Правительство Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Национальный исследовательский университет
«Высшая школа экономики»

Факультет Бизнес-информатики
Отделение Прикладной математики и информатики
Кафедра Анализа данных и искусственного интеллекта

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА
на тему

Машинное обучение для определения тональности и классификации текстов на несколько классов

Выполнил студент группы 472
Чусовлянов Дмитрий Сергеевич

Научный руководитель:
Кандидат наук, доцент,
Игнатов Дмитрий Игоревич

Консультанты:
PhD, Панченко Александр Иванович

Москва 2014

Оглавление

Аннотация

В данной работе исследуются два подхода к задаче определения тональности текста: подход, основанный на методах машинного обучения и подход, основанный на использовании словарей тональной лексики. Описан и реализован метод для автоматического извлечения из текста слов, несущих эмоциональную оценку. Описан и реализован метод последующей классификации текстов на основе полученного словаря тональностей.

Для некоторых стандартных методов машинного обучения (таких метод наивной байесовской классификации, метод опорных векторов, Random Forest) предложено использовать слова из полученного словаря тональности в качестве признаков классификации.

Методы протестированы на реальных данных – отзывах пользователей Интернет-ресурса imhonet.ru по трем предметным областям: книги, фильмы, камеры. Данные были представлены на Российском семинаре по оценке методов информационного поиска (РОМИП).

Экспериментально выявлен оптимальный размер автоматически конструируемого словаря тональности, основываясь на сентимент-величине, согласно которой ранжируются слова в словаре. В ходе экспериментов выявлено оптимальное значение сентимент величины, по которой производится отсечение части словаря тональности. Это позволяет исключить из рассмотрения часть словаря с низкой концентрацией оценочных слов.

В работе представлены эксперименты, направленные на то, чтобы сравнить качество построения словарей тональности для различных предметных областей и различных наборов входных данных.

Для методов машинного обучения помимо показателей качества (такие как Accuracy, Macro_Precision, Macro_Recall) для каждого метода в отдельности подсчитаны также средние величины, что позволяет провести сравнение методов машинного обучения с методами, использующими словари тональности.

Также проведены эксперименты, на основе которых возможно сравнить методы машинного обучения, работающие со словами из полученного словаря как с критериями сортировки, с методом подсчета агрегированной сентимент-величины текста на основе слов, входящих в данный текст и принадлежащих словарю тональности. Сравнения проводились как для средних показателей качества, полученных при тестировании методов машинного обучения, так и для отдельного сравнения методов друг с другом.

Ключевые слова: Анализ данных, машинное обучение, анализ тональности, словарь тональности, SVM, Random Forest, Naive Bayes.

Abstract

This paper examines two approaches to the problem of opinion mining: an approach based on machine learning methods and an approach based on the use of sentiment-lexicons. The method of automated extraction words from the text that carry emotional value was described and implemented. Subsequent text classification method based on the obtained sentiment-lexicons was described and implemented.

For some standard machine learning techniques (such method is naive Bayes classification, support vector machines, Random Forest) it is proposed to use words from the obtained sentiment-lexicon as classification features.

Methods are tested on real data –feedback and comments of Internet resource imhonet.ru user's on three subject areas: books, movies and camera. The data were presented at the seminar on Russian Information Retrieval Evaluation Seminar (ROMIP).

The optimal size of a sentiment-lexicon (which is automatically constructed) is experimentally determined and is based on sentiment-value, according to which words are ranked in the dictionary. The experiments revealed the optimal value of the sentiment value, over which the part of the dictionary is cut-off. This eliminates from consideration part of the vocabulary with low concentration of sentiment words.

The paper presents experiments aimed to compare the quality of constructed sentiment-lexicons for different domains and different sets of input data.

For machine learning methods indicators of quality were calculated in addition to average ones (such as Accuracy, Macro_Precision, Macro_Recall). That allows us to compare machine learning methods and method based on sentiment lexicons.

It is also possible to compare the machine-learning techniques, working with words from the dictionary as classification features, and the method of calculating the aggregated sentiment-based text value based on sentiment of the words in the text and belonging to the lexicon. Comparisons were made for both the average quality values obtained testing machine learning techniques and sentiment approach, and for the separate methods.

Key words: Opinion mining, machine learning, sentiment analysis, SVM, Random Forest, Naive Bayes, lexicon-based approach.

Введение

С развитием интернет-сервисов каждый пользователь получил в числе прочих возможность выражать свое мнение. Это может быть мнение относительно товара или услуги, фильма или книги, компании или политического деятеля. Таким образом возникла потребность обрабатывать огромные объемы информации для определения отношения пользователей к тому или иному объекту.

Очевидно, что количество отзывов публикуемых, например, в социальных сетях достигает десятков тысяч, и обработка отзывов вручную экспертами оказывается невозможной. В связи с этим широкое распространение получили такие области Computer Science, как Opinion Mining и Sentiment Analysis (от английского sentiment – чувство). Стало возможным автоматически получить или «извлечь» мнение, выраженное в тексте. Для этого применялись методы машинного обучения (Machine Learning), позднее стали появляться методы, основанные на использовании словарей тональных слов (lexicon-based approach).

Основной задачей являются автоматическое извлечение мнений из текстов, то есть определение, содержит ли данный текст субъективную составляющую, а также классификация текстов на основе тональности на два (позитив и негатив) или более классов. Под тональностью здесь понимается эмоциональная оценка, выраженная автором относительно некоторого объекта.

Крупная торговая сеть по отзывам в интернете может узнать отношение клиентов к процессу обслуживания в магазине, исследуя отзывы пользователей. Банк может узнать мнение клиентов о процессе обслуживания в конкретных отделениях. Например, один из крупнейших мировых автоконцернов провел исследование мнений автовладельцев, в ходе которого удалось установить связь между неисправностью и номером партии, вследствие чего некоторые партии автомобилей были полностью отозваны из продажи.

Проблематика в задаче определения тональности

Прежде всего, сама задача определения эмоциональной оценки текста субъективна. Так, согласно некоторым опытам [4], разные люди могут по-разному оценить один и тот же текст. Более того, мнения могут быть противоположными – часть испытуемых относит текст в положительный класс, а часть – в отрицательный.

Также тексты на естественном языке являются неструктурированными объектами, что осложняет работу с ними. Кроме того, в тексте могут присутствовать сарказм, шутки, опечатки, которые не всегда может понять человек, не говоря уже о машине.

Еще одна сложность заключается в том, что методы разрабатывается для конкретного языка и могут не работать на другом языке. Например, методы, разрабатываемые для текстов на английском языке, могут быть неприменимы для русских текстов.

Тональность текста напрямую зависит от предметной области. В частности, при использовании списка оценочных слов (словаря тональностей) эмоциональная оценка одного и того же слова может меняться в разных предметных областях. Так, слово нейтральное в одной области («я изучаю **немецкий** язык» - нейтральное предложение), может быть оценочным словом в другой области («настоящее **немецкое** качество» - положительная оценка). Слово может принимать также разную тональность в зависимости от контекста: «в магазине продавали исключительно **старое** вино» и «в отделении банка только **старые** банкоматы»).

В данной работе рассматривалась задача классификации тональности текстов. Исследовались методы классификации на основе машинного обучения и с использованием словаря тональностей.

Целью работы являлось исследование, модификация, программная реализация и последующее сравнение качества методов автоматического построения словаря тональностей. Последующее сравнение алгоритмов классификации текстов на основе полученного словаря и на основе методов машинного обучения.

Исследования проводились на реальных данных – отзывах пользователей сайта Imhonet.ru о книгах и фильмах. Данные были представлены на Российском семинаре по оценке методов информационного поиска (РОМИП).

Для текстов на русском языке практически не существует экспертно-размеченных словарей тональности (первый словарь на русском языке представлен в работе [1]), что определило научную новизну работы. Реализация метода на языке программирования Python дает возможность протестировать алгоритм на реальных данных и сравнить показатели с классическими алгоритмами классификации, основанными на методах машинного обучения, что определило практическую значимость работы.

В процессе работы были решены следующие задачи:

1. Обзор задачи и методов анализа мнений, классификации текстов.
2. Исследование и разработка метода извлечения оценочных слов для заданной предметной области;
3. Разработка методов классификации текстов на основе построенного словаря;
4. Эксперименты на реальных данных.

Структура работы

Данная работа структурирована следующим образом. В первой главе проводится анализ задачи анализа тональности и классификации текстов. Приводится формальная постановка задач. Производится обзор основных подходов к решению данных задач. В главе рассмотрен подход к определению

качества работы программ-классификаторов.

Во второй главе описываются алгоритмы классификации текстов на основе тональности. Описываются некоторые особенности и ограничения работы алгоритмов.

В третьей главе приведены алгоритмы, используемые в данной работе. Некоторые из них реализованы на языке программирования Python, некоторые были протестированы с помощью пакета Weka, используемого для решения задач классификации документов на основе реализованных в пакете методов машинного обучения.

В четвертой главе приведена методология, используемая и применяемая в работе. Подробно описываются все этапы, проводимые в рамках данной работы.

В пятой главе описываются результаты тестирования реализованных алгоритмов. Произведены тесты на реальных данных, получены основные показатели качества алгоритмов. Экспериментально выбраны наилучшие значения констант, используемых в данных методах классификации.

В заключении описаны полученные результаты, на основе результатов сделаны выводы, подведены итоги работы и приведены возможные дальнейшие направления исследования.

Глава 1. Задачи классификации и определения тональности текстов.

1.1. Формализация задачи классификации текстов

Определение. Классификация документов — одна из задач информационного поиска, заключающаяся в отнесении документа к одной из нескольких категорий на основании содержания документа.

Определение. Классификатор – это алгоритм соотносящий некие входные данные с одним или несколькими классами. В отличие от алгоритмов кластеризации эти классы должны быть определены заранее.

Пусть заданы некоторое конечное множество категорий $C = \{c_1 \dots c_{|C|}\}$, конечное множество документов $D = \{d_1 \dots d_{|D|}\}$ и неизвестная целевая функция Ф, которая для каждой пары <документ, категория> определяет, соответствуют ли они друг другу: Ф: D×C →{0,1}.

Задача состоит в том, чтобы найти максимально близкую к функции Ф функцию Ф'. Функцию Ф' называют **классификатором**.

Машинное обучение основывается на начальной коллекции документов $Q = \{d_1 \dots d_{|Q|}\} \subseteq D$. При этом, значение целевой функции Ф известно для каждой пары $< d_i , c_j > \in Q \times C$. Документы из Q разделяют на две непересекающиеся коллекции:

- **''учебную"** $T_r = \{d_1 \dots d_{|T_r|}\}$. Коллекция документов, *с* помощью которой создается классификатор Ф'. Ф' обучается индуктивно, основываясь на замеченных характеристиках этих документов.
- **''тестовую"** $T_e = \{d_{|T_r|+1} \dots d_{|Q|}\}$. Коллекция документов, на которой тестируется эффективность построенного классификатора. Каждый «тестовый» документ подается на вход классификатору Ф', затем сравнивается результат классификатора $\Phi'(d_i , c_j)$ *с* известным

значением функции $\Phi(d_i, c_j)$. Классификатор считается тем эффективнее, чем чаще эти значения совпадают.

Документ $d \in Q$ называется положительным или отрицательным примером для категории c, если значение функции $\Phi(d, c)$. равно 1 или 0, соответственно.

