Анализ тональности русскоязычного текста

В. В. Осокин, М. В. Шегай

Задача анализа тональности играет важную роль в обработке естественного языка. Рассматривается задача классификации русскоязычного текста на два класса в зависимости от его эмоциональной окраски: положительный и отрицательный. В качестве классификатора используется наивный байесовский классификатор. Используются различные методы для отбора признаков, производится сравнение полученных результатов с результатами классификации англоязычного текста. Достигнута точность 78.4% на заданном тестовом наборе данных.

Ключевые слова: анализ тональности, наивный байесовский классификатор, выбор признаков.

Введение

В настоящее время наблюдается интенсивный рост сети Интернет. С увеличением пользователей возрастает количество генерируемого ими контента. Одно из направлений искусственного интеллекта, *обработка естественного языка (NLP — Natural Language Processing)*, позволяет вычислительным системам извлекать различную информацую из естественного языка.

Важной задачей обработки естественного языка является задача определения *тональности текста*. Задачей определения тональности текста является извлечение авторской эмоциональной оценки, выраженной в тексте.

Анализ тональности можно применять в различных областях, например в социологии (для определения отношения пользователей соц. сетей к тем или иным событиям), маркетинге для определения отношения покупателей к тем или иным продуктам, психологии (для определения депрессиии у пользователей соц. сетей).

Существует большое количество работ, посвящённых обработке естественного языка и, в частности, анализу тональности. Но большая часть из них адаптирована для применения к английскому языку.

В данной работе рассматривается задача классификации русскоязычного текста на два класса (бинарной классификации): текст, несущий в себе положительную оценку и текст, несущий в себе отрицательную оценку.

Постановка задачи и полученные результаты

Назовём *признаком* отображение $f : X \to D_f$, где D_f — множество допустимых значений признака. В данной работе рассматривается случай, когда признаками являются слова и словосочетания.

Если заданы признаки $f_1, \dots, f_n$, то вектор $x = (f_1(x), \dots, f_n(x))$ называется признаковым описанием объекта $x \in X$.

Признаковые описания допустимо отождествлять с самими объектами. При этом множество $X = D_{f_1} \times \cdots \times D_{f_n}$ называют *признаковым пространством*.

Пусть X — множество описаний объектов, $Y = \{0, 1\}$ — множество номеров (или наименований) классов. Существует неизвестная целевая зависимость — отображение $y^* \colon X \to Y$, значения которой известны только на объектах конечной обучающей выборки $X^m = \{x_1, \dots, x_m\}$. Требуется построить алгоритм $a \colon X \to Y$, способный классифицировать произвольный объект $x \in X$.

В рамках поставленной задачи был построен классификатор с точностью около 78.4%.

Результаты проверялись эмпирическим путём. Непосредственно сам классификатор был написан на языке java, а модули для предварительной обработки данных на ruby и на python.

Исходный код классификатора и скриптов для предварительной обработки текста можно найти по адресу https://github.com/ncanter/sentiment.

Данные

В качестве данных используются публично доступные отзывы к фильмам, взятые с сайта http://imhonet.ru/.

На сайте http://imhonet.ru/ используется десятибальная шкала оценки, которую пользователь сам выставляет во время написания отзыва.

В работе используется следующее допущение: все отзывы, имеющие оценку не превосходящую 5, классифицируются, как отрицательные, а отзывы, имеющие оценки от 6 до 10, классифицируются, как положительные.

Всего использовалось 140000 отзывов.

Методика тестирования

Обучающая и тестовые выборки содержат одинаковое число отзывов (по 70000). При этом, как обучающая так и тестовые выборки содержат одинаковое число положительных и отрицательных отзывов.

Точностью распознавания будем считать величину[1]

$$acc = \frac{tr}{|D|},$$

tr — количество правильно распознаных отзывов, $|D|$ — общее количество отзывов.

Для уменьшения влияния явления *переобучения* при тестировании использовалась методика, сходная со *скользящим контролем* (cross-validation).

Пусть у нас есть 2 множества: P и N. P содержит положительные отзывы. N содержит отрицательные отзывы.

Занумеруем все элементы каждого из множеств числами от 1 до k ($k = |P| = |N|$).

Тестирование будем проводить в t итераций ($t = \frac{k}{m}$; $1 \leqslant m \leqslant k$).

