Санкт-Петербургский государственный университет

Математико-механический факультет

Кафедра системного программирования

Анализ тональности текста

Курсовая работа студента 361 группы

Калмыкова Алексея Владимировича

Научный руководитель …….. ст. пр. Губанов Ю.А.

Санкт-Петербург
2012

Оглавление

Введение

Классификация текстов – одна из областей обработки натуральных языков (англ. Natural Language Processing). Эта область набирает всё большую популярность с каждым годом. Информатизация населения и перевод текстов в электронный вид (например, электронный документооборот в Российской Федерации) приводят к необходимости разработки эффективных алгоритмов анализа и классификации этих текстов.

Одной из задач классификации текстов является распознавание эмоциональной окраски текста. Эта задача может использоваться не только для классификации текста (например, в судебно-криминалистическом анализе), но и в задачах искусственного интеллекта. Распознавание эмоциональной окраски также называют анализом тональности текста. **Анализ тональности текста** (сентимент-анализ) – область компьютерной лингвистики, занимающаяся выделением из текстов эмоционально окрашенной лексики или эмоциональной оценки автора.

Любым коммерческим фирмам, производящим какой-либо продукт, интересно знать мнение покупателей об этом продукте. Эти данные могут быть использованы для повышения качества продукта, определения целевой аудитории, а также для определения главных достоинств и недостатков конкурентов. Эту задачу решает анализ мнений (англ. Opinion mining). Этот анализ заключается в двух подзадачах: 1) морфологическом анализе для выделения сущностей, которые будут оцениваться, и 2) анализе эмоциональной окраски выражений, относящихся к этой сущности.

Также, сентимент-анализ является неотъемлемой частью многих задач искусственного интеллекта. Он используется в специальных автоматизированных программах для анализа историй сообщений и других приложениях, в которых человеко-машинное взаимодействие происходит при помощи естественного языка.

Постановка задачи

Целью этой курсовой работы является разработка алгоритма, который сможет решать следующие задачи:

1) Принимать на вход файлы формата .doc, .docx, .pdf, .odt, .txt
2) Анализировать эмоциональную окраску текста из этих файлов
3) Сообщать пользователю, какая у текста эмоциональная окраска: позитивная или негативная

Также требуется быстрая скорость программы и точность не ниже 70%.

Обзор существующих решений

Существует несколько решений задачи анализа эмоциональной окраски текста, основанных на разных методах.

Twitter Sentiment

Сервис Twitter сейчас очень популярен. Пользователи сервиса, в частности, пишут в своих сообщениях отзывы о товарах. Веб-сервис Twitter Sentiment [1] позволяет анализировать информацию о продукте, который упоминают пользователи, при помощи данных из веб-сервиса Twitter. Пользователю Twitter Sentiment достаточно ввести слово, и программа проанализирует все последние 100 записей об этом слове. При этом будет построен график соотношения положительных и негативных отзывов. В совокупности, предоставляется легкий способ проанализировать мнения пользователей о любых продуктах. Twitter Sentiment использует метод машинного обучения и также имеет API.