Стоит отметить, что существует два различных наиболее распространённых вида классификации. В зависимости от ответа, классификация бывает:

- точная: Φ': D×C $\rightarrow \{0,1\}$.
- ранжированная: Φ': D×C $\rightarrow [0,1]$

Таким образом, классификация может быть **точной**, когда каждой паре <документ, класс> ставится в соответствие булево значение – истина или ложь, то есть, соответствует документ категории или нет. Второй тип классификации называется **ранжированным.** Каждой паре <документ, класс> классификатор сопоставляет число, характеризующее степень принадлежности документа к тому или иному классу и лежащее в диапазоне [0,1].

1.2. Задача анализа мнений

Определение. Анализ тональности текста (сентимент-анализ - Sentiment analysis / Opinion mining) - класс методов, предназначенный для автоматизированного выявления в текстах эмоционально окрашенной лексики и эмоциональной оценки мнений авторов по отношению к объектам, речь о которых идёт в тексте.

Эмоциональная оценка, выраженная в тексте, также называется тональностью, или сентиментом текста (от англ. sentiment - мнение, настроение). Эмоциональная составляющая, выраженная на уровне лексемы или коммуникативного фрагмента, называется лексической тональностью (или

лексическим сентиментом). Тональность целого текста определяется лексической тональностью составляющих его единиц и правилами их сочетания.

Тональность текста определяется тремя факторами:

1) субъект тональности;
2) тональная оценка (например, позитив/нейтрально/негатив);
3) объект тональности.

Под субъектом тональности подразумевается автор статьи (автор цитаты, прямой или косвенной речи), под объектом тональности — тот, о ком автор высказывает свое мнение, под тональной оценкой — эмоциональное отношение автора к такому объекту.

Формальная модель объекта [2]: каждый объект o представляется в виде конечного набора $атрибутов$ $F = \{f_1 \dots f_{|F|}\}$, который включает в себя и сам объект в виде специального атрибута.

Формальная модель мнения [2]: В общем виде некоторый документ d содержит мнения о наборе объектов $o_1 \dots o_{|Q|}$ от набора владельцев мнений $h_1 \dots h_{|Q|}$. Мнение по каждому объекту o_j выражено в отношении подмножества его атрибутов F_j.

Определение. $Мнение$ — это пятёрка

$$(o_j,\ f_{jk}, oo_{ijki}, h_i,\ t_i),$$

где o_j это некоторый объект, f_{jk} это атрибут объекта o_j, oo_{ijki} это тональность мнения по отношению к атрибуту f_{jk}, h_i это владелец мнения, а t_i это время, в которое было высказано мнение.

1.3. Основные этапы работы классификатора

Классификацию можно разделить на три основных этапа:

A. Индексация документа;

B. Обучение классификатора;

C. Оценка качества работы классификатора.

1.3.1. Обработка документа

Индексацией называется процесс приведения документов к единому формату. Зачастую приходится иметь дело с большими объемами информации, поэтому при индексации необходимо удалить термы, не несущие эмоциональной оценки. К примеру, некоторые слова (предлоги, союзы) и знаки препинания могут очень часто встречаться во всех документах, при этом не нести никакой смысловой нагрузки.

Второй этап обработки текста – приведение слов к начальной форме, что позволяет представить текст (документ) в виде вектора:

Будем считать, что каждый документ - это просто набор слов (термов). Множество всех термов обозначим за T. Каждый терм $t_i \in T$ имеет вес w_{ij} по отношению к документу $d_j \in D$. Таким образом, каждый документ можно представить в виде вектора весов его термов $\overrightarrow{d_j} =< w_{1j} \dots w_{|T|j} >$. Веса документов нормируют так, чтобы $0 < w_{ij} < 1$, для $\forall i, j: 0 \leq i \leq |T|, 0 \leq j \leq |D|$.

Вес терма в документе можно определить следующим образом:

$$w_{ij} = TF_{ij} * IDF_i$$

где TF_{ij} - это отношение числа термов t_i в документе d_j к общему числу термов в этом документе, а IDF_i - число, обратное количеству документов, в котором встречается терм t_i. Таким образом, чем чаще слово встречается в документе, но реже встречается вообще во всех документах, тем больше вес этого терма в данном документе.

1.3.2.Работа классификатора

В машинном обучении

В алгоритмах классификации на основе машинного обучения происходит сравнение текстов (поступающих на вход алгоритму) с ранее размеченным эталонным корпусом по выбранной мере близости и отнесение (классификация) текста к тому или иному классу на основании полученного результата сравнения.

Обучение классификатора состоит в выборе общей формы классифицирующего правила со множеством различных параметров. Далее производится настройка параметров на ’’обучающем” наборе документов.

Наиболее простой способ состоит в следующем: классификатор ставит в соответствие каждому документу число: $F: D \to R$. То есть все тексты ранжируются по некоторой шкале. Далее для каждой категории c_i выбирется пороговое значение μ_i. Документ d соответствует классу c_i если $F(d) > \mu_i$.

В словарях тональности

Поиск эмоционально окрашенной лексики (тональности) в тексте по заранее составленным словарям тональности с применением лингвистического анализа, либо построенных автоматически с использованием статистических или иных методов. По суммарному количеству найденных термов тексту ставится в соответствие число, отражающее оценку текста по некоторой шкале, в соответствии с которой происходит последующее отнесение к тому или иному классу.

1.3.3. Оценка качества работы классификатора

Существует два подхода к **оценке качества классификации.** Первый - это сравнение классификаторов между собой, второй - абсолютная оценка качества. Вообще говоря, довольно сложно оценить качество классификации. Часто даже опытные эксперты не могут определить к какой категории отнести

конкретный документ. Наиболее распространённая система (метрика) оценки качества классификации включает в себя оценку двух характеристик классификатора – точность и полноту.

Точность системы в пределах класса – это доля документов, действительно принадлежащих данному классу относительно всех документов, которые система отнесла к этому классу. Полнота системы – это доля найденных классификатором документов, принадлежащих классу относительно всех документов этого класса в тестовой выборке.

Эти значения легко рассчитать на основании таблицы контингентности, которая составляется следующим образом:

Таблица 1.1. Разбиение документов на классы согласно оценке классификатора

Категория i		Экспертная оценка	
		Положительная	Отрицательная
Оценка системы	Положительная	TP	FP
	Отрицательная	FN	TN

В таблице содержится информация о том, сколько раз система приняла верное и сколько раз неверное решение по документам заданного класса. А именно:

- *TP* — истино-положительное решение;
- *TN* — истино-отрицательное решение;
- *FP* — ложно-положительное решение;
- *FN* — ложно-отрицательное решение.

На основе вышеописанных величин производится подсчет метрик, которые не только оценить качество работы классификатора, но и позволяют сравнить произвести сравнение классификаторов между собой. В данной работе

были использованы следующие метрики, которые являются самыми распространенными и часто используемыми:

Полная точность или **аккуратность (accuracy):**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FP}$$

- доля корректно классифицированных документов от общего числа документов, подаваемых на вход классификатору.

Точность (Precision):

$$Precision = \frac{TP}{TP + FP}$$

- Для положительных примеров - доля документов, являющихся позитивными от общего числа примеров, классифицированных как позитивные. Другими словами, количество положительных документов среди всех документов, которые классификатор считает положительными.

$$Precision = \frac{TN}{TN + FN}$$

- Для негативных документов - доля документов, являющихся негативными от общего числа примеров, классифицированных как негативные.

Полнота (Recall):

$$Recall = \frac{TP}{TP + FN}$$

- Для положительных примеров - доля правильно классифицированных позитивных примеров от общего числа позитивных примеров. Другими словами, это доля действительно положительных документов из всех документов, распознанных как положительные

$$Recall = \frac{TN}{FP + TN}$$

- Для отрицательных примеров - доля правильно классифицированных негативных примеров от общего числа негативных примеров. Другими словами, это доля действительно негативных документов из всех документов, распознанных как негативные.

F – мера (F – measure):

$$F = \frac{2 * Precision * Recall}{Precision + Recall}$$

- мера, комбинирующая точность и полноту.

Macro_P [25]:

Пусть $C = \{c_1 \dots c_{|C|}\}$ – множество всех классов (категорий) данной задачи классификации. Тогда мера Macro_P – есть среднее арифметическое величин Precision для всех классов из множества *C*:

$$Macro\_P = \frac{1}{|C|} * \sum_{\forall i:\, c_i \in C} precision(c_i)$$

Macro_R[25]:

Аналогичным образом вводится величина Macro_R . Мера Macro_R – есть среднее арифметическое величин Recall для всех классов из множества *C*:

$$Macro\_R = \frac{1}{|C|} * \sum_{\forall i:\, c_i \in C} Recall(c_i)$$

Выводы и результаты

В главе 1:

- приведена формальная постановка задачи классификации текстов, приводятся основные определения, используемые в работе.
- приведено формальное описание задачи анализа мнений, рассмотрены основные понятия, такие как субъект и объект тональности. Приведены основные определения, касающиеся данной задачи.
- рассматривается подход к классификации документов, на основе их тональности.
    - Выделяются такие процессы, как нормализация документа, что позволяет оптимизировать входные данные для программы-классификатора.
    - Описываются особенности работы классификатора в случае двух основных подходов- с использованием методов машинного обучения или словарей тональности.
    - рассмотрен подход к оценке качества произведенной классификации. Введены основные показатели (метрики) определения качества работы классификатора для классификации на произвольное количество классов.

Глава 2. Обзор существующих методов классификации текстов.

На данный момент существует два основных подхода к проблеме анализа тональности текстов: подход, основанный на методах машинного обучения и подход, основанный на использовании словарей тональной лексики. **[18].** В основе подхода, основанном на использовании словарей тональности, лежит анализ тональности отдельных слов (термов) в тексте и последующее определение тональности всего текста согласно оценкам отдельных слов, входящих в этот текст. Для этого в основном используются словари тональности, в которых каждому слову ставится в соответствие величина, которая отражает «вес» слова в тональности всего текста. В последствии, согласно предложенному методу строится функция, которая на вход принимает количество вхождений в текст каждого слова и вычисляет агрегированную величину тональности всего текста [9, 10, 18, 25]:

$$W_i = f\{n_i(w_1), \dots, n_i(w_\alpha)\},$$

Где $n_i(w_j)$ – количество вхождений слова j в текст i. α – количество слов в словаре тональности. W_i – тональность текста i

В подходе, основанном на машинном обучении задача анализа тональности сводится к задачи классификации текстов [17], которая может быть решена путем обучения классификатора на заранее размеченной коллекции текстов [4, 7, 19, 20].

Каждый подход обладает своими преимуществами и недостатками. Так, например, методы, основанные на использовании словарей тональности не нуждается в обучающей коллекции, то есть нет необходимости в ручной разметке текстов. Также данные методы не нуждаются в составлении обучающей функции. Таким образом, «решения», принимаемые классификатором могут быть легко объяснены. В то же время для данных

методов требуются заранее размеченные словари тональности, которые кроме всего прочего должны учитывать предметную область исследуемого текста.
В методах машинного обучения не требуются словари тональности, и на практике классификаторы демонстрируют высокое качество классификации. Более того, качество классификации можно улучшить за счет выбора параметров (признаков) классификации и правильно подобранные комбинации текстовых документов в обучающей выборке. В то же время классификатор, обученный на текстах одной предметной области, может не справляться со своей задачей для текстов из другой предметной области. [8].

Ниже приведены методы, используемые для решения задачи анализа тональности.

2.1. Наивный метод Байеса

Пусть *P(c|d)* - Вероятность того, что документ *d* принадлежит классу *c*.

