Московский авиационный институт
(Национальный исследовательский университет)

Факультет прикладной математики и информационных технологий
Кафедра математической кибернетики

Курсовой проект

Реализация алгоритма определения тональности текста англоязычных твитов

(с использованием языка программирования Python)

Выполнили
студенты гр. 8О-105М:
А. П. Иванов
В. Д. Ревина
В. И. Руснак

Проверил
преподаватель:
В. Н. Пановский

Москва, 2019 г.

Содержание

1. Введение

Анализ тональности текста или сентимент-анализ (sentiment analysis) - это область компьютерной лингвистики, занимающаяся выделением из текстов эмоционально окрашенной лексики или эмоциональной оценки автора.

Рассматриваемая концепция как теоритически, так и пратически, может использоваться во многих областях, рассмотрим некоторые из них.

Во-первых, анализ тональности текстов способен помочь разобраться в законах, по которым живет естественный язык и научить компьютер воспринимать его на уровне, приближенном к человеческому. До недавнего времени машина понимала тексты на абстрактном уровне – в основном, через лексемы (слова), которые для нее обладали формой (набор букв) и содержанием (значение). Данная концепция предлагает ввести еще одну функцию – так называемую лексическую тональность текста (в простейшем случае она будет определяться как сумма лексических тональностей каждой отдельной лексемы).

Во-вторых, анализ тональности способен значительно повысить качество машинного перевода. Известно, что эталоном машинного перевода служит результат перевода текста человеком – профессиональным переводчиком. За 50 с лишним лет разработок в этой области исследователи убедились в том, что научить машину «думать, как переводчик» можно лишь приняв во внимание все те соображения, которыми пользуется профессионал, переводя тот или иной текст. Естественно, при переводе не обойтись без первичного анализа текста и отдельных слов – в том числе, анализа тональности как таковой.

В-третьих, целью анализа тональности текста может быть некое мнение автора или сам автор. Это – наиболее интересная сфера применения, поскольку здесь видится не только способ делегирования машине некоторых полномочий ученого (например, филолога, который исследует произведение того или иного автора), но и снова попытка приблизить образ мышления компьютера к человеческому. С этой точки зрения анализ тональности, возможно, является одним из самых важных и перспективных шагов к развитию искусственного интеллекта.

В том числе, если говорить о сентимент-анализе англоязычных твитов популярной социальной сети, собранная и проанализированная информация может использоваться для составления статистики, некого обзора, мнений людей в виде диаграммы в той или иной рассматриваемой сфере жизни (например, в сфере обслуживания - кино, рестораны и т.д.). Каждый положительный, негативный или нейтральный отзыв поможет охарактеризовать исследуемый объект коммерческой или независимой компанией в целом и принять некоторое решение, позволяющее улучшить способ и качество обслуживания в данной сфере исследоваемого объекта.

2. Постановка задачи

Реализовать алгоритм определения тональности англоязычных твитов с использованием языка программирования Python с ожидаемой точностью не менее 80%. Провести исследование-обзор, а также сравнительный анализ, сделать выводы, получить дальнейшие рекомендации в случае успешной реализации поставленной задачи.

3. Обзор существующих методов

Существует несколько методов, на базе которых существуют решения задачи определения тональности текстов. Рассмотрим некоторые из них.

В методе, основанном на словарях с учетом отрицаний, составлены словари положительных и отрицательных слов. Для их заполнения был использован переведенный список 6800 слов обоих категорий списка на английском [1]. Также словари были вручную скорректированы. При анализе в тексте подсчитывается количество положительных и отрицательных слов, также учитывается частица «не» перед словом. Если эта частица встречается, то слово приобретает противоположную эмоциональную окраску. Перед обработкой также стираются предлоги и союзы как неинформативные. В методе используется стемминг - отсечение от слова окончаний и суффиксов, чтобы оставшаяся часть слова "stem"была одинаковой для всех грамматических форм слова [2].