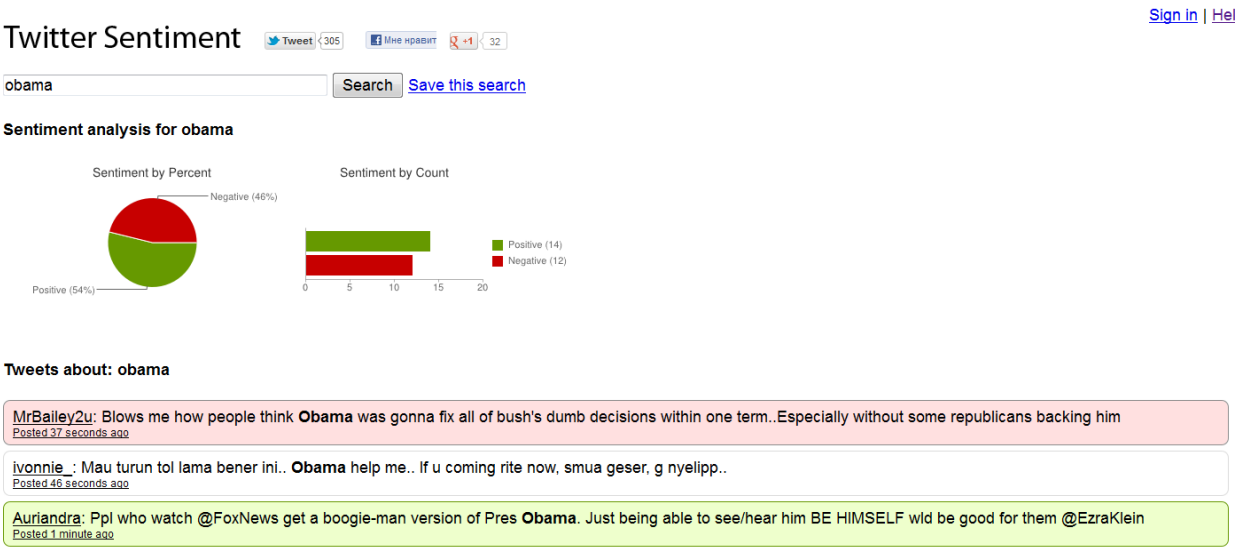

Рис. 1.1: Пример работы сервиса Twitter Sentiment

I-Teco

Данный программный продукт [2] является веб-сервисом, который позволяет определять эмоциональную окраску текста, введенного пользователем. В нем используются метрики и специальные словари эмоционально окрашенной лексики.

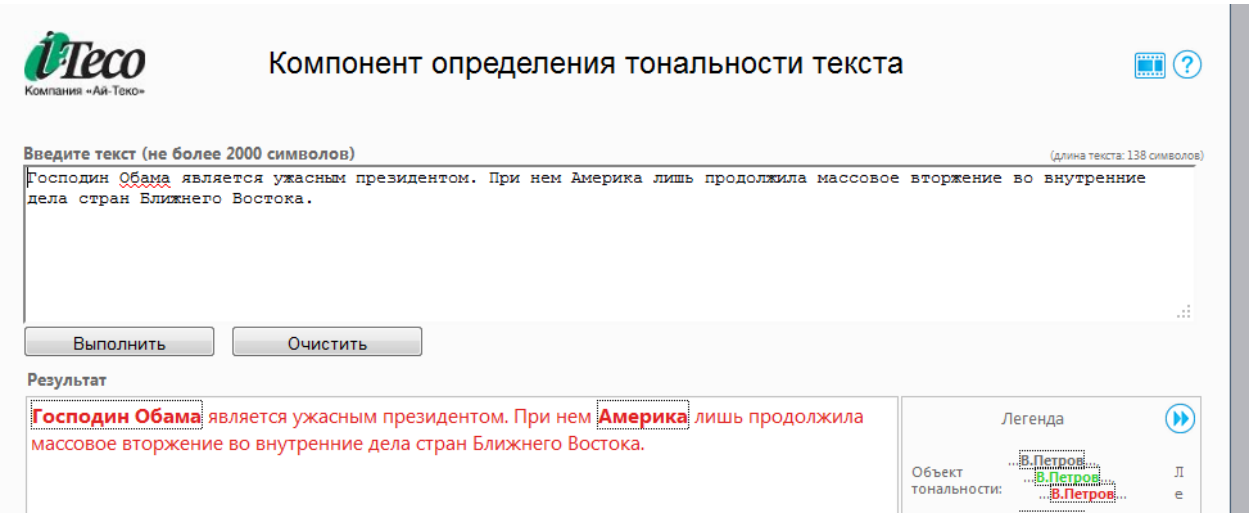

Рис. 1.2: Пример работы сервиса I-Teco

Sentiment Analysis with Python NLTK Text Classification

Данный программный продукт имеет веб-интерфейс [4]. Используется метод машинного обучения [7] с применением наивного байесового классификатора [8].