В наивной байесовской классификации задача состоит в нахождении наиболее вероятного класса c_m для данного документа *d*

$$c_m = \underset{c \in C}{\operatorname{argmax}} P(c|d)$$

используя формулу Байеса, можно переписать выражение для *P(c|d)*

$$c_m = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)} = \underset{c \in C}{\operatorname{argmax}} P(d|c)P(c)$$

Где

- знаменатель *P(d)* не зависит от *c* и, следовательно, не влияет на нахождение максимума, поэтому его можно опустить;
- *P(c)* - вероятность того, что встретится класс *c*, независимо от рассматриваемого документа. $P(c) = \frac{N_c}{N}$
    - N_c - количество документов в классе *c*;
    - N - общее количество документов в обучающем множестве.

- *P(d|c)* - вероятность встретить документ *d* среди документов класса *c*.

Документ можно представить в виде вектора входящих в него термов (слов). Тогда величина *P(d|c)* вычисляется следующим образом:

$$P(d|c) = P(t_1, t_2, ..., t_{n_d}|c) = P(t_1|c)P(t_2|c)...P(t_{n_d}|c) = \prod_{k=1}^{n_d} P(t_k|c)$$

Следовательно, формула для вычисления наиболее вероятного класса принимает следующий вид:

$$c_m = \underset{c \in C}{\operatorname{argmax}} \hat{P}(d|c)\hat{P}(c) = \underset{c \in C}{\operatorname{argmax}} \hat{P}(c) \prod_{k=1}^{n_d} \hat{P}(t_k|c)$$

Метод наивной байесовской классификации строится на двух предположениях

- о условной независимости термов и
- о независимости позиций термов.

Другими словами, не принимается во внимание тот факт, что в тексте на естественном языке появление двух разных слов часто взаимосвязано (например, вероятнее, что слово **«удар»** встретится в одном тексте со словом **«бокс»**, чем со словом **«балет»**), и, во-вторых пренебрежение тем фактом, что вероятность встретить одно и то же слово на разных позициях (местах) в тексте различна. Именно поэтому рассматриваемая модель естественного языка называется наивной. Несмотря на это, модель показывает достаточно точные результаты классификации.

2.2. Деревья

Идея данного метода состоит в построении **разрешающего дерева** на «обучающем» наборе документов. Дерево строится по следующему правилу: выбираем терм, документы, содержащие этот терм кладем направо, остальные налево. Таким образом, документы разделились на две непересекающиеся коллекции. Для каждой коллекции выбирается новый терм и повторяется описанная выше процедура. Так продолжается до тех пор, пока не получится

однородная коллекция, то есть коллекция, в которой либо все документы соответствуют категории, либо все документы соответствуют ее дополнению.

Пусть имеется Q - обучающая выборка документов, а $C = \{c_1 \dots c_{|C|}\}$ - множество классов. Пусть множество классов состоит из $|C| = m$ элементов. Для каждого примера из Q известна его принадлежность к какому-либо из классов $\{c_1, \dots, c_{|C|}\}$, то есть известно значение целевой функции для всех документов в обучающей выборке. $\forall d \in Q$ Ф: D×C →{0,1}.

В листьях разрешающего дерева размещаются значения целевой функции, в прочих узлах — условия перехода, определяющие направление движения вдоль ребер дерева. Для классификации каждого примера алгоритму необходимо пройти все дерево от корня до одного из листов. Тем самым получить значение целевой функции. Алгоритм C4.5 является одним из самых популярных алгоритмов построения деревьев решений.

Алгоритм

Имеется корень и связанное с ним множество *Q*, которое необходимо разделить на подмножества. Для этого используется один из атрибутов w_j в качестве проверки. Выбранный атрибут w_j принимает *k* значений, что позволяет разделить множество на *k* подмножеств. После чего создаются *k* потомков корня. Каждому потомку соответствует некоторое подмножество, полученное после разбиения множества *Q*.

Функция выбора атрибута w_j и последующего разбиения по нему рекурсивно применяется ко всем *k* потомкам и завершает свою работу в двух случаях:

- после очередного разделения на подмножества в вершине i оказываются документы из одного класса (тогда вершина отмечается как *лист*. Класс, которому принадлежат документы в этой вершине, является обозначением листа);

- вершина после выполнения очередного шага оказалась ассоциированной с пустым множеством (в таком случае она назначается листом, а в качестве решения выбирается наиболее частый класс у непосредственного предка данной вершины).

2.3. Random Forest

Алгоритм Random Forest – ансамблевый метод машинного обучения, который использует ансамбль деревьев решений. Он основывается на основных подходах бэггинга и выбора случайных подмножеств признаков. Этот алгоритм позволяет достичь высокой точности классификации. Деревья в ансамбле строятся друг от друга независимо. [21]

Финальная классификация документов проводится с помощью «голосования», то есть итоговым классом объекта объявляется тот класс, за который проголосовало наибольшее количество деревьев.

Метод демонстрирует высокое качество классификации, сравнимое с SVM или бустингом, но в то же время обладает высокой сложностью по памяти для хранения деревьев - $O(N * K)$. [22]

2.4. Метод опорных векторов (SVM)

Метод опорных векторов (Support Vector Machine, SVM) – метод, в котором основой является построение (оптимальной) разделяющей гиперплоскости. Некоторая выборка линейно разделима, если в ней возможно получить (построить) линейный пороговый классификатор:

$$sign\left(\sum_{i=1}^{m} w_i * x^i - w_0\right) = sign(< w, x > - w_0)$$

где $x = (x^1, \dots, x^n)$ - признаковое описание объекта x; вектор $w = (w^1, \dots, w^n) \in \mathbb{R}^n$ и скалярный порог $w_0 \in \mathbb{R}$ являются параметрами алгоритма.

Таким образом задача состоит в том, чтобы подобрать значения вектора w такие, при которых функционал, определяющий число ошибок равен нулю:

$$\sum_{i=1}^{n} [y_i(< w, x_i > - w_0) \leq 0] = 0$$

Где $< w, x >= w_0$ – разделяющая гиперплоскость. Более подробное описание алгоритма можно найти в [7].

2.5. Модель мешок слов

Для реализации методов машинного обучения существует классическая модель “Мешок слов” (Bag of Words). Формальная постановка задачи выглядит следующим образом:

Пусть $f_1, \dots, f_m$ - множество, состоящее из m признаков (атрибутов), которые могут присутствовать в документе; пусть $n_i(d)$ – это количество вхождений признака f_i в документ d. Далее каждый документ d представляется в виде вектора следующим образом:

$$\vec{d} = (n_1(d), n_2(d), \dots, n_m(d))$$

Выделяют два основных типа атрибутов:

a. Частотные - каждое значение в векторе $\vec{d}$ соответствуют количеству вхождений признаков в документ d; тогда $n_i(d) \in (0; +\infty)$

b. Бинарные (наличия/отсутствия), каждое значение в векторе $\vec{d}$ бинарное (true/false или 0/1) и отражает факт присутствия признака f_i в документе d. Тогда $n_i(d) = \{0,1\}$

Далее документы, представленные в виде векторов своих признаков (атрибутов), используются для обучения классификатора, реализованного с

помощью одного из методов машинного обучения.

Выводы и результаты

В главе 2:

- Проведен сравнительный анализ двух подходов к задаче классификации: подхода, основанного на методах машинного обучения и подхода, основанного на использовании словарей тональности.
- Вместе с этим представлен обзор работ в области, посвященных данным подходам.
- Приведены существующие методы машинного обучения для задачи классификации текстов.

Глава 3. Методы, используемые в работе

3.1. Метод странности слов

I. Входные данные

В основе данного метода лежит понятие «странности» слова. Для вычисления признака «Странность» необходимо два корпуса текстов. Возможны два варианта:

a. Первый текст нейтральный, второй - тональный (положительный или отрицательный). Таким образом слово несущее тональность (положительную или отрицательную) было бы «странно» встретить в нейтральном корпусе текстов, например, в новостях или обзорных статьях – эмоционально окрашенная лексика встречается в таких текстах крайне редко.

b. Два корпуса текстов с противоположными тональностями. Один положительный, другой - отрицательный. В данном случае ситуация аналогична. «Странно» встретить ярко окрашенную положительную лексику в негативных комментариях о фильме или недавно купленной бытовой технике.

II. Алгоритм

Величина странности слова рассчитывается следующим образом:

a. Рассчитывается вероятность появления слова в исследуемой коллекции

$$P_s(w) = \frac{N_s(w)}{N_s}$$

Где $N_s(w)$ – количество слов w в исследуемой коллекции; N_s – суммарное количество слов в исследуемой коллекции.

b. Рассчитывается вероятность появления слова в исследуемой коллекции

$$P_g(w) = \frac{N_g(w)}{N_g}$$

Где $N_g(w)$ – количество слов w в исследуемой коллекции; N_g – суммарное количество слов в исследуемой коллекции.

c. Так как имеется два текста: один — с высокой концентрацией оценочных слов, другой — с низкой концентрацией оценочных слов (контрастный текст), слова, которые несут оценки, будут «странными» в текстах контрастного корпуса. Сама величина странности для слова w вычисляется так:

$$Weirdness(w) = \frac{P_s(w)}{P_g(w)}$$

d. Слова, попавшие в список, несут в себе такую же тональность, как и у исследуемого корпуса текстов. Таким образом, при сравнении положительного и нейтрального текстов, «странные» слова будут нести положительную тональность. Следовательно, в словаре, построенном по двум корпусам текстов, тональность всех слов одинаковая и совпадает с тональностью исследуемого корпуса текстов.

III. Ограничения, накладываемые на величину $Weirdness(w)$

a. $\forall j: P_g(w_j) > 0$

b. $\forall j: P_s(w_j) \geq 0$. Но, фактически, если $\exists j: P_s(w_j) = 0 \Rightarrow Weirdness(w_j) = 0$. В дальнейшем слова ранжируются по величине $Weirdness(w_j)$ от максимального до минимального значения. Слова, для которых $Weirdness(w_j) = 0$, не представляют интереса, так как встречаются только в одном из корпусов текстов. Принимаются во внимание только слова, встречающиеся в обоих корпусах текста, то есть $\forall j: P_s(w_j) \neq 0$ и $P_g(w_j) \neq 0$. Поэтому используется более строгое ограничение $P_s(w_j) > 0$

c. Следовательно (из пунктов a и b), $\forall w: Weirdness(w) \in (0; +\infty)$

d. $\forall j: y(w_j)$ = {-1;0;+1}

IV. Результаты работы программы

Результатом работы алгоритма является словарь, в котором каждому слову поставлены в соответствие следующие величины: частота данного слова в первом корпусе текстов, частота слова во втором корпусе текстов, странность слова, нормализованная (по максимальной величине странности среди всех слов) странность слова, тональность:

$$w_j \rightarrow [P_g(w_j), P_s(w_j), weirdness(w_j), y(w_j)]$$

В результате работы метода получены словари странности слов для нескольких предметных областей. Построенный таким образом словарь тональности может использоваться для классификации на любое число классов.

3.2. Метод классификации текстов на основе странности слов

Например, возможен следующий вариант алгоритма:

Подсчитывается сентимент значение текста. Происходит это следующим образом: Сентимент величина текста — это сумма по всем словам в данном тексте, каждый член этой суммы есть произведение сентимент значения слова, умноженное на тэг класса (тональность) слова.

$$W_i = \sum_{\forall w_j \in W_i} y(w_j) * weirdness(w_j)$$

где W_i– сентимент величина текста i; $weirdness(w_j)$ –"Странность" слова j; $y(w_j)$ – ключ класса (тональность), к которому относится слово j. $y(w_j) = \{-1, 0, +1\}$.

Таким образом, величина W позволяет отсортировать тексты согласно значению их тональности.