[1]Часто для оценки качества классификации используют меру f_1, которая учитывает ошибки первого и второго рода, в нашем случае это излишне, в связи с тем, что в тестовой выборке количество положительных и отрицательных отзывов одинаково

На каждой итерации тестовая и обучающая выборки делятся на два непересекающихся множества.

На i-й итерации будем брать элементы из каждого из множеств с номерами $\overline{mi \dots mi+n}$, если $mi+n \leqslant k$ и номерами $\overline{mi \dots k} \cup \overline{1 \dots (mi+n) \mod k}$, если $(mi+n > k)$. Эти элементы образуют обучающую выборку, остальные элементы образуют тестовую выборку.

На каждой итерации будем вычислять значение acc_i.

Тестирование проводилось при значениях $k = 70000$, $m = 7000$, $n = 35000$.

Точностью классификатора A (в дальнейшем просто *точностью*) назовём величину равную среднеарифметическому acc_i.

$$A = \frac{\sum_{i=1}^{t} acc_i}{t}.$$

Наивный байесовский классификатор

Воспользуемся теоремой Байеса

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)},$$

$P(c|d)$ — апостериорная вероятность принадлежности к классу c для отзыва d, $P(c)$ — априорная вероятность класса c, $P(d)$ — вероятность появления отзыва d, $P(d|c)$ — вероятность появления отзыва d в классе c.

$$C_{MAP} = \arg\max_{c \in C} P(c|d) = \arg\max_{c \in C} \frac{P(d|c)P(c)}{P(d)},$$

C_{MAP} — максимальная апостериорная вероятность класса.

Заметим, что в последнем выражении можно избавиться от знаменателя, так как ищется максимум по всем классам c для фиксированного отзыва d. Имеем:

$$C_{MAP} = \arg\max_{c \in C} P(d|c)P(c).$$

Здесь под вероятностью $P(d/c)$ появления отзыва d в классе c подразумевается вероятность повяления вектора признаков $x_1, \dots, x_n$ в данном классе. Тогда:

$$C_{MAP} = \arg\max_{c \in C} P(x_1, \ldots, x_n|c)P(c).$$

Вероятность $P(x_1, \ldots, x_n|c)$ может быть оценена только при наличии очень большой обучающей выборки и её вычисление требует больших вычислительных мощностей.

В связи с этим воспользуемся двумя упрощениями:

1) Условная независимость слов (наивное предположение) — предположим, что разные слова в тексте появляются независимо друг от друга
2) «Мешок слов» (Bag of words) — предположим, что взаиморасположение слов не имеет значения

Тогда получим:

$$C_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{i \in positions} P(x_i|c_j),$$

positions — всевозможные позиции, на которых появляются слова.

Введём функцию $\varphi(w, c)$, которая будет показывать, сколько раз слово w встречается в классе c.

Параметры для классификатора будем вычислять следующим образом:

$$\widehat{P}(w_i|c) = \frac{\varphi(w_i, c) + 1}{\sum\limits_{w \in V} (\varphi(w, c) + 1)} = \frac{\varphi(w_i, c) + 1}{\sum\limits_{w \in V} \varphi(w, c) + |V|}.$$

Добавление единицы имеет название сглаживания Лапласа (add-one smoothing), это позволяет получать ненулевые вероятности для слов, которые встречаются впервые, тем самым, не обращая всё произведение в ноль.

Описанную выше модель будем использовать в качестве классификатора, которую в дальнейшем будем обозначать NB.

В английском языке для решения задачи анализа тональности бывает удобно использовать наивный байесовский классификатор Бернулли (Bernoulli naive Bayes classifier) [2]. Основное отличие заключается в том, что для данного документа рассматривается не количество вхождений слова, а только их наличие или отсутствие.

Эмпирически было установленно, что применение модели Бернулли для отзывов на русском языке сильно снижает точность.

Предварительная обработка текста

В работе была произведена предварительная обработка текста, которую можно разделить на 3 этапа:

1) токенизация — это процесс выделения из текста отдельных слов, чисел и занков пунктуации;
2) стемминг[2] — это процесс нахождения основы слова для заданного исходного слова;
3) обработка отрицаний.

Цель стемминга — приведение слов, имеющих одинаковую основу к единой форме (так же за счёт этого происходит уменьшение размерности задачи). После стемминга теряется часть морфологической информации, поэтому, как показали результаты исследований, применение стемминга при использвании NB для анализа тональности текста на русском языке, как и в английском языке, не увеличивает точность.

Токенизация производится на основе регулярного выражения.

