МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ. АНАЛИЗ ТОНАЛЬНОСТИ ТЕКСТА.

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

Ройзиной Алины Иосифовны 2 курса 248 группы
направления 09.04.03 — Прикладная информатика

механико-математического факультета

Научный руководитель
доцент, д. ф.-м. н. _____ П. А. Терехин

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Обработка естественного языка (Natural Language Processing, NLP) — общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков. Применительно к искусственному интеллекту анализ означает понимание языка, а синтез — генерацию грамотного текста. Решение этих проблем будет означать создание более удобной формы взаимодействия компьютера и человека.

Ранние системы NLU анализировали естественный язык с помощью ручных правил для явных семантических представлений, и использовали вручную написанные автоматы для генерации конкретных ответов на выходе разбора. Такие системы, как правило, ограничиваются ситуациями, представленными разработчиком, и большая часть работы по разработке включает в себя создание дополнительных правил для повышения надежности анализа. Эти системы хрупки, и улучшение происходит медленно.

Статистические системы могут предложить более простой путь и более надежное поведение данных. Такие системы избегают ненужных ограничений и узких мест, неизбежно навязываемых разработчиком системы. В этом контексте, понимание естественного языка может быть оценено меньше в терминах явного семантического представления,благодаря полезности самой системы. Системам нужно не только изучать язык, но и учиться делать что-то полезное с этим. В практической части работы будет рассматриваться задача предложения ответов в отношении человека к человеку. Диалоговая система должна будет учиться быть последовательной в течение диалога, поддерживающей некоторую память от поворота к повороту. Для машинного обучения требуется большое количество данных и множество полезных пользователей для разработки посредством живых взаимодействий.

Автоматическая классификация эмоциональной окраски текстов, также известная под термином ‘анализ тональности ’, с каждым годом становится все более актуальной задачей с теоретической и практической точек зрения. В первую очередь, это связано с развитием интернета и изменением формата коммуникаций в современном мире – для подавляющего большин-

ства людей социальные сети стали занимать лидирующее положение среди остальных источников информации и площадок для дискуссий. Пользователями социальных сетей ежедневно генерируются значительные объемы текстовой информации. Анализ тональности текстов из социальных сетей применяется в бизнес сегменте, социальных и политических исследованиях:

— Определения уровня лояльности потребителя к бренду
— Определение политических взглядов горожан на основе сообщений в социальных сетях
— Прогнозирование результатов политических выборов

Текстам в социальных сетях более характерен разговорный стиль речи, нежели литературный. Как следствие, это вызывает серию существенных трудностей при автоматической обработке, так как в разговорном стиле чаще встречаются сленг, фразеологизмы, авторская пунктуация, опечатки и ошибки, а также другие стилистические особенности, которые сложно обрабатывать в автоматическом режиме.

Целью данной работы является изучение методов решения задачи автоматического анализа тональности текста, выбор инструментария, позволяющего реализовать систему автоматического анализа тональности текста, отвечающую следующим требованиям и реализация данной системы:

— Система должна определять тональность небольших текстовых высказываний (формат комментариев)
— Точность определения тональности текста должна быть не ниже 70%
— Определение тональности текста должно происходить в большинстве популярных предметных областей
— Проектируемая система должна определять тональность предложений с новыми словами или с использованием новых конструкций

Работа прошла апробацию на различных конференциях, в частности, на ежегодной студенческой конференции «Актуальные проблемы математики и механики» которую проводил механико-математический факультет СГУ в апреле 2019 года, в секции «Анализ данных» и в VII Международной молодежной научно-практической конференции «Математическое и компьютерное моделирование в экономике, страховании и управлении рисками», ноябрь 2018 года.

1 Анализ тональности текста

Компьютер считает быстрее человека, но гораздо хуже понимает естественный язык. Например, существуют два слова, обозначающие одно и то же. Человек, опираясь на свой жизненный опыт, на свои знания, может понять, что эти слова имеют одинаковую смысловую нагрузку. Компьютер же «думает» совсем иначе, сразу встает вопрос, как научить понимать компьютер, что значение этих слов одинаково. Для того чтобы подобраться к смысловому значению слова, необходимо произвести семантический анализ. Необходимо смоделировать, у каких слов схожее значение. Если слова обозначают одно и то же, то и их представления должны быть похожи.