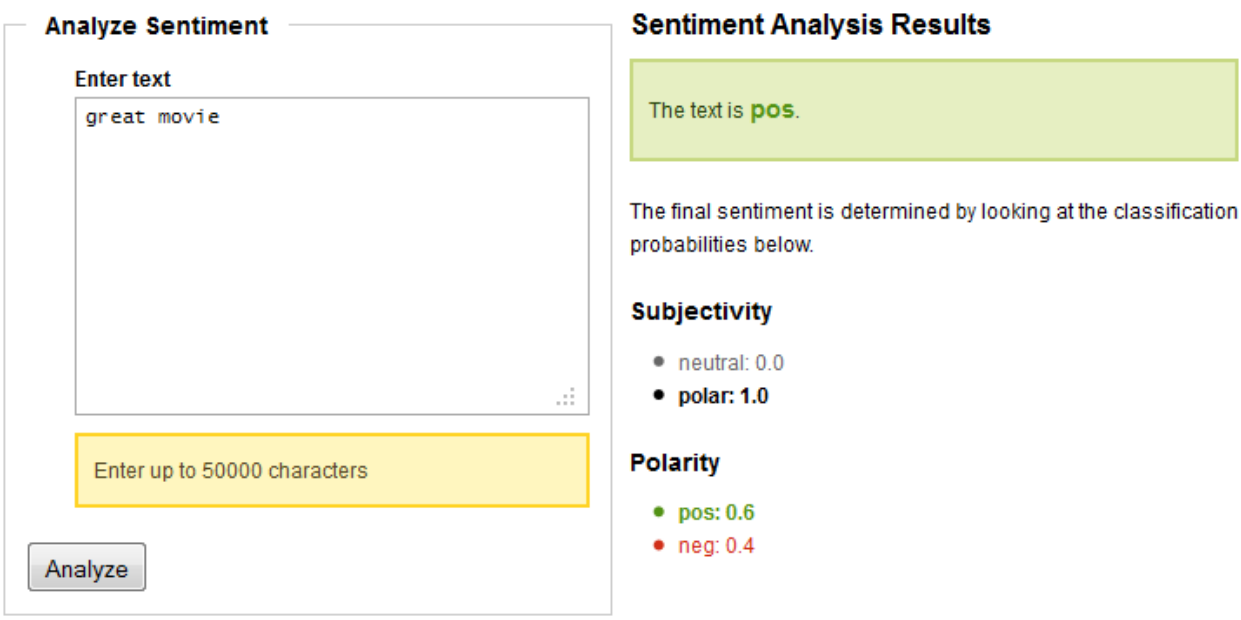

Рис. 1.3: Пример работы веб-сервиса Sentiment Analysis with Python NLTK Text Classification

Реализация

Этапы

Создание программного продукта для анализа тональности текста разбивается на несколько этапов:

1) Импорт документа в формате .doc, .docx, .pdf, .odt в текстовый формат с удалением ненужной информации (изображений, разметки, гиперссылок и т. д.)
2) Создание текстового классификатора. В данной курсовой работе будут использоваться отзывы с сайта Яндекс.Маркет [3]. Этот процесс разбивается на следующие подшаги:
    i) Выделение базы однокоренных слов для приведения их к одинаковому виду ("стемминг")
    ii) Создание матрицы документов. В этой матрице каждый из отзывов — строка, а столбец — слово из всех проанализированных отзывов. В таблице ставится 0, если это слово не встречается в отзыве и 1, если встречается.
    iii) Выделение наиболее значимых слов, которые сильнее всего влияют на рейтинг отзыва (этот шаг необходим, т.к. таблица получается очень большой)
    iv) Анализ текста на основе алгоритма логит-регрессии

Рассмотрим каждый этап подробнее.

Импорт документа

Для извлечения текста из файлов используются специальные библиотеки:

Формат	**Библиотека**	**Лицензия**
.doc	NPOI	Apache
.docx	SDK from Microsoft	Proprietary
.pdf	PDFBox	BSD
.odt	AODL	Apache

Все лицензии, под которыми распространяются данные библиотеки, позволяют использовать библиотеки бесплатно, в том числе и для коммерческого использования.

Размеченная база текстов

В качестве размеченной базы текстов будет использоваться набор отзывов с сайта Яндекс.Маркет [3]. Яндекс.Маркет – веб-сервис, предназначенный для поиска разнообразных товаров и магазинов, где эти товары можно купить. Данный веб-сервис является одним из самых популярных в русской части интернета в своём сегменте.