Метод с использованием библиотеки анализа тональности Стэнфорда и API-сервиса переводов Яндекса представляет собой использование библиотеки обработки естественного языка для английского, включающей в себя теоретико-графовый метод, разработанной Стэнфордом и Yandex Translate API для предварительного перевода текста. В результате проверки были получены следующие результаты: тональность была определена лучше методом, основанном на словарях — оценка была определена правильно для 71 % отзывов. Вторым методом тональность текста была определена для 56 % отзывов.

Конечно, все рассмотренные выше методы и решения не являются идеальными и всегда существуют способы улучшить их. Ошибки данных методов объясняются следующими проблемами:

- многочисленные орфографические ошибки в отзывах;
- нет связи с объектов: тональность определяется для всего текста;
- не всегда об отношении автора можно сказать по наличию или отсутствию положительных, отрицательных или нейтральных отзывов.

Улучшить результаты автоматического определения тональности текста возможно при помощи использования методов автоматического исправления орфографических ошибок, совершенствования словарей (для методов, основанных на словарях) и обучающей выборки (для методов машинного обучения). Также возможно повысить точность работы алгоритмов, применяя разработки по другим проблемам естественного языка, таких как:

- автоматическое реферирование[1],

[1] Автоматическое реферирование (Automatic Text Summarization) - это извлечение наиболее важных сведений из одного или нескольких документов и генерация на их основе лаконичных отчетов.

- выявление кореферентности[1], референционального множества[2],
- анализ сравнений,
- извлечение объектов из текстов и выявление отношений между ними.

Рассмотрим некоторые готовые программные продукты, используемые для определения эмоциональной окраски анализируемого текста.

Веб-сервис Twitter Sentiment [3] позволяет анализировать информацию об объекте, который упоминают пользователи, при помощи данных из Twitter. Пользователю этого сервиса достаточно ввести слово, и программа проанализирует последние 100 записей-упоминаний об этом слове,при этом будет построен график соотношения положительных и отрицательных отзывов. Twitter Sentiment использует метод машинного обучения, а так же имеет API[3].

В программном продукте I-Teco [4] пользователь вводит интересущий его текст и веб-сервис на основе использования метрики и специальных словарей эмоционально окрашенной лексики выдает результат в виде окрашенного текста: красный, зеленый и черный, которые означают, что текст негативный, положительный или нейтральный соответственно.

Программный продукт Sentiment Analysis with Python NLTK Text Classification имеет веб-интерфейс [5] и использует метод машинного обучения с использованием наивного байесового классификатора [6].

Рассмотрение способов определения эмоциональной окраски текстов показало, что необходимо анализировать связи между словами для корректного определения тональности. Невозможно проанализировать отзыв, игнорируя особенности структуры языка, на котором анализируется комментарий, т. е. необходимы или выявление отношений между объектами предложений или анализ синтаксической структуры предложения. Для данных задач разработано большое количество методов и готовых программных средств для английского языка. Для русского языка ситуация обстоит сложнее - для него готовых решений меньше, и многие из них являются коммерческими.

---

[1] Кореферентность - это «отношения между элементами высказывания, которые обозначают один и тот же внеязыковой объект». Пр.: *Вася себя ценит*, слово *себя* означает самого Васю.

[2] Референциальное множество - это множество, которое соотносит языковые выражения с объектами и ситуациями внешнего мира. Пр.: *Врач осмотрел его и поставил диагноз: «Голова правильной формы, облысение идет нормально».* В этом предложении именная группа "врач"соотносится с неким объектом внешнего мира – в данном случае с определенным, индивидуализированным, объектом.

[3] API (Application Programming Interface) - описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

4. Описание алгоритма

Прежде, чем приступить к основной части алгоритма определения тональности текста, необходимо назначить ту выборку, которую будет анализировать алгоритм. Осуществить это можно с помощью предварительно полученного токена[1], необходимого для доступа к API. Далее с помощью API получаем выборку твитов интересующей нас категории.

Итак, рассмотрим этапы алгоритма автоматического определения тональности англоязычных твитов.

