machine Learning with R. Packt Publishing, Birmingham - Mumbai, 2013, 396 p.
23. Liu B. Sentiment analysis: Mining opinions, sentiments, and emotions. – Cambridge University Press, 2015, 381.
24. Nadeau, D. and Sekine, S., 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30 (1), pp. 3–26.
25. Natural Language Toolkit [Электронный ресурс] // Википедия : свободная энцикл. – Электрон. дан. – [Б. м.], 2019. – URL: https://ru.wikipedia.org/wiki/Natural_Language_Toolkit (дата обращения: 29.05.2019).
26. Nguyen, D.Q., Nguyen, D.Q., Vu, T., Pham, S.B. (2014) Sentiment Classification on Polarity Reviews: An Empirical Study Using Rating-based Features. In *Proceeding of 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Maryland pp 128–135.
27. Nilesh, M.S., Shriniwas Deshpande, Vilas Thakre. (2012) Survey of Techniques for Opinion Mining. *International Journal of Computer Applications*, 57(13), 0975-8887
28. NumPy [Электронный ресурс] // Python 3 для начинающих. – Электрон. дан. – [Б. м.], 2012-2017. – URL: <https://pythonworld.ru/numpy/1.html> (дата обращения: 19.06.2019).
29. Pak, A., Paroubek, P. (2011) Classification en polarité de sentiments avec une représentation textuelle à base de sous-graphes d'arbres de dépendances. *TALN*

30. pandas [Электронный ресурс] // Википедия : свободная энцикл. – Электрон. дан. – [Б. м.], 2019. – URL: <http://ru.wikipedia.org/wiki/Pandas> (дата обращения: 19.06.2019).
31. Pang B. et al. Opinion mining and sentiment analysis //Foundations and Trends® in Information Retrieval. – 2008. – Т. 2. – №. 1–2. – С. 1-135.
32. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In Proceedng of Empirical Methods in Natural Language Processing,. EMNLP (2002),79-86
33. Paredes-Valverde, M. A., Colomo-Palacios, R., Salas-Zárate, M. D. P., & Valencia-García, R. (2017). Sentiment Analysis in Spanish for Improvement of Products and Services: A Deep Learning Approach. Scientific Programming, 2017.
34. pip [Электронный ресурс] // Python 3 для начинающих – Электрон. дан. – [Б. м.], 2012-2017. – URL: <https://pythonworld.ru/osnovy/pip.html> (дата обращения: 08.11.2012).
35. Rafrafi, A., Guigue,V., Gallinari, P. (2011) Réseau de neurones profond et SVM pour la classification des sentiments. In Proceeding of Conférence en Recherche d'Information et Applications CORIA, 121-133
36. Rushdi-Saleh, M., Martín-Valdivia, M.T, Ureña-López, L.A., Perea-Ortega, J.M (2011) OCA: Opinion corpus for Arabic. ASIS&T. 62, 2045–2054
37. Scikit-learn [Электронный ресурс] // Scikit-learn: machine learning in Python. – Электрон. дан. – [Б. м.], 2019. – URL: <https://scikit-learn.org/stable/> (дата обращения: 19.06.2019).
38. SciPy [Электронный ресурс] // Википедия: свободная энциклопедия – [Б. м.], 2018. – URL: <https://ru.wikipedia.org/wiki/SciPy> (дата обращения: 19.06.2019).
39. scipy.sparse.hstack [Электронный ресурс] // Numpy and Scipy Documentation – [Б. м.], 2008-2019. – URL: <https://ru.wikipedia.org/wiki/SciPy> (дата обращения: 19.06.2019).
40. Sentiment Analysis: Concept, Analysis and Applications [Electronic resource] // Towards Data Science. – Electronic data. – [S. l.], 2019. – URL:

<https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17> (access date: 28.05.2019).

41. SentiWordNet [Electronic resource] // [S. I.], 2010. – URL: <http://sentiwordnet.isti.cnr.it/> (access date: 17.06.2019).

42.

Sindhu, C., ChandraKala, S. , (2013) A Survey On Opinion Mining And Sentiment Polarity Classification. J. IJETAE. 3

43. sklearn.ensemble.RandomForestClassifier [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> (дата обращения: 19.06.2019).

44. sklearn.feature_extraction.text.TfidfVectorizer [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (дата обращения: 19.06.2019).

45. sklearn.metrics.confusion_matrix [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html (дата обращения: 19.06.2019).

46. sklearn.metrics.f1_score [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (дата обращения: 19.06.2019).

47. sklearn.model_selection.cross_val_score [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html (дата обращения: 19.06.2019).