Яндес.Маркет предоставляет пользователям возможность публиковать отзывы о товарах и выставлять оценку товарам. Благодаря данной функции существует возможность получить текст с его эмоциональной окраской, ведь обычно отзыв с плохой оценкой окрашен отрицательно, а с высокой – положительно. Отзывы со средней оценкой имеют нейтральную окраску, и зачастую в таких отзывах встречаются как положительно окрашенная лексика, так и отрицательная.

Анализ веб-страниц

Поскольку необходимо анализировать веб-страницы сайта Яндекс.Маркет, чтобы получить отзывы с этого сайта, требуется использовать библиотеку для анализа сайтов. В данной курсовой работе будет использоваться библиотека HtmlAgilityPack [9]. Данная библиотека написана на языке программирования C# и позволяет анализировать и генерировать HTML-файлы. Также существует возможность анализа синтаксически неверных html-файлов (например, которые не проходят валидацию W3C [5]). Это необходимо в связи с тем, что даже главная страница Яндекс.Маркет содержит 19 ошибок.

Стемминг

Стемминг — это процесс выделения основы слова для заданного слова. Полученная основа является одинаковой для всех однокоренных слов. Стемминг используется для того чтобы привести все однокоренные слова к единому виду и в программе они выглядели идентично. Однако основа слов не всегда совпадает с настоящим корнем слова. Также стемминг не учитывает приставки.

Существуют разные алгоритмы стемминга В данной курсовой работе будет использоваться стеммер Портера [6]— алгоритм, который не требует словарей для своей работы. Стеммер Портера применяет последовательно ряд правил, при помощи которых он отсекает окончания и суффиксы, используя особенности языка. Стеммер Портера является самым популярным алгоритмом стемминга, а так же одним их самых простых.

Выделение важных слов

После создания матрицы документов становится понятно,

что она содержит очень много информации (в ней содержатся все слова, которые встречаются в выборке). Однако для определения эмоциональной окраски текста достаточно лишь несколько слов, которые эмоционально окрашены (они-то и задают эмоциональную окраску текста). Поэтому необходимо выделить наиболее важные слова из данной таблицы. Для этого вычисляется коэффициент корреляции между каждым словом и оценкой эмоциональной окраски из обучающей выборки. Получаем простое правило: чем больше корреляция, тем менее это слово важно для анализа.