Следующий шаг – определение цены деления шкалы, по которой ранжируются документы. В рассматриваемой модели все полученные промежутки имеют одинаковую длину. Возможное улучшение модели – более сложная система подсчета W^*

$$W^* = \frac{Max(W_i) - Min(W_i)}{|C|}$$

W^* - длина отрезка (на шкале распределения W), на котором лежат документы из одного класса, $|C|$ - количество классов

Последний шаг – отнесение каждого текста определенному классу (k) согласно условию:

$$W_i \in c_k \Leftrightarrow (k-1) \cdot W^* < W_i < k \cdot W^*$$

В результате все документы распределены по шкале W, что позволяет отнести каждый документ к определенному классу.

3.3. Классификация методами машинного обучения с использованием словаря тональности.

Идея данного метода состоит в следующем. Необходимо представить каждый документ в виде вектора, состоящего из булевых переменных, которые принимают истинные значения, если слово w_j из словаря тональности присутствует в документе, и ложное значение в противном случае. Или формально:

Пусть имеется коллекция документов $Q = \{d_1 \dots d_{|Q|}\} \subseteq D$, где D – множество документов. Тогда

$$\forall i: d_i = \{n_i(w_1), \dots, n_i(w_\alpha)\}$$

Где $n_i(w_j) = \begin{cases} 1, \ if \ w_j \in d_i \\ 0, \ if \ w_j \notin d_i \end{cases}$, $d_i \in Q$- документ i из исходной коллекции документов.

Далее применяется наивный метод Байеса, где признаки документа $n_i(w_j)$ задаются описанным выше образом.

Выводы и результаты

В главе три приводится обзор методов, предложенных и реализованных в работе. Проводится анализ алгоритмов. Предложенные методы:

1. Автоматического построения словаря тональности
2. Классификации текстов на основе полученного словаря тональности.
3. Классификация текстов с применением методов машинного обучения, где качестве признаков выступают слова из словаря тональности.

Глава 4. Методология

Схема исследования

В данной работе использовалась следующая схема анализа данных (представлена на рисунке 4.1):

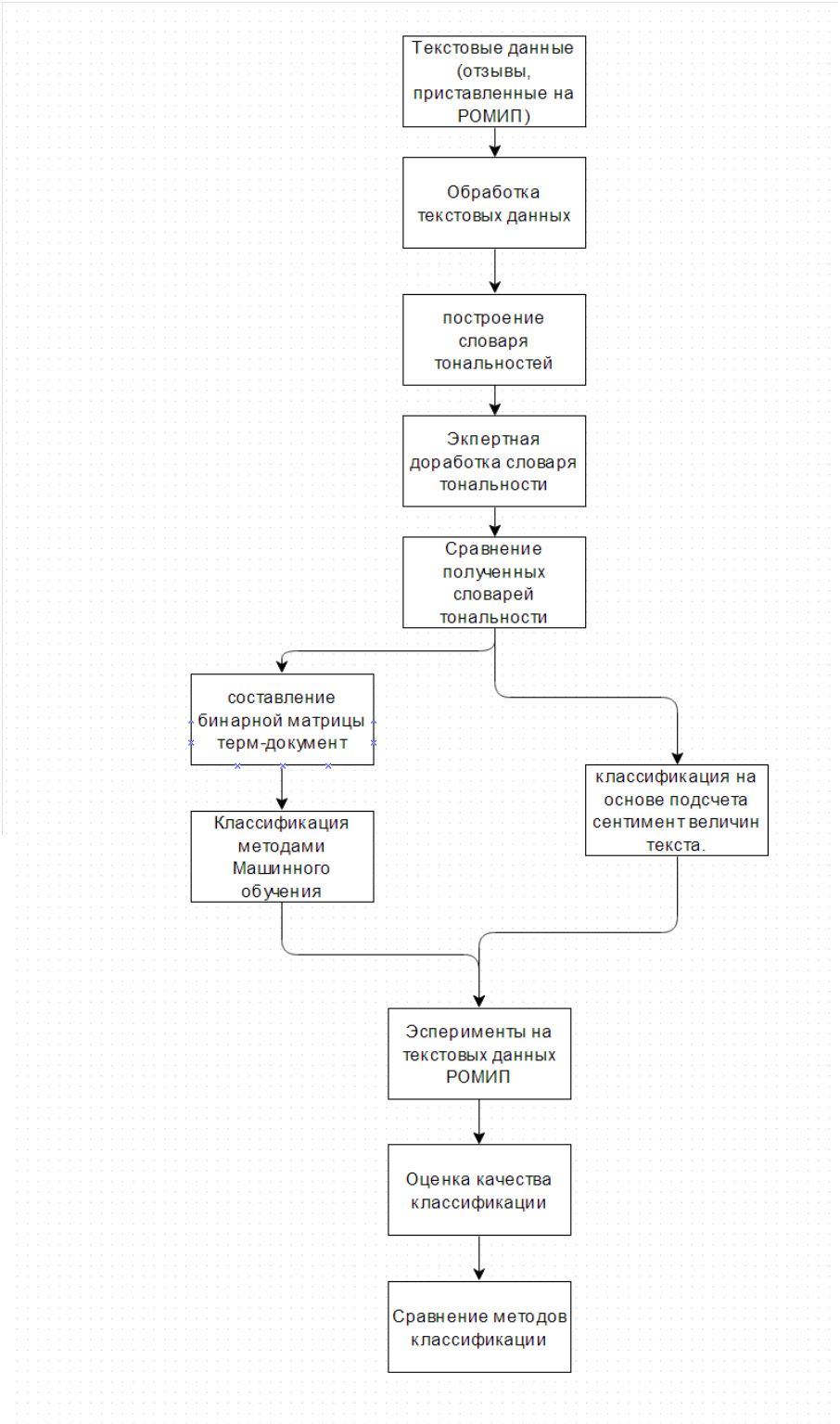

рисунок 4.1. схема методологии, применяемая в работе

Рассмотрим подробнее каждый элемент схемы:

1. Тестовые данные, используемые в работе, были представлены на Российском семинаре по Оценке Методов Информационного Поиска (РОМИП). Данные размечены экспертно на три класса – нейтральный, позитивный и негативный;
2. Обработка данных. Каждый текст был представлен в виде набора входящих в него слов (термов). После чего была произведена нормализация текста, описанная в разделе 1.3.1:
    a. Слова текста приведены к нормальной форме. Для этого использовалась библиотека PyMorphy2 на языке программирования Python;
    b. Удаление стоп-слов. Слова, имеющие длину менее 4 символов были удалены, как заведомо не являющиеся оценочными. Данная операция позволила существенно сократить время работы алгоритма построения словаря тональности.
3. Построение словаря тональности. Каждому слову поставлена в соответствие его частота в тексте. После чего эта величина нормируется. Вычисляется величина Weirdness для каждого слова, согласно методу, описанному в главе 3 разделе 3.1.
4. Дополнительная обработка словаря тональности. Данный этап включает в себя два шага:
    a. Экспериментальное определение порогового значения в словаре тональности, позволяющее существенно уменьшить размер словаря в основном за счет удаление слов, не несущих эмоциональную оценку (шумов). Более подробно данный этап описан в главе 5 раздел 3;
    b. Дополнительная экспертная разметка слов, входящих словарь. Данная операция позволяет свести зашумленность словаря к абсолютному минимуму.

5. Сравнение словарей тональности, которое позволяет определить плотность оценочных слов в словаре и оценить качество полученных словарей.
6. Классификация на основе агрегированного сентимент-значения слов, входящих в документ. Данный алгоритм описан более подробно в главе 3 раздел 3.2.
7. Альтернативный вариант– **классификация методами машинного обучения** с использованием словаря тональностей. Слова, входящие в словарь тональности выступают в качестве атрибутов классификации. для применения такого метода необходимо составить матрицу терм-документ. В данной работе значения атрибутов являются бинарными. Данный подход описан более подробно в главе 3 раздел 3.3.
8. Классификация одним из методов машинного обучения, с использованием матрицы построенной в пункте 7.
9. Эксперименты на текстовых данных РОМИП. Тестирование методов для различных наборов входных данных: словарей тональности и документов, принадлежащих разным предметным областям.
10. Сравнение методов на основе полученных результатов. Анализ полученных результатов и формулировка выводов.

Выводы и результаты

В данной главе была приведена схема, используемая в работе. Описаны основные шаги анализа. В следующей главе будут рассмотрены эксперименты, сделаны выводы на основе полученных результатов.

Глава 5. Эксперименты

Работа с данными

В данной работе используются следующие данные:

1. Отзывы о книгах с сайта imhonet.ru
2. Отзывы о фильмах с сайта imhonet.ru
3. Отзывы о камерах с сайта яндекс-маркет
4. Посты пользователей сайта livejournal
5. Статьи сайта Wikipedia

Данные, представляющие собой отзывы о книгах, фильмах и камерах были представлены на Российском семинаре по оценке методов информационного поиска (РОМИП). Эти данные размечены экспертами вручную, то есть произведено отнесение каждого текста к конкретному классу.

Отзывы о фильмах, книгах, камерах представляют собой тексты с высоким содержанием эмотивной лексики, то есть лексики, несущей определенную тональность.

Тексты, содержащие оценочную лексику в каждой из областей, разделены на три части – нейтральные, негативные и позитивные. В дальнейшем в каждом из этих разделов выделены два класса – обучающие и тестирующие тексты. Их доли составляют 90% и 10% соответственно.

Таблица 5.1. Документы, разделенные по предметным областям и по тональности

	cameras	moovies	books	∑
+	8 161	10 459	17 625	36 245
0	946	3 252	3 232	7 430
-	1 104	1 106	1 250	3 460
∑	10 211	14 817	22 107	**47 135**

Таблица 5.2. Документы, разделенные по предметным областям, по тональности, и с дополнительной разбивкой на тестовую и обучающую выборку.

	%	cameras	moovies	books
+	90%	7 345	9 413	15 863
	10%	816	1 046	1 763
0	90%	851	2 927	2 909
	10%	95	325	323
-	90%	994	995	1 125
	10%	110	111	125

Кросс-валидация

Для более точной оценки качества классификации, используется метод кросс-валидации (cross-validation), который усредняет показания метрик качества по различным разбиениям исходного множества документов. Тем самым становится возможным сгладить возможные шумы отдельных документов и сделать и получить результат, не зависящий от способа разбиения исходного множества на тестовую и обучающую выборки.

Алгоритм выглядит следующим образом:

1. Исходное множество документов, обладающее мощностью *n* разбивается на k непересекающихся подмножеств одинакового размера $Q = \{q_1, \dots, q_k\}$, где k – параметр кросс- валидации.
2. Для $i = 1,\dots,k$
   - q_i принимается тестовым множеством, $\bigcup_{j \neq i} q_j$ – обучающим множеством.

- Алгоритм запускается, выполняется классификация документов, определяется качество классификации.

3. Для проведенных k тестов вычисляются усредненные метрики качества работы классификатора.

В процессе построения словарей тональности нет необходимости делить коллекцию текстов на обучающую и тестируемую, так как методы классификации с использованием словаря тональности не нуждаются в обучающей выборке. Таким образом, метод классификации описанный в главе 3 (раздел 3.2) работал со всей коллекцией документов как с тестовой выборкой. Для второго алгоритма, основанного на методе наивной классификации Байеса, применялся метод кросс-валидации.

Словари тональности

Для построения словарей тональности использовался метод странности слов, описанный в главе 3 (раздел 3.1). Словари строились для трех предметных областей: обзоры пользователей по купленным фотокамерам, отзывы о фильмах, отзывы о книгах. Также был построен словарь тональности по предметной области, состоящей из объединения отзывов о фильмах, камерах и книгах.

