Автоматическое определение тональности текста

Бегалиев Р.А.

Москва 2019 г

Содержание

Введение

Мнения людей имеют огромную ценность, например, правительство может оценить уровень доверия к президенту, реакцию населения на новый закон, компании, производящие различные продукты могут узнать какую оценку дают покупатели данному продукту и что можно улучшить и т.д. В современном мире большую популярность приобрели социальные сети и люди без особых затрат имеют возможность выразить свое мнение о чем-либо (товары, услуги, персоны, …). В связи с этим появилась необходимость в анализе текстов, который можно реализовать несколькими способами. Допустим, компания «X» хочет узнать мнение людей о ее новом товаре и сразу реагировать на сообщения людей. Для решения данной задачи «X» может нанять группу людей, которая будет анализировать мнения в одной из социальных сетей, например, Twitter. Очевидно, что этот вариант нереален так как в каждую секунду будет появляться много сообщений и наша команда не будет успевать анализировать их со скоростью даже близкой к скорости их появления. Следовательно, появляется необходимость в автоматической обработке текстов.

В каждом тексте интерес вызывают следующие пункты:

- Автор текста
- Дата публикации
- Тональность (положительная, отрицательная, нейтральная)

В данной работе представлены результаты автоматического определения тональности текста. Также стоит отметить, что классификация является бинарной, то есть рассматриваются лишь тексты с положительной и отрицательной тональностью. Для исследования использовалась коллекция текстов русскоязычного твиттера [1].

Выбор модели и оценка качества

Для решения поставленной задачи определения тональности текста используется машинное обучение с учителем (наивный байесовский классификатор, линейный классификатор, нейронная сеть), которое показывает более точные результаты классификации нежели методы машинного обучения без учителя или обучение на основе правил и словарей [2]. Для применения методов с учителем необходимы размеченные данные, благо они имеются в наличии [1].

Следует выделить два типа размеченных данных:

- Тренировочные и тестовые данные включают тексты из определенной предметной области
- Тренировочные и тестовые данные включают тексты из различных предметных областей

Данные относящие к первому типу содержать только тексты одной из предметной области, например, отзывы о фильмах, коллекцию новостей и т.п. Данные второго типа могут содержать как рецензии к фильмам, так и коллекцию новостей или отзывы о товаре. В данной работе исследование будет производиться на данных второго типа.

Оценка качества результатов классификации будет производиться по точности: accuracy.

Подготовка данных

Подготовка данных является одним из условий успешно построенной модели машинного обучения. Для обучения классификатора необходимо представить данные в качестве вектора признаков. N-граммы – последовательность N символов или слов. Пополнение коллекции новыми сообщениями приведет к добавлению небольшого числа терминов [2]. Используемая в данной работе коллекция состоит из 111 923 негативных и 114 991 положительных сообщений. Проведем следующие операции с данными:

- Удалим все английские буквы
- Удалим из твитов имена пользователей, ссылки и метки о ретвите
- Приведем все слова к нижнему регистру
- Удалим всю пунктуацию

Постановка задачи

Задача классификации представляется в следующем виде: пусть d-описание документа из векторного пространства документов X, C = {c1, c2}, где c1 и c2 положительный и отрицательный класс соответственно. Необходимо построить классифицирующую функцию F(D)=r, которая отображает пространство документов в классы, где D = {<d, c> | <d, c> є X x C}. Документы из обучающей и тестовой выборок есть k-мерные векторы

признаков, то есть d=(w1,w2, .. , wv), где V- множество всех уникальных униграмм из обучающей выборки [3].

Мы сталкиваемся со словами, которые часто встречаются в текстах обоих классов и не содержат полезной или отличительной информации. В этой работе используются метод tf-idf, который может использоваться для понижающего взвешивания этих часто встречающихся слов в векторах признаков:

$$tfidf(t,d) = tf(t,d) * idf(t,d)$$

Здесь tf(t, d) – это частота термина, idf(t, d) – обратная частота термина.

$$idf(t,d) = \log(n_d/(1 + df(d,t)))$$

где n_d- общее число документов, df(d, t) – число документов d, содержащих термин t. В библиотеке scikit-learn для idf и tfidf реализованы несколько другие уравнения [4]:

$$idf(t,d) = \log((n_d + 1)/(1 + df(d,t)))$$

$$tfidf(t,d) = tf(t,d) * (idf(t,d) - 1)$$

Алгоритм

1. Объединяем положительные и отрицательные отзывы в один объект DataFrame библиотеки pandas.
2. Выделяем признаки с которыми дальше будем работать (текст, метка).
3. Обрабатываем получившиеся данные (приводим к нижнему регистру, удаляем все буквы кроме русских, удаляем пунктуацию, ссылки, метки о ретвите, имя пользователя).
4. Проводим векторизацию (Count Vectorizer, TfIdfVectorizer).
5. Выбираем модели, которые будем использовать (Наивный Байесовский классификатор, линейный классификатор SGD).
6. Кросс валидация Наивного Байесовского классификатора (размер текстовой выборки 0.1).
7. Кросс валидация линейного классификатора SGD (размер текстовой выборки 0.1) с сеточным поиском.
8. Сравнение и выбор лучшей модели

Итоговые результаты

Лучший результат показал метод линейный классификатор SGDClassifier() с регуляризацией L2, alpha=1e-6, функция потери log, векторизация tf-idf. Верность классификации составил 0.76 с использованием униграммы+биграммы+триграммы.

Литература

1. http://study.mokoron.com/
2. «Разработка и исследование предметно независимого классификатора текстов по тональности» - Ю.В. Рубцова
3. Introduction to information Retrieval - Manning D., Raghavan P., Shutze H.
4. Python and machine learning - Sebastyan R.