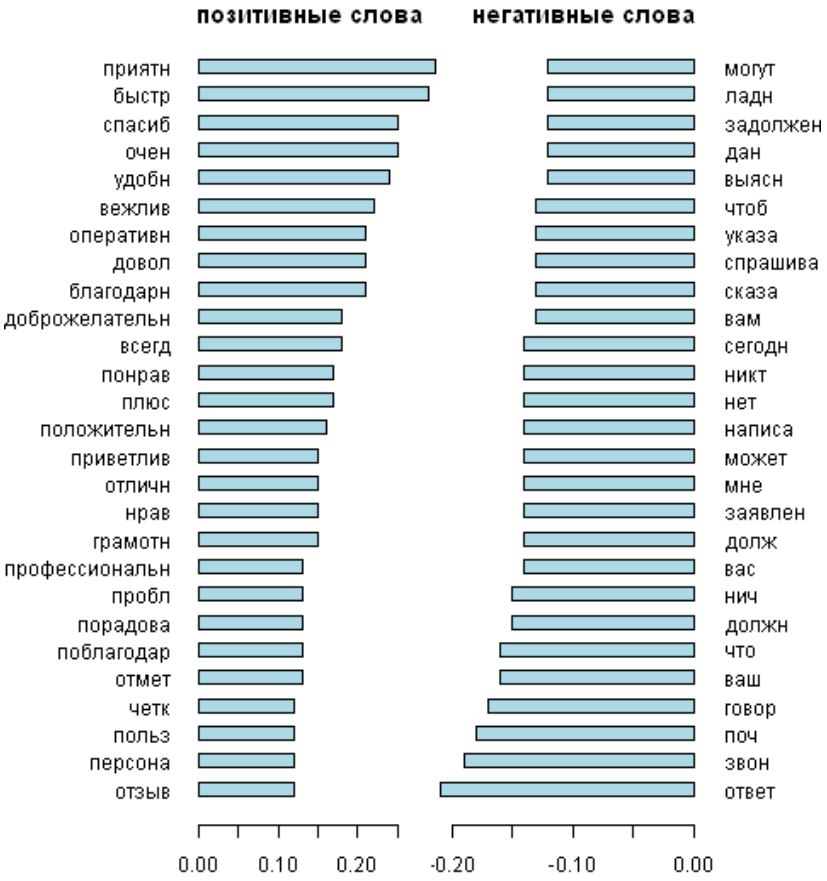

рис. 2.1. Значение корелляции отдельных слов

Логит-регрессия

Логит-регрессия (логистическая регрессия) – это статистическая модель, используемая для предсказания вероятности некоторого

события по значениям множества признаков. Для этого вводится так называемая зависимая переменная, принимающая два значения: 0 (текст окрашен негативно) или 1(текст окрашен положительно). Также вводится множество независимых переменных (отдельные слова из отзыва), на основе которых вычисляется вероятность принятия того или иного значения зависимой переменной. Делается предположение, что вероятность наступления события:

$$\Pr\{y = 1|x\} = f(z) \qquad (1)$$

Где $f(z)$ – логистическая функция, заданная формулой (2):

$$f(z) = \frac{1}{1 + e^{-z}} \qquad (2)$$

Если $\Pr\{y = 1|x\} > 0.5$, то документ считается положительно окрашен, иначе – негативно.

Для вычисления вероятности получаем формулу (3):

$$\ln\left(\frac{p}{1-p}\right) = a * word_1 + b * word_2 + \cdots \qquad (3)$$

где p – вероятность, которую необходимо вычислить, а коэффициенты a,b,… - корреляция, вычисленная в предыдущих шагах.

Архитектура приложения

Данное приложение будет написано на основе архитектуры Модель-Представление-Поведение. Эта архитектура подразумевает, что приложение разбивается на три составляющих: модель, представление и поведение.

В данной курсовой работе моделью будет являться размеченная база текстов. В ней присутствует два основных метода: void Train(string text, int mark) и int GetMarkForText(string text). Метод Train позволяет модели обучаться на основе нового текста и соответствующей ему эмоциональной окраске, а метод GetMarkForText возвращает значение

эмоциональной окраски текста в соответствии с результатами классификатора. В этой модели так же будет храниться матрица документов, список “важных слов”, а так же стеммер Портера. Представление не будет иметь прямого доступа к методам модели. Таким образом, основные методы обработки данных будут изолированы от представления, и изменения никак не коснутся саму модель.

Оценка ошибок

В рамках данной курсовой работы было обработано 3000 отзывов с сайта Яндекс.Маркет. Для обучения были использованы 2000 отзывов, для тестирования – 1000. Тестирование показало, что точность алгоритма составляет около 60%(верно были проанализированы 582 отзыва). Стоит заметить, что точность алгоритма выше, когда на вход попадают положительно окрашенные данные. Это связано с тем, что в обучающей выборке количество положительных отзывов было намного больше чем отрицательных.

Заключение

Результаты

В рамках данной курсовой работы было достигнуты следующие результаты:

- проведен обзор предметной области и существующих решений;
- реализован программный продукт для создания базы размеченных текстов;
- реализован алгоритм определения тональности текста на основе логистической регрессии,
- реализован алгоритм стемминга (стеммер Портера),
- реализован программный продукт для анализа эмоциональной окраски текста из файлов .doc, .docx, .pdf, .odt, .txt с точностью не менее 60%.

Исходный код проекта можно скачать по ссылке http://code.google.com/p/sentiment-analyzator/.

Литература

[1] Веб-сервис Twitter Sentiment: http://www.sentiment140.com

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

[3] Веб-сервис Яндекс.Маркет, http://market.yandex.ru

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

[5] Веб-сервис валидации HTML-страниц W3C, http://validator.w3.org

[6] The Porter Stemming Algorithm, http://tartarus.org/~martin/PorterStemmer/

[7] Машинное обучение, http://ru.wikipedia.org/wiki/Машинное_обучение

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

[9] Библиотека HtmlAgilityPack, http://htmlagilitypack.codeplex.com/