48. sklearn.model_selection.GridSearchCV [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)

(дата обращения: 19.06.2019).

49. sklearn.model_selection.StratifiedKFold [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

(дата обращения: 19.06.2019).

50. sklearn.model_selection.train_test_split [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

(дата обращения: 19.06.2019).

51. Snowball [Электронный ресурс] // Snowball – Электрон. дан. – [Б. м.], 2019. – URL: <https://snowballstem.org/> (дата обращения: 19.06.2019).

52. snowballstemmer 1.2.1 [Электронный ресурс] // PyPI – the Python Package Index – Электрон. дан. – [Б. м.], 2019. – URL: <https://snowballstem.org/> (дата обращения: 19.06.2019).

53. Source code for nltk.corpus [Электронный ресурс] // NLTK 3.4.3 documentation – Электрон. дан. – [Б. м.], 2019. – URL: https://www.nltk.org/_modules/nltk/corpus.html (дата обращения: 19.06.2019).

54. Stefano Baccianella [Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining](#) (англ.) // Proceedings of LREC : конференция. – 2010. – P. 2200–2204 (access date: 17.06.2019).

55. Stemmers [Электронный ресурс] // NLTK 3.4.3 documentation. – Электрон. дан. – [Б. м.], 2019. – URL: <http://www.nltk.org/howto/stem.html> (дата обращения: 19.06.2019).

56. Support Vector Machines [Электронный ресурс] // scikit-learn: machine learning in Python – [Б. м.], 2019. – URL: <https://scikit-learn.org/stable/modules/svm.html> (дата обращения: 19.06.2019).

57. Tang, D., Qin, B., & Liu, T. (2015) Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 1422–1432

58. Text Mining and Sentiment Analysis - A Primer [Electronic resource] // Data Science Central. – Electronic data. – [S. l.], 2019. – URL: <https://www.datasciencecentral.com/profiles/blogs/text-mining-and-sentiment-analyses-a-primer> (access date: 29.05.2019).
59. Tripathi, G., Naganna, S. (2015) Feature Selection And Classification Approach For Sentiment Analysis. MLAIJ. 2201
60. V. S. Pagolu, K. N. Reddy, G. Panda and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, Paralakhemundi, 2016, pp. 1345-1350.
61. Victoria Bobicev, Victoria Maxim, Tatiana Prodan, Natalia Burciu, Victoria Angheluș [Emotions in words: developing a multilingual WordNet-Affect](#) (АНГЛ.) : [CICLing](#) 2010, [Iasi](#), Romania. — 2010. — P. 1-10 (access date: 17.06.2019).
62. Vinodhini, G., Chandrasekaran, R.M. (2013) Effect of Feature Reduction in Sentiment analysis of online reviews. IJARCET, 2278 – 1323
63. Wang, S., Manning, C.D. (2012) Baselines and bigrams: simple, good sentiment and topic classification". In Proceeding of 50th Annual Meeting of the Association for Computational Linguistics, ACL,(2012). 90 – 94
64. Wordnet-Affect [Electronic resource] // WordNet Domains. – Electronic Data. – [S. l.], 2009. – URL: <http://wndomains.fbk.eu/wnaffect.html> (access date: 17.06.2019).
65. Zhang, L., Hua, K., Wang, H., Qian, G., Zhang, L. (2014) Sentiment Analysis on Reviews of Mobile Users. In Proceeding of 11th International Conference on Mobile Systems and Pervasive Computing, Procedia Computer Science 34, 458 – 465.
66. Zhang, Q., Wang, B., Wu, L., Huang, X. (2007) FDU at TREC 2007 : Opinion Retrieval of Blog Track. In Proceeding of E. M. Voorhees, , L. P. Buckland (eds), TREC 2007, vol. Special Publication ,pp.500- 274.