Обработка отрицаний

Для увеличения точности был использован алгоритм обработки отрицаний, описанный Десом и Ченом [4]. Суть его заключается в следующем: при появлении частицы «не» к началу каджого слова между этой частицой и последующим знаком препинания либо другой частицой «не» приписывается приставка «not_»[3].

Пример. Предложение:

«Мне не понравился этот фильм.»

Преобразуется к виду:

«Мне не not_понравился not_этот not_фильм.»

Обработка отрицаний позволила улучшить точность, но на значительно меньшую величину, по сравнению с англоязычными тесктами [5].

[2]Для стемминга используется алгоритм, подробное описание которого можно найти по ссылке http://snowball.tartarus.org/algorithms/russian/stemmer.html [3]

[3]Использование слов, в которых частица «не» пишется слитно с отсеиванием слов, которые без «не» не употребляются снизило точность.

Выбор признаков

В качестве признаков в работе используются так называемые *n-граммы*.

N-грамма — последовательность из n элементов, в данном случае последовательность из n слов.

Для последовательностей, содержащих менее 4 элементов, приняты специальные обозначения. Последовательность длины один называется *униграмма*, два — *биграмма*, три — *триграмма*.

Было установленно, что наиболее эффективным является применение комбинации униграмм и биграмм. Использование триграмм заметно ухудшает точность, это связано с тем, что одинаковые комбинации из 3 слов встречаются достаточно редко.

Целью выбора признаков (feature selection) является устранение признаков, наличие которых не влияет (или даже ухудшает) точность.

Для выбора признаков использовалась мера *информационной пользы* (information gain) [6].

Информационная польза $G(w)$ для признака w определяется следующим образом:

$$G(w) = -\sum_{i=1}^{m} p(c_i)\log p(c_i) + p(w)\sum_{i=1}^{m} p(c_i|w)\log p(c_i|w) + \\ + p(\overline{w})\sum_{i=1}^{m} p(c_i|\overline{w})\log p(c_i|\overline{w}),$$

c_i — класс, w — признак, $\overline{w}$ — отсутствие признака.

Для каждого признака из пространства признаков высчитывается его информационная польза. После этого признаки, чья информационная польза ниже некоторого значения k_{th} удаляются.

В ходе экспериментов было установлено, используя информационную пользу, можно значительно сократить размер признакового пространства, не ухудшая точность. Наилучшых показателей точности удалось достигнуть при удалении 62% признаков (см. рис. 1).

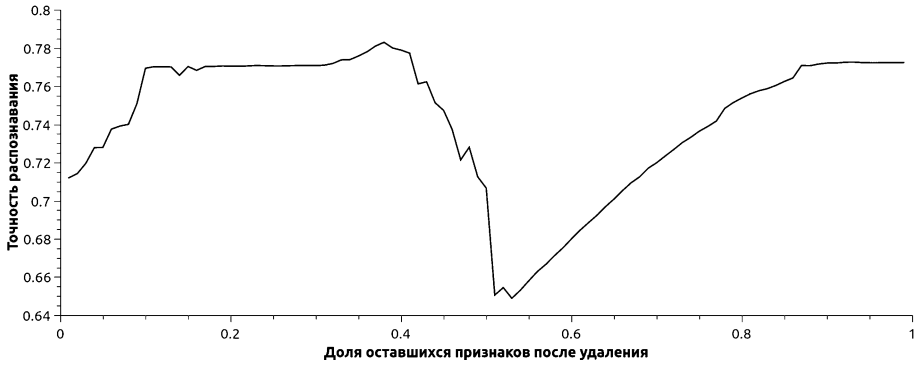

Рис. 1. Изменение точности при удалении признаков с меньшей информационной пользой.

Зависимость точности распознавания от размера обучающей выборки

Была установлена зависимость точности распознавания от объёма обучающей выборки. Для этого количество отзывов в обучающей выборке последовательно увеличивалось от 2000 до 70000 с шагом в 2000, а количество отзывов в тестовой выборке оставалось неизменным и составляло 70000.

На графике можно видеть, что точность возрастает при росте обучающей выборки (см. рис. 2). Это объясняется тем, что классификатор восстанавливает функцию плотности распределения по заданной конечной обучающей выборке, и при увеличении размера обучающей выборки, найденная плотность распределения более точно соответствует действительной[4].

Отказ от распознавания

Для повышения точности был введён *отказ от распознавания*, то есть такие условия, при которых классификатор не будет классифицировать данные.