Данные используемые в приведенном методе – это тексты, которые были представлены на Российском семинаре по оценке методов информационного поиска (РОМИП) и разделены экспертами на три класса – позитивный, негативный и нейтральный.

Согласно описанию алгоритма нахождения странности слов, можно использовать два разных набора входных данных:

(i) два корпуса текстов противоположной тональности;

(ii) тональный корпус текстов в купе с нейтральным корпусом текстов.

Эксперименты показали, что полученные словари чувствительны к шумам, что, безусловно является их существенным недостатком. Построенные списки слов требуют ручной постобработки, что достаточно трудозатратно в силу большого размера полученных словарей (5000-6000 слов).

Без ручной постобработки в словарях присутствует достаточно большое количество посторонних слов. То есть слов, не несущих в себе тональность и не являющихся оценочными. Примеры тональных и нетональных («шумов») слов в полученном словаре тональности представлены в таблице 5.3.

Таблица 5.3. Фрагменты словарей тональности, включающие в себя как оценочные слова, так и «шумы».

word	probability1	probability2	weirdness	weirdness normalised	class tag	value
непонятный	7,47E-05	0,000432952	19,706	0,696	-1	-0,696
примитивный	3,73E-05	0,000216476	19,732	0,697	-1	-0,697
безобразие	1,87E-05	9,62E-05	17,491	0,618	-1	-0,618
иностранный	1,87E-05	9,62E-05	17,491	0,618	-1	-0,618
порция	1,87E-05	9,62E-05	17,491	0,618	-1	-0,618
злобный	1,87E-05	9,62E-05	17,491	0,618	-1	-0,618
нелепость	1,87E-05	9,62E-05	17,491	0,618	-1	-0,618
увлекательный	0,000317283	4,81E-05	17,150	0,606	1	0,606
полицейский	0,000634565	9,62E-05	17,150	0,606	1	0,606
блистательный	0,000167973	2,41E-05	18,122	0,640	1	0,640
блестящий	0,000167973	2,41E-05	18,122	0,640	1	0,640
ангел	0,000167973	2,41E-05	18,122	0,640	1	0,640
душераздирающий	0,000167973	2,41E-05	18,122	0,640	1	0,640

Тональные слова распределены в словаре неравномерно. Чем больше сентимент величина слова (например, в данной работе это странность), тем чаще встречаются в словаре оценочные слова. При уменьшении сентимент величины (то есть при «движении» вниз по списку) частота появления оценочных слов в словаре начинает падать. Поэтому имеет смысл ввести пороговое значение α, и удалить слова w_j: $weirdness(w_j) < weirdness(\alpha)$.

Ниже приведены результаты экспериментального определения оптимального значения α для словаря, полученного в работе [1] и последующее применение полученного результата для словарей, построенных в данной работе.

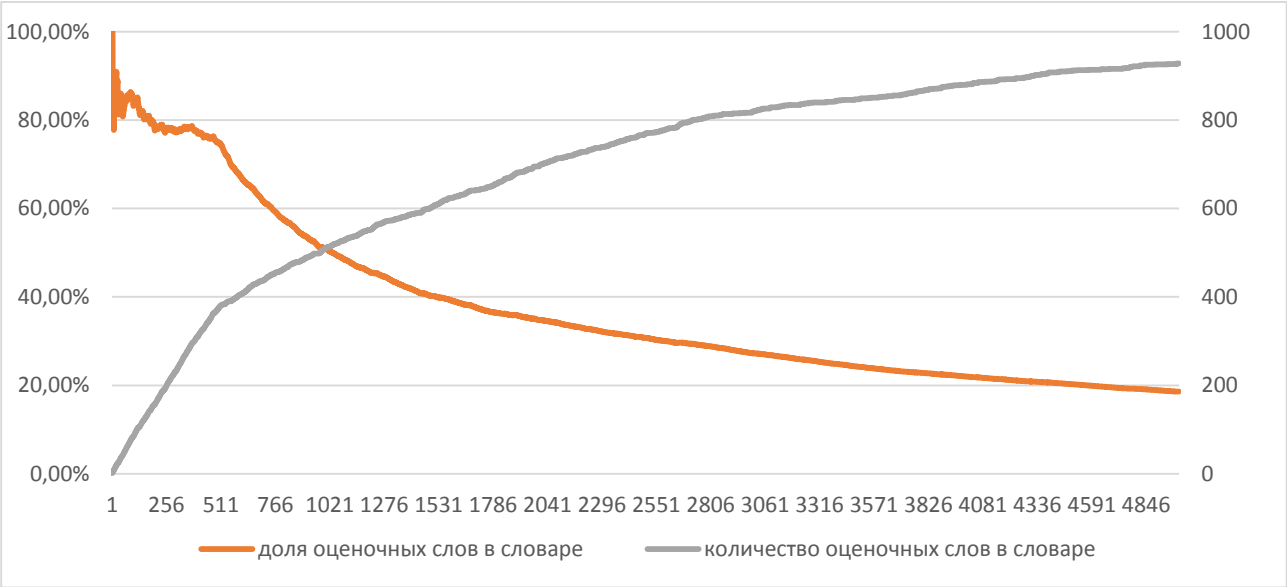

График 5.1. Распределение оценочных слов в словаре [1] на отрезке [1-5000]

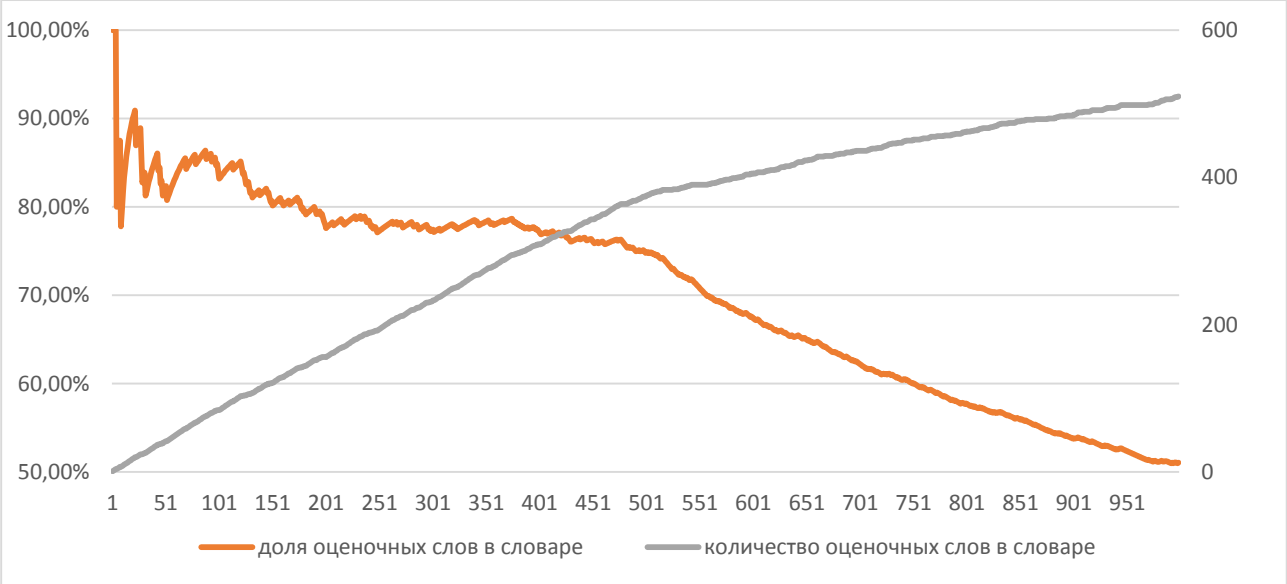

График 5.2. Распределение оценочных слов в словаре [1] на отрезке [1-1000]

Таблица 5.4. Соотнесение сентимент-величины и плотности слов в словаре тональности [1]

Row number	Процент оценочных слов	Value
4	100	0.933
25	87.50	0.878
250	77.11	0.656
350	78.22	0.582
478	76.31	0.528
517	74.22	0.516
600	66.61	0.490

Согласно полученным результатам оптимальное значение $\alpha \sim 0.52$. Таким образом, построенные в работе словари тональности имели $\alpha = 0.52$ в качестве порогового значения. Значение α может незначительно отклоняться в силу неравномерности распределения оценочных слов в конкретно взятом словаре тональности.

Проведена ручная проверка качества составления словарей. Среди построенных словарей наибольшей плотностью оценочных слов обладает словарь, построенный по предметной области, объединяющей отзывы по книгам, фильмам и камерам. Для этого словаря была проведена экспертная доработка всех слов, лежащих выше порогового значения α.

В дополнение к полученным словарям тональности был задействован словарь оценочных слов, представленный работе И. Четверкина [1]. Данный словарь не предполагает отнесение слова к конкретному классу тональности (отрицательный, положительный, или нейтральный). Была проведена соответствующая доработка словаря, то есть каждое слово в словаре было отнесено к определенному классу тональности.

Ниже представлена таблица с промежуточными итогами, по процентному содержанию тональных слов в словарях тональности для weirdness > α, где α ~ 0.52. Словари строились для каждой предметной области двумя способами (два разных набора входных данных):

(i) два корпуса текстов противоположной тональности;

(ii) тональный корпус текстов в купе с нейтральным корпусом текстов.

Таблица 5.5. Соотнесение сентимент-величины и плотности слов в построенных словарях тональности

	(i)	(ii)	α
Словарь Четверкина И. [1] – *ручная разметка*	*0.742*		**0,52**
Область книги	0.541	0.567	0,8
Область фильмы	0.517	0.509	0,8
Область камеры	0.458	0.606	0,8
Область книги ∪ фильмы ∪ камеры	**0.624**	0.593	**0,515**

Стоит отметить, что такое «усечение» словаря позволяет существенно снизить количество признаков при классификации наивным байесовским методом, в котором в качестве признаков выступают слова из словаря тональности, присутствующие в документе.

Тесты. Сравнение методов.

На базе полученных словарей тональности построен классификатор, механизм работы которого описан в главе 3 (раздел 3.2). Классификатор

реализован на языке программирования Python с использованием библиотеки pymorphy2.

Количество классов - три. Положительный, отрицательный и нейтральный. Количество классов обусловлено тем, что данные (отзывы о фильмах, книгах и камерах), на которых тестировался алгоритм классификации, разбиты экспертно на три класса.

Также на базе полученных словарей тональности построен классификатор, механизм работы которого описан в главе 3 (раздел 3.3). В классификаторе используется комбинированный подход, объединяющий словари тональности и метод машинного обучения (Наивный метод Байеса, Random Forest, Support Vector Machines (SVM)). Для тестирования данных методов применялся пакет Weka.

Пусть в модели действуют следующие обозначения:

- **I** - Словарь, построенный автоматически по текстам из объединения трех предметных областей – фильмы, книги, камеры.
- **II** – словарь построенный в п. I, с последующей дополнительной экспертной разметкой.
- **III** - Словарь, предложенный в работе И. Четверкина [1]. Данный словарь содержит список оценочных слов, извлеченных из коллекций отзывов в нескольких предметных областях (фильмы, книги, игры, телефоны, камеры)
- **W** - Алгоритм, описанный в главе 3 (раздел 3.2). Данный метод основан на использовании словарей тональности для классификации документов.
- **NB** – алгоритм наивной байесовской классификации, описанный в главе 3 (раздел 3.3). Данный метод использует комбинированный подход, применяется наивный метод байесовской классификации, с использованием признаков документа – слов из словаря тональности.