1) **Предобработка**. На первом этапе из выборки удаляются все html, тэги, пунктуация, числа, символы, артикли (a, the и т.д.), хэштеги. Данная операция осуществляется с помощью библиотеки python — «Beautiful Soup». Артикли и предлоги удаляются с помощью пакета Python Natural Language Toolkit (NLTK). После предобработки получаем набор слов, которые необходимо нормализовать, т.е. провести операцию стэмминга: убрать все окончания слов, префиксы, оставив только основную корневую часть слова.

2) **Представление набора слов в виде вектора**. Следующим этапом необходимо заменить каждое слово в наборе номером его семантической группы. В итоге мы получим нечто вроде «мешка слов» («Bag of words»). Для этого используется технология Word2Vec от Google. Найти «Bag of words» можно в пакете библиотеки gensim, со встроенными моделями Word2Vec.

   Суть модели Word2Vec заключается в следующем: на вход дается большой объем текста (например, 10000 отзывов) и, «обучаясь» на входных текстовых данных, создается словарь, затем вычисляется 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. Таким образом, на выходе получаем взвешенный вектор для каждого слова фиксированный длины (длина вектора задается вручную), которая встречается в нашем наборе отзывов. Например, для слова men, сравнивая со всеми словами и сортируя в убывающем порядке получается результат, приведенный в Таблице 1 на стр. 7 (за меру близости выбрано косинусное расстояние).

---

[1] Токен - объект, создающийся из лексемы в процессе лексического анализа («токенизации», от англ. tokenizing). Токены широко применяются в системах авторизации/идентификации и технически обычно реализуются в виде записи в БД, где токен является идентификатором записи о данных пользователя или предоставленного доступа.

Words	Measures
`woman`	0,6056
`guy`	0,4935
`boy`	0,4893
`men`	0,4632
`person`	0,4574
`lady`	0,4487
`himself`	0,4288
`girl`	0,4166
`his`	0,3853
`he`	0,3829

Таблица 1. Семантический близкие слова к слову «man»

3) **Кластеризация**. Далее для объединения близких по смыслу слов используется алгоритм автоматической кластеризации - DBScan (Density-based spatial clustering of applications with noise). Это алгоритм кластеризации основан на плотности — если дан набор точек в некотором пространстве, алгоритм группирует вместе точки, которые тесно расположены (точки со многими близкими соседями), а точки, которые находятся одиноко в областях с малой плотностью (ближайшие соседи которых лежат далеко) помечаются как выбросы.

   Итак, опишем формально, как работает данный алгоритм. Пусть задана некоторая симметричная функция расстояния $\rho(x, y)$ и константы ϵ и m Тогда:

   а) Назовём область *E(x)*, для которой $\forall y : \rho(x, y) \leq \epsilon$, ϵ - окрестность объекта x.

   б) Корневым объектом или ядерным объектом степени m называется объект, ϵ - окрестность которого содержит не менее m объектов: *E(x)* $\geq m$.

   в) Объект p непосредственно плотно-достижим из объекта q, если $p \in$ *E(q)* и q — корневой объект.

   г) Объект p плотно-достижим из объекта q, если $\exists\, p_1, p_2, \ldots, p_n, p_1 = q$, $p_n = p$, такие что $\forall i\ \exists 1 \ldots n-1 : p_{i+1}$ непосредственно плотно-достижим из p_i.

   Выберем какой-нибудь корневой объект из набора данных, пометим его и поместим всех его непосредственно плотно-достижимых соседей в список обхода. Теперь для каждой из списка: пометим эту точку, и, если она тоже корневая, добавим всех её соседей в список обхода. Тривиально доказывается, что кластеры помеченных точек, сформированные в

ходе этого алгоритма максимальны (т.е. их нельзя расширить ещё одной точкой, чтобы удовлетворялись условия) и связны в смысле плотностидостижимости. Отсюда следует, что если мы обошли не все точки, можно перезапустить обход из какого-нибудь другого корневого объекта, и новый кластер не поглотит предыдущий.

4) **Классификация**. Последним этапом алгоритма является использование метода классификаций Random Forest, который используется для классификаций твитов в этой работе. Алгоритм уже реализован в пакете scikit-learn, и все, что остается сделать это предоставить текстовые данные и указать количество деревьев. Дальше алгоритм все берет на себя, тренируется на обучающей выборке, сохраняет все необходимые данные.