Существует два фундаментальных подхода к моделированию семантики:

— Подход, построенный на знаниях. Также этот подход можно назвать подходом «сверху вниз». Это трудоемкий способ, требующий огромных человеческих ресурсов. В данном случае тысячи экспертов должны построить онтологическую модель, описав какие слова являются синонимами, антонимами, смысловыми подчастями других слов и т. д.

— Дистрибутивный подход или подход «снизу-вверх». Здесь извлекается значение из употребления слов в тексте.

Дистрибутивная семантика — это область лингвистики, которая занимается вычислением степени семантической близости между лингвистическими единицами на основании их распределения (дистрибуции) в больших массивах лингвистических данных (текстовых корпусах).

Каждому слову присваивается свой контекстный вектор. Множество векторов формирует словесное векторное пространство.

Семантическое расстояние между понятиями, выраженными словами естественного языка, обычно вычисляется как косинусное расстояние между векторами словесного пространства.

Дистрибутивная семантика основывается на дистрибутивной гипотезе: лингвистические единицы, встречающиеся в схожих контекстах, имеют близкие значения.

В качестве способа представления модели используются векторные пространства из линейной алгебры. Информация о дистрибуции лингвистиче-

ских единиц представляется в виде многоразрядных векторов, которые образуют словесное векторное пространство. Векторы соответствуют лингвистическим единицам (словам или словосочетаниям), а измерения соответствуют контекстам. Координаты векторов представляют собой числа, показывающие, сколько раз данное слово или словосочетание встретилось в данном контексте.

Векторная модель — в информационном поиске представление коллекции документов векторами из одного общего для всей коллекции векторного пространства.

Векторная модель является основой для решения многих задач информационного поиска, как то: поиск документа по запросу, классификация документов, кластеризация документов.

Документ в векторной модели рассматривается как неупорядоченное множество термов. Термами в информационном поиске называют слова, из которых состоит текст.

Различными способами можно определить вес терма в документе — «важность» слова для идентификации данного текста. Например, можно просто подсчитать количество употреблений терма в документе, так называемую частоту терма, — чем чаще слово встречается в документе, тем больший у него будет вес. Если терм не встречается в документе, то его вес в этом документе равен нулю.

Все термы, которые встречаются в документах обрабатываемой коллекции, можно упорядочить. Если теперь для некоторого документа выписать по порядку веса всех термов, включая те, которых нет в этом документе, получится вектор, который и будет представлением данного документа в векторном пространстве. Размерность этого вектора, как и размерность пространства, равна количеству различных термов во всей коллекции, и является одинаковой для всех документов $d_j = (w_{1j}, w_{2j}, \ldots, w_{nj})$ где d_j — векторное представление j-го документа, w_{ij}— вес i-го терма в j-м документе, n — общее количество различных термов во всех документах коллекции.

Располагая таким представлением для всех документов, можно, например, находить расстояние между точками пространства и тем самым решать задачу подобия документов — чем ближе расположены точки, тем больше

похожи соответствующие документы. В случае поиска документа по запросу, запрос тоже представляется как вектор того же пространства — и можно вычислять соответствие документов запросу.

В рамках магистерской работы использовался метод векторного представления слов основанный на технологии Word2Vec.

Word2vec — программный инструмент анализа семантики естественных языков, представляющий собой технологию, которая основана на дистрибутивной семантике и векторном представлении слов. Этот инструмент был разработан группой исследователей Google в 2013 году.

Работа этой технологии осуществляется следующим образом: word2vec принимает большой текстовый корпус в качестве входных данных и сопоставляет каждому слову вектор, выдавая координаты слов на выходе. Сначала он создает словарь, «обучаясь» на входных текстовых данных, а затем вычисляет векторное представление слов. Векторное представление основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами (а следовательно, имеющие схожий смысл), в векторном представлении будут иметь близкие координаты векторов-слов. Полученные векторы-слова могут быть использованы для обработки естественного языка и машинного обучения.

Получаемые на выходе координатные представления векторов-слов позволяют вычислять «семантическое расстояние» между словами. И, именно основываясь на контекстной близости этих слов, технология word2vec совершает свои предсказания. Так как инструмент word2vec основан на обучении нейронной сети, чтобы добиться его наиболее эффективной работы, необходимо использовать большие корпусы для его обучения. Это позволяет повысить качество предсказаний.

Модель, предложенная Миколовым очень проста (и потому так хороша) - предсказание вероятности слова по его окружению (контексту). То есть будут изучаться такие вектора слов, чтобы вероятность, присваиваемая моделью слову была близка к вероятности встретить это слово в этом окружении в реальном тексте.