- **RF** – алгоритм классификации методом Random Forest, описанный в главе 2 (разделы 2.2 и 2.3). Данный метод использует комбинированный подход, применяется наивный метод байесовской классификации, с использованием признаков документа – слов из словаря тональности.
- **SVM** – алгоритм классификации методом опорных векторов, описанный в главе 2 (раздел 2.4). Данный метод использует комбинированный подход, применяется наивный метод байесовской классификации, с использованием признаков документа – слов из словаря тональности.

Результаты работы классификатора W представлены в таблице 5.6.

Таблица 5.6. Результаты тестов

словарь	method	Предметная область	Accuracy	Macro_P	Macro_R
I	W	книги	0,556	0,371	0,439
II	W	книги	0,637	0,293	0,315
III	W	книги	0,651	0,512	0,449
I	W	фильмы	0,586	0,391	0,604
II	W	фильмы	0,660	0,440	0,455
III	W	фильмы	0,679	0,453	0,468
I	W	камеры	0,573	0,291	0,495
II	W	камеры	0,611	0,387	0,401
III	W	камеры	0,643	0,429	0,443

Из таблицы отчетливо виден тренд, согласно которому словарь I на каждой предметной области показал результаты значительно хуже, чем словари II и III, что говорит о низкой эффективности автоматического метода и необходимости использования ручной разметки. Вместе с этим, словарь III показал лучшие результаты, чем словарь II. Стоит отметить, что на областях «книги» и «фильмы» разница между II и III небольшая (0,014 и 0,019), в то

время как на «фильмах» словарь III показал больший отрыв (0,032). Лучший результат был показан на таком наборе данных классификатором, работающим на основе словаря III с корпусом текстов из области «фильмы» - 0,679. Стоит отметить, что показатель Macro_Recall наивысший у словаря I.

В связи с полученными результатами принято решение, отказаться от словаря I. Данный словарь заведомо более зашумлен, чем два остальных, которые вручную отчищены от шумов.

Далее были проведены тесты на словарях II и III для трех методов машинного обучения. Результаты приведены в таблице 5.7 и таблице 5.8.

Таблица 5.7. Результаты тестов ML

словарь	method	Предметная область	Accuracy	Macro_P	Macro_R
II	RF	книги	0,647	0,453	0,513
III	RF	книги	0,712	0,469	0,293
II	SVM	книги	0,659	0,561	0,382
III	SVM	книги	0,693	0,611	0,497
II	NB	книги	0,696	0,352	0,284
III	NB	книги	0,682	0,455	0,410
II	RF	фильмы	0,763	0,710	0,553
III	RF	фильмы	0,773	0,604	0,541
II	SVM	фильмы	0,736	0,439	0,397
III	SVM	фильмы	0,679	0,546	0,458
II	NB	фильмы	0,694	0,463	0,479
III	NB	фильмы	0,671	0,596	0,463
II	RF	камеры	0,653	0,623	0,546
III	RF	камеры	0,707	0,587	0,416
II	SVM	камеры	0,611	0,713	0,611
III	SVM	камеры	0,721	0,558	0,650
II	NB	камеры	0,655	0,437	0,452

III	NB	камеры	0,714	0,537	0,612

Таблица 5.8. Результаты тестов ML-средние показатели

словарь	method	Предметная область	Accuracy	Macro_P	Macro_R
II	ML-average	книги	0,681	0,455	0,393
III		книги	0,696	0,512	0,400
II		фильмы	0,731	0,537	0,476
III		фильмы	0,708	0,582	0,487
II		камеры	0,640	0,591	0,536
III		камеры	0,714	0,561	0,559

В данном случае ситуация не так очевидна. Словарь III в большинстве случаев дает результат лучше, чем словарь II. В частности, по усредненным результатам на разделах «книги» и «камеры» точнее была классификация с применением словаря III, в то время как в разделе «фильмы» большую точность показал словарь II. Это можно объяснить следующим образом - словарь III строился на текстах из 5 предметных областей (фильмы, книги, игры, телефоны, камеры), таким образом при примерно одинаковом количестве слов в словарях II и III, словарь III обладает большим набором «технических» оценочных слов, в то время как в словаре II чаще встречаются «чувствительные» оценочные слова, характерные для описания фильмов и книг.

Классификация методом Байеса показала себя хуже, чем методы RF и SVM, превзойдя данные методы в одном тесте из шести (1/6)- область камеры, словарь II- показав при этом точность практически равную методу RF. SVM показал результаты, лучше других на двух тестах из шести (2/6), в то время как RF на трех тестах из шести (3/6).

Сравнивая результаты тестовых запусков на словарях II и III для методов комбинированного подхода и метода, использующего только словари

тональности, можно сделать вывод, что методы машинного обучения лучше справились с задачей, превзойдя показатели «Accuracy» в среднем на 4,8%, «Macro_P» в среднем на 12%, «Macro_R» в среднем на 5,4%. Вместе с этим необходимо отметить, что алгоритм классификации W лучше справился с задачей на отдельных наборах входных параметров и данных, чем некоторые отдельно взятые алгоритмы машинного обучения. Основываясь на этом можно сделать **вывод**, что методы машинного обучения в данной работе значительно превзошли метод классификации, основанный на использовании исключительно словарей тональности.

Выводы и результаты

В главе 5

- Описаны данные и предметные области, на которых тестировались алгоритмы, используемые в работе.
- Проведено исследование оптимального размера словаря тональности, определяемое с помощью сентимент - значения, используемого в словаре (например, величины странность).
- В главе проведен сравнительный анализ алгоритмов, описанных в главе три. Методы применились на основе построенного и оптимизированного словаря тональности, а также на размеченном вручную словаре из работы [1].
    - Произведено сравнение методов машинного обучения при работе со словарем тональностей как с набором атрибутов, сделаны выводы по качеству работы классификатора;
    - Алгоритм W протестирован на трех различных словарях, сделаны выводы по эффективности работы алгоритма;
    - Произведено сравнение методов машинного обучения с методом W на основе результатов тестов.

Заключение

В данной работе производен сравнительный анализ алгоритмов машинного обучения, принимающих в качестве параметров классификации слова из словаря тональностей, и методов, использующих исключительно словарь тональностей. Более детально:

- Построены словари тональности на различных наборах входных данных, произведена оценка качества полученных словарей;
- Произведено сравнение методов машинного обучения при работе со словарем тональностей как с набором атрибутов, сделаны выводы по качеству работы классификатора;
- Алгоритм W протестирован на трех различных словарях, сделаны выводы по эффективности работы алгоритма;
- Произведено сравнение методов машинного обучения с методом W на основе результатов тестов.

В работе использовались данные, представлены на Российском семинаре по оценке методов информационного поиска (РОМИП), экспертно размеченные на три класса – нейтральный, отрицательный и положительный. В работе предложены алгоритмы и методы классификации текстов на несколько классов. Тестирование алгоритмов проводилось для количества классов, равного трем. Это связано с тем, что данные, на которых тестировались методы, были размечены на три класса.

Получены следующие результаты:

- Экспериментально установлен оптимальный порог, позволяющий произвести сокращение размерности словаря (уменьшение количества слов в словаре);
- Установлено, что для существенного улучшения качества словаря, для существенного увеличения плотности эмоционально окрашенной лексики

в словаре необходимо производить дополнительно ручную разметку. Словари, построенные предложенным методом [2] сильно разрежены с точки зрения содержания оценочных слов. Даже после экспертной разметки качество словаря оставляет желать лучшего. Количество оценочных слов в наилучшем словаре тональности составила ~62% после отсечения части словаря приведенным выше методом.

  - Стоит также отметить, что словарь, обладавший наибольшей концентрацией эмоционально окрашенной лексики, был получен на коллекции текстов, включающей в себя тексты всех трех предметных областей. Это связано с небольшим объемом коллекции документов из отдельно взятой предметной области.

- Тестирование метода подсчета сентимент значений текстов W производилось на трех словарях. Два из которых были получены в работе (один с дополнительной экспертной разметкой, другой-без дополнительной разметки), третий взят за основу в работе [1] и экспертно размечен. Тесты показали, что алгоритм подсчета сентимент значений текстов справляется с задачей классификации лучше, используя словарь III. Незначительно уменьшается качество классификации, при использовании словаря II. Словарь I показал худший результат (со значительным отставанием от второго места) и был исключен из дальнейшего рассмотрения.
- Согласно полученным результатам, качество работы методов машинного обучения зависит от словаря тональности и предметной области. Тем не менее, алгоритмы Random Forest и Support vector machine в среднем показали более качественные результаты, чем алгоритм наивной байесовской классификации, который, однако, на некоторых наборах входных данных показал более высокое качество классификации, чем два первых метода. Данные эксперименты еще раз подтвердили зависимость задачи классификации от предметной области – один и тот же алгоритм по-разному справлялся с задачей на разных входных данных.

- В работе было произведено сравнения качества классификации методов машинного обучения и метода подсчета сентимент значения документов. В среднем более высокое качество классификации показали методы машинного обучения с преимуществом до 12% в зависимости от метрики. При более детальном рассмотрении можно заметить, что для некоторых наборов входных данных методы подсчета сентимент значения текста оказываются более качественными, что еще раз подтверждает зависимость результата классификации от предметной области.

Также в рамках работы реализованы методы

1. автоматического построения словаря тональности,
2. алгоритма классификации на основе подсчета странности слов.

Данные методы реализованы на языке Python с использованием библиотеки pymorphy2.

Данная работа производит сравнительный анализ алгоритмов классификации на основе словаря тональности, полученного автоматически, с последующей доработкой и без нее. Сравнивались алгоритмы подсчета сентимент величины текста и методы машинного обучения. Классификация производилось на три класса.

Предложенные в работе методы возможно использовать для классификации на любое число классов. Таким образом, возможное продолжение работы - *тестирование алгоритмов при классификации на большее число классов* (например, пять). Также возможна *разработка нового или усовершенствование существующего алгоритма* построения словаря тональности, с целью улучшения качества словаря и увеличения плотности содержания оценочных слов.

Также возможно усовершенствовать метод подсчета сентимент значения текста, что позволит более точно определять тональность документа, исходя из слов, входящих в данный текст.

Список литературы

[1] Chetviorkin I. I. , Loukachevitch N. V. Extraction of Russian Sentiment Lexicon for Product Meta-Domain // In Proceedings of COLING 2012: Technical Papers , pages 593–610

[2] Четверкин И. И. - Автоматизированное формирование базы знаний для задачи анализа мнений // Автореферат диссертации на соискание учёной степени кандидата физико-математических наук. Москва, 2014

[3] Liu, B.: Sentiment Analysis and Subjectivity. Handbook of Natural Language Processing, 2010

[4] Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis// Foundations and Trends® in Information Retrieval: Vol. 2: No 1–2, pp 1-135. 2008.

[5] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M. Lexicon-based methods for sentiment analysis// Computational Linguistic, Volume 37 Issue 2, p. 267-307, 2011

[6] Pang B., Lee L. Thumbs up? Sentiment classification using machine learning techniques // Proc. of the Conferenceon Empirical Methods in Natural Language Processing (EMNLP). Philadelphia: ACL, 2002. 79–86

[7] К.В. Воронцов. Математические методы обучения по предедентам (теория обучения машин).

[8] Taboada M., Brooke J., Tofiloski M., Voll K., Stede M. (2011), Lexicon based methods for sentiment analysis, Computational Linguistics, Vol. 37(2), pp. 267–307.