67. Автоматическая обработка текстов на естественном языке и анализ данных : учеб. пособие / Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. — М.: Изд-во НИУ ВШЭ, 2017. — 269 с.
68. [Апресян Ю. Д.](#) Коннотации как часть прагматики слова // Апресян Ю. Д. [Избранные труды. Т. 2. Интегральное описание языка и системная лексикография.](#) — М.: Школа «Языки русской культуры», 1995. — 766 с.
69. Апресян Ю. Д. Лексическая семантика. — М., 1974, с. 56–114.
70. Большие данные [Электронный ресурс] // Википедия : свободная энцикл. — Электрон. дан. — [Б. м.], 2019. — URL: http://ru.wikipedia.org/wiki/Большие_данные (дата обращения: 28.05.2019).
71. Как легко понять логистическую регрессию [Электронный ресурс] // Хабр . — Электрон. дан. — [Б. м.], 2019. — URL: <https://habr.com/ru/company/io/blog/265007/> (дата обращения: 29.05.2019).
72. Кондаков Н. И. Логический словарь-справочник. — М.: Наука, 1975.— 721 с.
73. Лемматизация [Электронный ресурс] // Агентство копирайтинга Text iS. — Электрон. дан. — [Б. м.], 2017. — URL: <http://textis.ru/lemmatizatsiya/> (дата обращения 17.06.2019).
74. Математическая лингвистика и автоматическая обработка текстов : учеб. пособие / Т. В. Батура ; Новосиб. гос. ун-т. — Новосибирск : РИЦ НГУ, 2016. — 166 с.
75. Метрики в задачах машинного обучения [Электронный ресурс] // Блог компании Open Data Science / Хабр — [Б. м.], 2006-2019. — URL: <https://habr.com/ru/company/ods/blog/328372/> (дата обращения: 19.06.2019).
76. Нелюбин Л. Л., Хухуни Г. Т. Наука о переводе (история и теория с древнейших времен до наших дней). — М.: Флинта: МПСИ, 2006. — 416 с.
77. Обработка языковых данных в Python 3 с помощью NLTK [Электронный ресурс] // Электрон. дан. — [С. л.], 2016. — URL:

<https://www.8host.com/blog/obrabotka-yazykovyx-dannyx-v-python-3-s-pomoshhyu-nltk/> (дата обращения 17.06.2019).

78. Обработка языковых данных в Python 3 с помощью NLTK [Электронный ресурс] // Электрон. дан. – [S. l.], 2016. – URL: <https://www.8host.com/blog/obrabotka-yazykovyx-dannyx-v-python-3-s-pomoshhyu-nltk/> (дата обращения 17.06.2019).

79. Регулярные выражения [Электронный ресурс] // Википедия : свободная энцикл. – Электрон. дан. – [Б. м.], 2019. – URL: http://ru.wikipedia.org/wiki/Регулярные_выражения (дата обращения: 19.06.2019).

80. Румынский и русский WordNet-Affect [Электронный ресурс] // Laboratorul de Inginerie a Limbajului Uman. – Электрон. дан. – [Б. м.], – URL: http://lilu.fcim.utm.md/resourcesRoRuWNA_ru.html (дата обращения: 17.06.2019).

81. Стемминг [Электронный ресурс] // Агентство копирайтинга Text iS. – Электрон. дан. – [Б. м.], 2017. – URL: <http://textis.ru/stemming/> (дата обращения 17.06.2019).


82. Стоп-слова [Электронный ресурс] // Агентство копирайтинга Text iS. – Электрон. дан. – [Б. м.], 2017. – URL: <http://textis.ru/stop-slova/> (дата обращения 17.06.2019).

83. Тест Тьюринга [Электронный ресурс] // Википедия: свободная энцикл. – Электрон. дан. – [Б. м.], 2019. – URL: https://ru.wikipedia.org/wiki/Тест_Тьюринга (дата обращения: 08.11.2019).


84. pandas.DataFrame.drop [Электронный ресурс] // pandas: Python Data Analysis Library – [Б. м.], 2019. – URL: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html> (дата обращения: 19.06.2019).

Выпускные ра... X fayl_01.pdf X AntiPlagiat X Краткий отчет X Вывод отчета X Новое письмо X из ворд в pdf X Скачать файл X + -

← → ↻ https://users.antiplagiat.ru/report/print/18?short=true&c=0 ☆

 **АНТИПЛАГИАТ**
ТВОРИТЕ СОБСТВЕННЫМ УМОМ

Отчет о проверке на заимствования №1




Автор: dolana.94@mail.ru / ID: 4146311
 Проверяющий: dolana.94@mail.ru / ID: 4146311
 Отчет предоставлен сервисом «Антиплагиат»- <http://users.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 18
 Начало загрузки: 25.06.2019 05:31:05
 Длительность загрузки: 00:00:02
 Имя исходного файла: дипломная сарыгбай 20.06.2019
 Размер текста: 1532 кБ
 Символов в тексте: 85814
 Слов в тексте: 10298
 Число предложений: 990

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
 Начало проверки: 25.06.2019 05:31:08
 Длительность проверки: 00:00:12
 Комментарии: не указано
 Модули поиска: Модуль поиска Интернет



ЗАИМСТВОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
18,58%	0%	81,42%

Показать все X

дипломная сарыг...pdf ^

Windows taskbar: 9:32 25.06.2019