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

Здесь w_o — вектор целевого слова, w_c — это некоторый вектор контекста, вычисленный (например, путем усреднения) из векторов окружающих нужное слово других слов. А $s(w_1, w_2)$ — это функция, которая двум векторам сопоставляет одно число. Например, это может быть упоминавшееся выше косинусное расстояние.

Приведенная формула называется softmax, то есть "мягкий максимум", мягкий — в смысле дифференцируемый. Это нужно для того, чтобы модель могла обучиться с помощью backpropagation, то есть процесса обратного распространения ошибки.

Процесс тренировки устроен следующим образом: берется последовательно (2k+1) слов, слово в центре является тем словом, которое должно быть предсказано. А окружающие слова являются контекстом длины по k с каждой стороны. В модели сопоставлен уникальный вектор, который меняется в процессе обучения модели.

В word2vec существуют два основных алгоритма обучения : CBOW (Continuous Bag of Words) и Skip-gram. CBOW — «непрерывный мешок со словами» модельная архитектура, которая предсказывает текущее слово, исходя из окружающего его контекста.В целом, этот подход называется CBOW — continuous bag of words, continuous потому, что в модели на вход подается последовательно наборы слов из текста, а BoW потому что порядок слов в контексте не важен.

Архитектура типа Skip-gram действует иначе: она использует текущее слово, чтобы предугадывать окружающие его слова. Пользователь word2vec имеет возможность переключаться и выбирать между алгоритмами. Порядок слов контекста не оказывает влияния на результат ни в одном из этих алгоритмов. Модель пытается из данного слова угадать его контекст (точнее вектор контекста). В остальном модель не претерпевает изменений.

Существует несколько основных вариантов задач анализа тональности:

1. Определение исключительно тональности текста - рассматривается только тональность мнения, которое выражено в тексте (часто предполага-

ется, что оно единственное).

Обычно тональность представлена определенной шкалой. Выделяют следующие типы шкал:

— двухзначная шкала. Шкала тональности имеет только два значения - положительная тональность и отрицательная.

— трехзначная шкала. К предыдущим двум вариантам добавляется третье значение - нейтральное, которое может обозначать либо отсутствие тональности, либо одновременное наличие как положительной, так и отрицательной тональности.

— многозначная шкала. Шкала тональности имеет более 3 значений. Существует множество вариантов таких шкал, отличающихся количеством значений тональности и наличием нейтрального значения.

2. Определение тональности, субъекта и объекта - кроме тональности мнения определяется выразитель мнения, субъект и объект , по отношению к которому выражается мнение.

Для решения задачи в такой постановке кроме методов определения тональности требуется также применение методов извлечения сущностей из текста.

3. Определение мнения в целом - мнение рассматривается как полное выражение, т. е. по сравнению с предыдущим вариантом кроме выделения сущности (объекта мнения) требуется определение её аспектов

В данной работе мы будем рассматривать задачу определения тональности короткого текста с трехзначной шкалой.

Для решения задачи данной работы - написания системы, определяющей тональность текста, необходимо выбрать метод машинного обучения, для него подобрать наилучшую программную реализацию - библиотеку. Затем необходимо выбрать язык программирования. Далее, подобрав корректные данные для обучения и тестирования системы, будет выбран и описан один из рассмотренных алгоритмов. После реализации выбранного алгоритма в системе, будет проведено обучение системы и тестирование получившихся результатов на заранее подготовленной выборке данных.

В работе было решено использовать сверточные нейронные сети в качестве классификатора, так как в последнее время во многих конкурсах (вста-

вить ссылки и названия) стоит задача определения тональности коротких текстовых сообщений из мессенджеров, и была сформирована гипотеза: при определении тональности с техзначной шкалой для коротких текстовых сообщений сверточные нейронные сети показывают наилучшие результаты, среди множества других методов машинного обучения. Так как в данной работе будут соблюдены все условия данной гипотезы - будет считать сверточные нейронные сети наилучшим методом машинного обучения для задачи определения тональности текста.

Сверточная нейронная сеть (convolutional neural network, CNN) — специальная архитектура искусственных нейронных сетей, предложенная Яном Лекуном в 1988 году и нацеленная на эффективное распознавание образов. Использует некоторые особенности зрительной коры, в которой были открыты так называемые простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определенного набора простых клеток. Таким образом, идея сверточных нейронных сетей заключается в чередовании сверточных слоев (convolution layers) и субдискретизирующих слоев (subsampling layers или pooling layers, слоёв подвыборки). Структура сети — однонаправленная (без обратных связей), принципиально многослойная. Для обучения используются стандартные методы, чаще всего метод обратного распространения ошибки. Функция активации нейронов (передаточная функция) — любая, по выбору исследователя.