[9] Ding X., Liu B., Yu P. S. (2008), A holistic lexicon based approach to opinion mining, Proceedings of the Conference on Web Search and Web Data Mining (WSDM), pp. 231–240. Hu

[10] Hu M., Liu B. (2004), Mining and summarizing customer reviews, Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2004), Seattle, pp. 168–177.

[11] Turney P. (2002), Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews, Proceedings of the Association for Computational Linguistics (ACL), pp. 417–424.

[12] Sebastiani F. (2002), Machine learning in automated text categorization, ACM Computing Surveys, Vol. 34, pp. 1–47.

[13] Agarwal A., Xie B., Vovsha I., Rambow O., Passonneau R. (2011), Sentiment analysis of twitter data, Proceedings of the Workshop on Language in Social Media (LSM 2011), pp. 30–38.

[14] Go A., Bhayani R., Huang L. (2009), Twitter sentiment classification using distant supervision, Association for Computational Linguistics, pp. 30–38.

[15] Saif H., He Y., Alani H. (2012), Alleviating data sparsity for twitter sentiment analysis, Workshop: The 2nd Workshop on Making Sense of Microposts (#MSM2012): Big things come in small packages at World Wide Web (WWW), Lyon, France.

[16] He Y. (2012), Incorporating sentiment prior knowledge for weakly supervised sentiment analysis, ACM Transactions on Asian Language Information Processing, Vol. 11(2).

[17] Lewis, David D, 1998, Naive (Bayes) at forty: The independence assumption in information retrieval. In Proe, of the European Conference on Machine Learning (ECML) p, 4-15

[18] Liu, Bing, 2011, Web Data Mining: Exploring Hyperlinks, Contents and Usage Data. Springer, Berlin,

[19] Kobavashi, Nozomi and Inui, Kentaro and Matsumoto, Yuji, 2007, Extracting aspeet-evaluation and aspeet-of relations in opinion mining. Proceedings of EMLP’07

[20] Laffertv, John and Andrew MeCallum and Fernando Pereira, 2001, Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of ICML’01, p. 282-289

[21] Charles Elkan. The Foundations of Cost-Sensitive Learning// Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2. 2001

[22] Caruana R., Niculescu-Mizil A., An Empirical Comparison of Supervised Learning Algorithms Using Different Performance Metrics

[23] Rudy Prabowo, Mike Thelwall. Sentiment Analysis: A Combined Approach// Journal of Informetrics 3(2), 143-157. 2009

[24] Alaa Hamouda, Mahmoud Marei, Mohamed Rohaim. Building Machine Learning Based Senti-word Lexicon for Sentiment Analysis// Journal of Advances in Information Technology, Vol 2, No 4 (2011), 199-203, Nov 2011.

[25] Четверкин И. И. , Лукашевич Н. В. Тестирование систем анализа тональности на семинаре РОМИП-2012 // Т. 2: Доклады специальных секций РОМИП — М.: Изд-во РГГУ, 2013.

[26] Erik Boiy, Marie-Francine Moens. A Machine Learning Approach to Sentiment Analysis in Multilingual Web Texts// Information Retrieval, Volume 12, Number 5 (2009), 526-558.

[27] Andrew McCallum, Kamal Nigam. A Comparison of Event Models for Naive Bayes Text Classification// In Proc. of the AAAI-98 workshop on learning for text categorization, 1998.

[28] Chao Chen. An Empirical Study of Learning from Imbalanced Data Using Random Forest// Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference

[29] Liviu P. Dinu and Iulia Iuga. The Naive Bayes Classifier in Opinion Mining: In Search of the Best Feature Set// CICLing 2012.

[30] Shoushan Li, Zhongqing Wang, Guodong Zhou, Sophia Yat Mei Lee. Semi-Supervised Learning for Imbalanced Sentiment Classification// International Joint Conference on Artificial Intelligence, 2011.

[31] Milos Radovanovic, Mirjana Ivanovic. Text mining: approaches and applications// Novi Sad Journal of Mathematics 38(3), 2008, P. 229-233.

[32] Adnan Duric, Fei Song. Feature Selection for Sentiment Analysis Based on Content and Syntax Models// ACL Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, 2011.

Приложение 1

Программа построения словаря тональностей:

```
def get_probabilities(path,h):
    import csv
    import codecs
    import pymorphy2.tokenizers



    ## reading from the file, splitting the words.
    Rfile = csv.DictReader(codecs.open(path, "r","utf-8"),
  delimiter=';')
    words = []
    for row in Rfile:
        words +=
  pymorphy2.tokenizers.simple_word_tokenize(row["text"])
    print("reading - Done!")

    #for i in range(len(words)):
    #    print (words[i])

    ## normalisation
    j = 0
    finish = len(words)
    while j < finish:
        #print(finish, "finish")
        #print(j, "  ", words[j])
        try:
```

```
            words[j] =
pymorphy2.MorphAnalyzer().parse(words[j])[0].normal_form
#.decode('ascii', errors='ignore')
            if len(words[j]) < 4:
                del words[j]
                finish = finish - 1
            else:
                print(j, "  ", words[j])
                j = j + 1
        except UnicodeDecodeError:
            del words[j]
            finish = finish - 1
        except IndexError:
            del words[j]
            finish = finish - 1
    #for i in range(len(words)):
        #print (words[i])
    print("normalisation - Done!")

    ## finding unique words
    unique_words = set(words)

    ## probabilities of each word
    p = dict.fromkeys(unique_words, 0)
    for item in words:
        p[item]+=1./len(words)
    print("probabilities - Done!")
    if h==1:
        Wfile = open("D:\\prob1.csv", "w") #,"utf-8"
```

```
    if h==2:
        Wfile = open("D:\\prob2.csv", "w") #,"utf-8"

    Wfile.write("word;probability\n")
    for key2,value2 in p.items():
        result = str(key2) + ";" + str(value2) +"\n"
        Wfile.write(result)

    print("total words", len(p))
    print("\n")

    #print("Complete!\n")

    return(p)


def both_probabilities(f_set,f_dict):
    ## comparing set of words and dictionary.
    ## delete items from dictionary which are not contained in
   set
    for value,key in f_dict.items():
        flag=0;
        for item in f_set:
            if(item==value):
                flag=1
        if flag==0:
            del f_dict[value]
    print("new total words", len(f_dict))
    return(f_dict)
```

```
def main():

    prob1 = get_probabilities("D:\\2.csv",1)
    prob2 = get_probabilities("D:\\3.csv",2)

    set_both = set(dict.keys(prob1)) & set(dict.keys(prob2))

    prob1 = both_probabilities(set_both,prob1)
    prob2 = both_probabilities(set_both,prob2)

    #for key,value in prob1.items():
    #    print (key, value)
    #print("\n")
    #for key,value in prob2.items():
    #   print (key, value)


    print("writing to the file !\n")
    ##writing to the file
    Wfile = open("D:\\4.csv", "w") #,"utf-8"
    Wfile.write("word;probability1;probability2;weirdness\n")
    for key1, value1 in prob1.items():
        #flag=0;
        for key2,value2 in prob2.items():
            if(key1==key2):
                #flag=1
                weirdness = value1 / value2
```

```
                result = str(key1) +";" + str(value1) +";" +
    str(value2) + ";" + str(weirdness)+"\n"
                Wfile.write(result)

    print("Complete!\n")
if __name__ == "__main__":
    main()
```

Приложение 2

Программа классификации на основе словаря тональностей:

```
def main(path,path_dict):
    import csv
    import codecs
    import pymorphy2.tokenizers


    ## reading from the file, splitting the words.
    Rfile = csv.DictReader(codecs.open(path, "r","utf-8"),
   delimiter=';')
    dict1 = csv.DictReader(codecs.open(path_dict, "r","utf-
   8"), delimiter=';')

    new_f = open("D:\\res.csv", "w")
    new_f.write("text;W;class\n")

    #s_words = dict.fromkeys(dict1["word"], dict1["value"])

    for row in Rfile:
        W=0
        txt_class=0

        words =
   pymorphy2.tokenizers.simple_word_tokenize(row["text"])
        j = 0
        print("\n")
```

```
        finish = len(words)
        while j < finish:
            try:
                words[j] =
pymorphy2.MorphAnalyzer().parse(words[j])[0].normal_form
                #print(words[j])
                #print( "len", len(words[j] ))
                if len(words[j])< 4:
                    del words[j]
                    finish = finish - 1
                    print("if")
                    #print(j)
                else:
                    print("else")
                    for row_dict in dict1:
                        if words[j]==row_dict["word"]:
                            print("case")
                            W += float(row_dict["value"])
                    if W < -0.5:
                        txt_class=-1
                    if W > 0.5:
                        txt_class=1
                    j = j + 1

            except UnicodeDecodeError:
                del words[j]
                finish = finish - 1
                print("ex1")
            except IndexError:
```

```
                del words[j]
                finish = finish - 1
                print("ex2")



                #print(j)

        result = row["text"] +";" + str(W) +";" +
    str(txt_class)+"\n"
        new_f.write(result)
     print("complite\n")


if __name__ == "__main__":

     main("D:\\texts.csv","D:\\dict.csv")
```

Приложение 3

word	probability1	probability2	weirdness	class tag	value	value_normalised
пронзительный	0,000187	0,0000104	18,0014	1	18,0014	1,000
бред	0,002099	0,000116918	17,9489	-1	-17,9489	-0,997
равнодушный	0,000186	0,0000104	17,9226	-1	-17,9226	-0,996
потрясать	0,001668	0,0000932	17,8963	1	17,8963	0,994
настольный	0,000146	0,0000104	17,8701		0,0000	0,000
незабываемый	0,000186	0,0000104	17,8438	1	17,8438	0,991
гениальный	0,000185	0,0000104	17,8175	1	17,8175	0,990
оттенок	0,000146	0,0000104	17,7912		0,0000	0,000
прикольный	0,000185	0,0000104	17,7650	1	17,7650	0,987
гениально	0,000367	0,0000207	17,7387	1	17,7387	0,985
тупой	0,001036	0,0000585	17,7124	-1	-17,7124	-0,984
вампир	0,001432	0,000116918	17,6861		0,0000	0,000
изящно	0,000184	0,0000104	17,6599	1	17,6599	0,981
дыхание	0,001519	0,000134573	17,6336		0,0000	0,000
супер	0,001277	0,0000725	17,6073	1	17,6073	0,978
взахлёб	0,000364	0,0000207	17,5810	1	17,5810	0,977
сплошной	0,000641	0,0000585	17,5547		0,0000	0,000
влюбиться	0,000545	0,0000311	17,5285	1	17,5285	0,974
наизусть	0,000333	0,0000311	17,5022		0,0000	0,000
тратить	0,001022	0,0000585	17,4759	-1	-17,4759	-0,971
вменяемый	0,000104	0,0000104	17,4496		0,0000	0,000
щемить	0,000181	0,0000104	17,4234	1	17,4234	0,968
добряк	0,000181	0,0000104	17,3971	1	17,3971	0,966
обалденный	0,000181	0,0000104	17,3708	1	17,3708	0,965
кладбище	0,000104	0,0000104	17,3445		0,0000	0,000
совершенство	0,00018	0,0000104	17,3183	1	17,3183	0,962
слабовато	0,00018	0,0000104	17,2920	-1	-17,2920	-0,961
окунуться	0,000104	0,0000104	17,2657		0,0000	0,000
прекрасный	0,000179	0,0000104	17,2394	1	17,2394	0,958
пересмотреть	0,000104	0,0000104	17,2132		0,0000	0,000
масштабность	0,000179	0,0000104	17,1869	1	17,1869	0,955
красивейший	0,000178	0,0000104	17,1606	1	17,1606	0,953
спокойствие	0,000104	0,0000104	17,1343		0,0000	0,000
невозможный	0,000178	0,0000104	17,1080	-1	-17,1080	-0,950
прорисовать	0,000104	0,0000104	17,0818		0,0000	0,000
оторваться	0,002295	0,000134573	17,0555	1	17,0555	0,947
насыщенный	0,000353	0,0000207	17,0292	1	17,0292	0,946
волшебство	0,000352	0,0000207	17,0029	1	17,0029	0,945
противно	0,000993	0,0000585	16,9767	-1	-16,9767	-0,943
явно	0,000528	0,0000585	16,9504		0,0000	0,000
неприятный	0,00099	0,0000585	16,9241	-1	-16,9241	-0,940
кровать	8,32E-05	0,0000104	16,8978		0,0000	0,000
ненавидеть	0,000175	0,0000104	16,8716	-1	-16,8716	-0,937
гипноз	8,32E-05	0,0000104	16,8453		0,0000	0,000
грусть	0,000348	0,0000207	16,8190	-1	-16,8190	-0,934

