Московский Государственный Университет
им. М.В. Ломоносова

Классификатор отзывов об отелях

Отчёт по спецкурсу “Математические методы анализа текстов”

Студент: Астабацян Каро
Группа: 324
Факультет: ВМК
Преподаватель:
Dr Мстислав Масленников

Москва, май 2013г.

Мотивация

Целью нашей работы являлось создание программы (далее классификатора), способной определять тональность, эмоциональный окрас текста (в нашем случае это отзывы об отелях).

В этой работе мы исследуем проблему, какое количество слов-униграмм нужно использовать для определения эмоционального окраса текста.

Наша гипотеза состоит в том, что для определения эмоционального окраса текста достаточно использовать менее 1% слов из отзывов («отлично», «ужасно», «супер», «превосходно»). В качестве главного алгоритма был выбран метод опорных векторов. В связи с этим можно выделить несколько этапов нашей работы:

1. ***Подбор корпуса текстов отзывов***
2. ***Выделение характеристик (признаков)***
3. ***Написание самой программы и ее тестирование***

Отбор свойств

Мы заметили, что многие слова, например «отлично», «ужасно», «супер», «превосходно», часто встречаются в текстах одной группы отзывов и редко встречаются в текстах другой группы. Поэтому мы пытаемся сравнить распределения этих слов в положительно и отрицательно окрашенных текстах. Сначала все слова в текстах отзывов приводятся в начальную форму при помощи программы MyStem, разработанной компанией Yandex. Затем мы отбираем свойства на

основе этих слов в алгоритме FindFeatures:

FindFrequentWord(PosTexts, NegTexts)

- Пусть $W=\{w_1, \ldots, w_N\}$ – множество слов в PosTexts, NegTexts
- Пусть $freq_i^+$ - частотность w_i в PosTexts, $freq_i^-$ - частотность w_i в NegTexts
- $freq_i = freq_i^+ - freq_i^-$
- Возьмём $W_{res}=\{w_i \mid freq_i^+ \geq \theta, freq_i^- \geq \theta, freq_i \geq \theta_{rel}, \}$, $\theta = 20$, $\theta_{rel} = 150$

Return W_{res}

Алгоритм FindFeatures составляет частотный словарь всех слов, т.е. для каждого слова встретившегося в тексте подсчитывается частота его употребления. Далее частотные слова положительных и отрицательных отзывов PosTexts, NegTexts сливаются по следующему правилу: в результирующее множество слов W_{res} входят те слова, которые встретились в обучающей выборке не менее некоторого наперед заданного числа раз $\theta = 20$ и разность частот употребления в положительных и отрицательных отзывах $\theta_{rel} = 150$. Таким образом, было получено множество слов, которое использовалось нами в качестве свойств.

FindFeatures(W_{res}, Document)

- Пусть $count_i$ - частотность w_i в Document
- Пусть count – кол-во слов в Document

- $feat_i = count_i / count$

Return $feat = (feat_1, \ldots, feat_N)$

В алгоритме FindFeatures в качестве вектора признаков feat для каждого отзыва Document используется n-мерный вектор действительных чисел от 0 до 1, где n – количество слов во множестве слов-характеристик, а i-ое число в векторе - отношение частоты употребления i-ого слова во множестве свойств к количеству слов в отзыве. Таким образом, каждый отзыв кодируется n-мерным вектором признаков.

Классификатор

Программа классификатора написана на языке Java и использует реализацию SVM на этом же языке в виде библиотеки SVMLight. При обучении программа поочередно генерирует для каждого отзыва из корпуса для обучения вектор чисел от 0 до 1 и обучается согласно реализованному в нем алгоритму. При тестировании программа генерирует вектор признаков для отзыва и относит его к положительным или отрицательным.

Эксперименты

В этой части мы описываем создание корпуса текстов, постановку эксперимента, полученные результаты и выводы.

Создание корпуса текстов отзывов

Итак, первым этапом нашей работы стал подбор обучающей и тестовой выборок – текстов, на которых программа будет обучаться и тестироваться. Источником таких текстов стал сайт www.tophotels.ru, на котором уже на протяжении многих лет пользователи оставляют отзывы об отелях, в которых они побывали, высказывают свои мнения по поводу обслуживания в отелях, комфорта, удобств и т.п. Каждый отзыв сопровождается некоторой оценкой от 0 до 5, которая отражает как раз таки общее настроение в тексте отзыва. Таким образом, отфильтровав отзывы с оценками выше и ниже некоторых пороговых значений и отнеся их к положительным или отрицательным отзывам, мы могли бы собрать отдельно ярко выраженные положительные и отрицательные отзывы. Для решения описанной задачи была написана программа на языках PHP и JavaScript, которая выполнила все вышеперечисленные действия по подбору отзывов и собрала более 40тыс текстов.

Постановка эксперимента

Целью экспериментов была проверка главной гипотезы, что использование менее 1% слов достаточно для определения эмоционального окраса текста.

Признаком качества программы является точность, т.е. то насколько хорошо она справляется с задачей определения настроения в тексте, а в нашем случае с задачей классификации отзывов. Точность мы рассчитывали по следующей формуле:

$$точность = \frac{количество \text{ попаданий}}{общее\, количество\, тестов},$$

где “попадания" – это те тесты, которые были правильно соотнесены с заранее известными оценками, данными пользователями в виде числовых оценок.

Для обучения мы использовали 30000 положительных и столько же отрицательных отзывов. Для тестирования - по 7000 положительных и отрицательных отзывов.

Результаты

При тестировании программы точность на тестовой выборке была равна 0,84, что говорит о том, что в среднем в 84 случаях из 100 программа правильно определяет общее настроение в тексте отзыва.

Процент слов в качестве свойств	Точность классификатора
30	0.54
10	0.62
5	0.72
1	0.84
0.01	0.7

Выводы

Только 1% слов влияет на определение эмоциональной окраски текста. Использование большего количества текстов не приводит к улучшению результата.