Название архитектура сети получила из-за наличия операции свертки, суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свертки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения.

Для решения задачи анализа тональности текста была использована архитектура сети, предложенная на крупнейшем ежегодном соревновании по компьютерной лингвистике SemEval-2017 и занявшей первые места в пяти номинациях в задаче по анализу тональности.

Входными данными сети является матрица с фиксированной высотой n, где каждая строка представляет собой векторное отображение токена в признаковое пространство размерности k. Для формирования признаково-

го пространства часто используют инструменты дистрибутивной семантики, такие как Word2Vec, Glove, FastText и т.д.

На первом этапе входная матрица обрабатывается слоями свертки. Как правило, фильтры имеют фиксированную ширину, равную размерности признакового пространства, а для подбора размеров у фильтров настраивается только один параметр — высота h. Получается, что h — это высота смежных строк, рассматриваемых фильтром совместно. Соответственно, размерность выходной матрицы признаков для каждого фильтра варьируется в зависимости от высоты этого фильтра h и высоты исходной матрицы n.

Далее карта признаков, полученная на выходе каждого фильтра, обрабатывается слоем субдискретизации с определенной функцией уплотнения (на изображении — $1 - maxpooling$), т.е. уменьшает размерность сформированной карты признаков. Таким образом извлекается наиболее важная информация для каждой свертки независимо от её положения в тексте. Другими словами, для используемого векторного отображения комбинация слоев свёртки и слоев субдискретизации позволяет извлекать из текста наиболее значимые n-граммы.

После этого карты признаков, рассчитанные на выходе каждого слоя субдискретизации, объединяются в один общий вектор признаков. Он подается на вход скрытому полносвязному слою, а потом поступает на выходной слой нейронной сети, где и рассчитываются итоговые метки классов.

Система определения тональности текста была написана на языке программирования Python с использованием библиотеки Keras.

Python - высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объем полезных функций.

Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою оче-

редь, могут объединяться в пакеты.

Keras – открытая нейросетевая библиотека, написанная на языке Python. Она представляет собой надстройку над фреймворками Deeplearning4j, TensorFlow и Theano. Нацелена на оперативную работу с сетями глубинного обучения, при этом спроектирована так, чтобы быть компактной, модульной и расширяемой.

Эта библиотека содержит многочисленные реализации широко применяемых строительных блоков нейронных сетей, таких как слои, целевые и передаточные функции, оптимизаторы, и множество инструментов для упрощения работы с изображениями и текстом.

Keras является второй по скорости роста системой глубокого обучения после TensorFlow Google, и третьей по размеру после TensorFlow и Caffe. Большая часть библиотек из представленных не поддерживают работу на операционной системе Windows, а значит мы не сможем использовать их для разработки нашего программного обеспечения, так как разработку планируется выполнять на Windows.

Наибольший интерес представляет библиотека Keras, так как она является кроссплатформенной и обладает большим потенциалом развития. Благодаря тому, что данная библиотека использует в своей работе TensorFlow, мы получаем все преимущества TensorFlow и в дополнение более удобный интерфейс разработки при создании нейронной сети.

ЗАКЛЮЧЕНИЕ

Обработка естественного языка (Natural Language Processing, NLP) — общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков. Применительно к искусственному интеллекту анализ означает понимание языка, а синтез — генерацию грамотного текста. Решение этих проблем будет означать создание более удобной формы взаимодействия компьютера и человека.

Целью данной работы является изучение методов решения задачи автоматического анализа тональности текста, выбор инструментария, позволяющего реализовать систему автоматического анализа тональности текста, отвечающую следующим требованиям и реализация данной системы:

- Система должна определять тональность небольших текстовых высказываний (формат комментариев)
- Точность определения тональности текста должна быть не ниже 70%
- Определение тональности текста должно происходить в большинстве популярных предметных областей
- Проектируемая система должна определять тональность предложений с новыми словами или с использованием новых конструкций

В рамках данной работы была реализована система определения тональности короткого высказывания с помощью сверточных нейронных сетей. Реализованная система была обучена на большой выборке рускоязычных текстов из популярной системы Twitter. Тестирование показало высокую точность работы системы (85%).