По окончанию эксперимента алгоритм будет выдавать результат с % - определением точности эмоциональной окраски текста в зависимости от его фактического содержания: положительный ли отзыв, отрицательный или нейтральный.

5. Некоторые пояснения к коду

TF-IDF

TF-IDF — статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции.

TF (term frequency — частота слова) — отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, оценивается важность слова в пределах отдельного документа.

DF (inverse document frequency — обратная частота документа) — инверсия частоты, с которой некоторое слово встречается в документах коллекции. Учёт IDF уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF.

Мера TF-IDF часто используется в задачах анализа текстов и информационного поиска, например, как один из критериев релевантности документа поисковому запросу, при расчёте меры близости документов при кластеризации.

Scikit-learn

Scikit-learn это библиотека для машинного обучения на языке программирования Python с открытым исходным кодом. С помощью нее можно реализовать различные алгоритмы классификации, регрессии и кластеризации, в том числе алгоритмы SVM, случайного леса, k-ближайших соседей и DBSCAN, которые построены на взаимодействии библиотек NumPy и SciPy с Python.

ROC-AUC

Площадь под ROC-кривой – один из самых популярных функционалов качества в задачах бинарной классификации. Часто результат работы алгоритма на фиксированной тестовой выборке визуализируют с помощью ROC-кривой (ROC = receiver operating characteristic, иногда говорят «кривая ошибок»), а качество оценивают как площадь под этой кривой – AUC (AUC = area under the curve).

RandomForest

RF (random forest) — это множество решающих деревьев. В задаче регрессии их ответы усредняются, в задаче классификации принимается решение голосованием по большинству. Все деревья строятся независимо по следующей схеме:

Выбирается подвыборка обучающей выборки размера samplesize (м.б. с возвращением) – по ней строится дерево (для каждого дерева — своя подвыборка). Для построения каждого расщепления в дереве просматриваем max_features случайных признаков (для каждого нового расщепления — свои случайные

признаки). Выбираем наилучшие признак и расщепление по нему (по заранее заданному критерию). Дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса), но в современных реализациях есть параметры, которые ограничивают высоту дерева, число объектов в листьях и число объектов в подвыборке, при котором проводится расщепление. Понятно, что такая схема построения соответствует главному принципу ансамблирования (построению алгоритма машинного обучения на базе нескольких, в данном случае решающих деревьев): базовые алгоритмы должны быть хорошими и разнообразными (поэтому каждое дерево строится на своей обучающей выборке и при выборе расщеплений есть элемент случайности).

F-мера

F-мера представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю.

F-мера является хорошим кандидатом на формальную метрику оценки качества классификатора. Она сводит к одному числу две других основополагающих метрики: точность и полноту. Имея в своем распоряжении подобный механизм оценки будет гораздо проще принять решение о том являются ли изменения в алгоритме в лучшую сторону или нет.

Метод k-средних

Метод k-средних (англ. k-means) — наиболее популярный метод кластеризации. Действие алгоритма таково, что он стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров.

По аналогии с методом главных компонент центры кластеров называются также главными точками, а сам метод называется методом главных точек и включается в общую теорию главных объектов, обеспечивающих наилучшую аппроксимацию данных.

6. Вывод

Проведена работа по исследованию тональности текстов. Получен результат f_1 score = 0.68. Использование данного метода не считается оптимальным, так как baseline показал более высокий процент f_1 score = 0.74

Список литературы

[1] Hu and Liu, A list of positive and negative opinion words or sentiment words for English, http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html#lexicon

[2] Стеммер, http://www.solarix.ru/for_developers/api/stemmer.shtml

[3] Веб-сервис TwitterSentiment: http://www.sentiment140.com

[4] Веб-сервис I-Teco: http://x-file.su/tm/Default.aspx

[5] Веб-сервис Sentiment Analysis with Python NLTK Text Classification, http://text-processing.com/demo/sentiment/

[6] Наивный байесовский классификатор, http://ru.wikipedia.org/wiki/Наивный_байесовский_классификатор