[4]Следует помнить о том, что невозможно точно восстановить функцию плотности по конечной выборке.

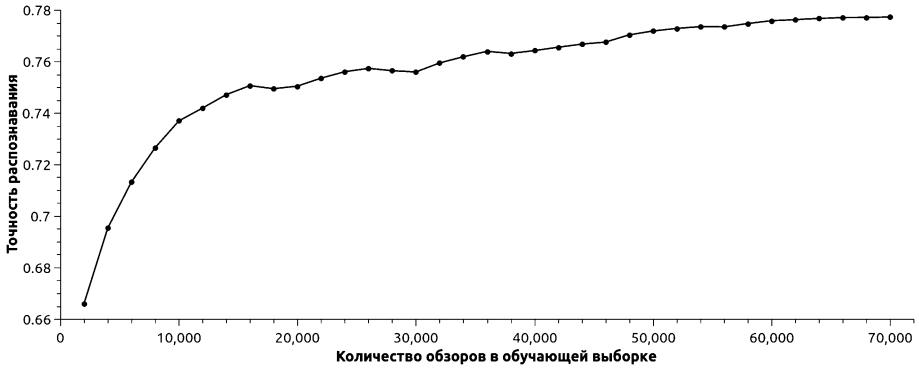

Рис. 2. Зависимость точности распознавания от размера обучающей выборки.

Описанный в данной работе классификатор не классифицирует данные, если:

$$|\ln p(pos) - \ln p(neg)| < t,$$

$p(pos)$ — вероятность того, что отзыв положительный, $p(neg)$ — вероятность того, что отзыв отрицательный.

Значение параметра t подбиралось таким образом, чтобы в число нераспознанных отзывов, попало как можно большее количество отзывов с оценками от 4 до 6.

Было подобрано $t = 1.9$, при котором доля отзывов с оценками от 4 до 6 составляла 52%.

Точность при применении отказа от распознавания составила 83.3%.

Если же принять отзывы, которые не были классифицированы за отдельный класс нейтральных отзывов, то в таком случае точность классификации на три класса составляет 67.9%.

Заключение

В таблице показано изменение точности после применения различных методик (см. табл. 1).

Был построен классификатор, который показывает уровень точности определения тональности около 78.4%. Такой результат сопо-

Описание	**Точность [%]**	**Пояснения**
Bernoulli NB	73.32	
Униграммы без стемминга	76.79	
Униграммы со стеммингом	75.66	
Комбинация униграмм и биграмм без стемминга	77.69	
Комбинация униграмм и биграмм со стеммингом	77.23	
Обработка отрицаний без стемминга	78.26	
Удаление излишних признаков без стемминга	78.39	размерность пространства признаков уменьшена до 38% от изначальной
Отказ от распознавания без стемминга	83.31	проигнорировано 16444 отзывов

Таблица 1. Изменение точности после применения различных методик.

ставим с результатами, полученными при применении NB в английском языке.

При этом были устновленны некоторые различия и сходства в применении NB для русского и английского языков.

Так же, как и в английском языке, в русском языке использование стемминга не улучшает точность.

В отличии от английского языка, в русском использование модели Бернулли для NB заметно ухудшает точность.

В отличии от английского языка, в русском обработка отрицаний в значительно меньшей мере улучшает точность.

Список литературы

[1] Кудрявцев В. Б., Гасанов Э. Э., Подколзин А. С. Введение в теорию интеллектуальных систем. — М.: Изд-во ф-та ВМиК МГУ, 2006. ISBN 5–89407–272–7.

[2] Jurafsky D, Martin J. H. Speech and Language Processing. — Prentice Hall. 2nd edition (May 16, 2008). ISBN 978–0131873216.

[3] http://snowball.tartarus.org/algorithms/russian/stemmer.html.

[4] Sanjiv D., Chen M. Yahoo! for Amazon: Extracting market sentiment from stock message boards // Proceedings of the Asia Pacific finance association annual conference (APFA). — 2001.

[5] Narayanan V., Arora I., Bhatia A. Fast and accurate sentiment classification using an enhanced Naive Bayes model // Intelligent Data Engineering and Automated Learning (IDEAL 2013). Lecture Notes in Computer Science. — 2013. Vol. 8206. — P. 194–201.

[6] Yan X., Gareth J.F., JinTao L., Bin W., ChunMing S. A study on mutual information-based feature selection for text categorization // Journal of Computational Information Systems. — 2007. 3 (3). — P. 1007–1012. ISSN 1553–9105.