эвкалипт	8,32E-05	0,0000104	16,7927		0,0000	0,000
плакать	0,001041	0,0000621	16,7665	-1	-16,7665	-0,931
восхитительный	0,000174	0,0000104	16,7402	1	16,7402	0,930
пафосный	0,000174	0,0000104	16,7139	-1	-16,7139	-0,928
реал	8,32E-05	0,0000104	16,6876		0,0000	0,000
удивительный	0,000173	0,0000104	16,6614	1	16,6614	0,926
запах	0,000499	0,0000621	16,6351		0,0000	0,000
изумительный	0,000173	0,0000104	16,6088	1	16,6088	0,923
воссоздавать	8,32E-05	0,0000104	16,5825		0,0000	0,000
незатейливый	0,000343	0,0000207	16,5562	-1	-16,5562	-0,920
умно	8,32E-05	0,0000104	16,5300		0,0000	0,000
сильнейший	0,000172	0,0000104	16,5037	1	16,5037	0,917
мощно	0,000171	0,0000104	16,4774	1	16,4774	0,915
заурядный	0,000171	0,0000104	16,4511	-1	-16,4511	-0,914
противоположный	8,32E-05	0,0000104	16,4249		0,0000	0,000
увлекательность	0,000171	0,0000104	16,3986	1	16,3986	0,911
летом	8,32E-05	0,0000104	16,3723		0,0000	0,000
превосходно	0,00017	0,0000104	16,3460	1	16,3460	0,908
омерзительный	0,00017	0,0000104	16,3198	-1	-16,3198	-0,907
самостоятельно	8,32E-05	0,0000104	16,2935		0,0000	0,000
вдохновлять	0,000169	0,0000104	16,2672	1	16,2672	0,904
прелесть	0,000672	0,0000414	16,2409	1	16,2409	0,902
погибший	0,000169	0,0000104	16,2147	-1	-16,2147	-0,901
необременительный	0,000168	0,0000104	16,1884	1	16,1884	0,899
проникаться	0,000335	0,0000207	16,1621	1	16,1621	0,898
полоть	8,32E-05	0,0000104	16,1358		0,0000	0,000
популярный	0,000942	0,0000585	16,1095	-1	-16,1095	-0,895
цитата	0,000452	0,0000585	16,0833		0,0000	0,000
великолепный	0,001995	0,000124221	16,0570	1	16,0570	0,892
подходить	0,000415	0,0000585	16,0307		0,0000	0,000
бредовый	0,000936	0,0000585	16,0044	-1	-16,0044	-0,889
http	0,001659	0,000233836	15,9782		0,0000	0,000
смех	0,000991	0,0000621	15,9519	1	15,9519	0,886
романтика	0,00033	0,0000207	15,9256	1	15,9256	0,885
божественный	0,000329	0,0000207	15,8993	1	15,8993	0,883
непредсказуемый	0,000146	0,0000207	15,8731		0,0000	0,000
мерзкий	0,000328	0,0000207	15,8468	-1	-15,8468	-0,880
мудрый	0,002784	0,00017598	15,8205	1	15,8205	0,879
нетерпение	0,000208	0,0000311	15,7942		0,0000	0,000
жизненно	0,00049	0,0000311	15,7680	1	15,7680	0,876
плохо	0,002761	0,000175377	15,7417	-1	-15,7417	-0,874
завораживать	0,000651	0,0000414	15,7154	1	15,7154	0,873
ярко	0,00027	0,0000414	15,6891		0,0000	0,000
ужасно	0,001831	0,000116918	15,6629	-1	-15,6629	-0,870
штамп	0,000915	0,0000585	15,6366	-1	-15,6366	-0,869
бессмысленный	0,000913	0,0000585	15,6103	-1	-15,6103	-0,867
запоем	0,000807	0,0000518	15,5840	1	15,5840	0,866
дерьмо	0,001819	0,000116918	15,5577	-1	-15,5577	-0,864
любимый	0,006592	0,000424422	15,5315	1	15,5315	0,863

обманывать	6,24E-05	0,0000104	15,5052		0,0000	0,000
чушь	0,00032	0,0000207	15,4789	-1	-15,4789	-0,860
шестидесятый	6,24E-05	0,0000104	15,4526		0,0000	0,000
ведущий	0,00016	0,0000104	15,4264	1	15,4264	0,857
специфика	6,24E-05	0,0000104	15,4001		0,0000	0,000
улыбнуться	0,000318	0,0000207	15,3738	1	15,3738	0,854
смаковать	0,000318	0,0000207	15,3475	1	15,3475	0,853
разносторонний	0,000159	0,0000104	15,3213	1	15,3213	0,851
отстой	0,000159	0,0000104	15,2950	-1	-15,2950	-0,850
тошнотворный	0,000159	0,0000104	15,2687	-1	-15,2687	-0,848
похвала	0,000316	0,0000207	15,2424	1	15,2424	0,847
залпом	0,000125	0,0000207	15,2162		0,0000	0,000
несравненный	0,000158	0,0000104	15,1899	1	15,1899	0,844
неглупый	0,000158	0,0000104	15,1636	1	15,1636	0,842
кошмар	0,000157	0,0000104	15,1373	-1	-15,1373	-0,841
талантливо	0,000157	0,0000104	15,1110	1	15,1110	0,839
утро	0,000125	0,0000207	15,0848		0,0000	0,000
правильно	0,000468	0,0000311	15,0585	1	15,0585	0,837
раскрытие	6,24E-05	0,0000104	15,0322		0,0000	0,000
безупречно	0,000156	0,0000104	15,0059	1	15,0059	0,834
мурашка	0,000125	0,0000207	14,9797		0,0000	0,000
нелепо	0,000156	0,0000104	14,9534	-1	-14,9534	-0,831
исход	6,24E-05	0,0000104	14,9271		0,0000	0,000
замысловатый	0,000155	0,0000104	14,9008	1	14,9008	0,828
патриотизм	0,000308	0,0000207	14,8746	1	14,8746	0,826
сообщить	6,24E-05	0,0000104	14,8483		0,0000	0,000
печально	0,000154	0,0000104	14,8220	-1	-14,8220	-0,823
заинтересовать	0,000154	0,0000104	14,7957	1	14,7957	0,822
неподражаемый	0,000154	0,0000104	14,7695	1	14,7695	0,820
желанный	0,000153	0,0000104	14,7432	1	14,7432	0,819
советник	6,24E-05	0,0000104	14,7038		0,0000	0,000
тёплый	0,000187	0,0000311	14,6841		0,0000	0,000
занятно	0,000153	0,0000104	14,6644	1	14,6644	0,815
послевоенный	6,24E-05	0,0000104	14,6381		0,0000	0,000
сломанный	0,000152	0,0000104	14,6118	-1	-14,6118	-0,812
сумбурно	0,000152	0,0000104	14,5855	-1	-14,5855	-0,810
волнующий	0,000151	0,0000104	14,5592	1	14,5592	0,809
убогий	0,000151	0,0000104	14,5330	-1	-14,5330	-0,807
практиковать	6,24E-05	0,0000104	14,5067		0,0000	0,000
дружелюбный	0,000151	0,0000104	14,4804	1	14,4804	0,804
запасть	0,00015	0,0000104	14,4541	1	14,4541	0,803
безжалостный	0,00015	0,0000104	14,4279	-1	-14,4279	-0,801
вдохновить	0,00015	0,0000104	14,4016	1	14,4016	0,800
нырять	6,24E-05	0,0000104	14,3753		0,0000	0,000
взбесить	0,000149	0,0000104	14,3490	-1	-14,3490	-0,797
отменно	0,000149	0,0000104	14,3228	1	14,3228	0,796
меньшинство	6,24E-05	0,0000104	14,2965		0,0000	0,000
советовать	0,003841	0,000269146	14,2702	1	14,2702	0,793
ненависть	0,000148	0,0000104	14,2439	-1	-14,2439	-0,791

паршивый	0,000148	0,0000104	14,2177	-1	-14,2177	-0,790
камешек	6,24E-05	0,0000104	14,1914		0,0000	0,000
невероятный	0,000147	0,0000104	14,1651	1	14,1651	0,787
пугать	0,000147	0,0000104	14,1388	-1	-14,1388	-0,785
дурной	0,000147	0,0000104	14,1125	-1	-14,1125	-0,784
грустный	0,000146	0,0000104	14,0863	-1	-14,0863	-0,783
клан	6,24E-05	0,0000104	14,0600		0,0000	0,000
утомительный	0,00029	0,0000207	14,0337	-1	-14,0337	-0,780
ненастоящий	0,000146	0,0000104	14,0074	-1	-14,0074	-0,778
шоколад	6,24E-05	0,0000104	13,9812		0,0000	0,000
классно	0,000145	0,0000104	13,9549	1	13,9549	0,775
восторженный	0,000288	0,0000207	13,9286	1	13,9286	0,774
дельный	0,000145	0,0000104	13,9023	1	13,9023	0,772
устаревший	0,000144	0,0000104	13,8761	-1	-13,8761	-0,771
истинно	6,24E-05	0,0000104	13,8498		0,0000	0,000
грязь	0,000809	0,0000585	13,8235	-1	-13,8235	-0,768
минута	0,000339	0,0000585	13,7972		0,0000	0,000
бедный	0,000806	0,0000585	13,7710	-1	-13,7710	-0,765
жуткий	0,000804	0,0000585	13,7447	-1	-13,7447	-0,764
мести	0,000354	0,0000621	13,7053		0,0000	0,000
литературный	0,001659	0,000292295	13,6856		0,0000	0,000
грустно	0,001274	0,0000932	13,6659	1	13,6659	0,759
глупость	0,001595	0,000116918	13,6396	-1	-13,6396	-0,758
потрясти	0,000423	0,0000311	13,6133	1	13,6133	0,756
ассоциация	0,000166	0,0000311	17,7912		0,0000	0,000
духом	0,000166	0,0000311	17,7650		0,0000	0,000
радоваться	0,000421	0,0000311	13,5345	1	13,5345	0,752
лёгкий	0,000603	0,000113869	13,5082		0,0000	0,000
невыносимый	0,000789	0,0000585	13,4819	-1	-13,4819	-0,749
бумага	0,000302	0,0000585	13,4556		0,0000	0,000
успешный	0,000786	0,0000585	13,4294	1	13,4294	0,746
отвращение	0,000784	0,0000585	13,4031	-1	-13,4031	-0,745
пошло	0,000783	0,0000585	13,3768	-1	-13,3768	-0,743
масса	0,000603	0,000116918	13,3505		0,0000	0,000
издательство	0,000779	0,0000585	13,3243	-1	-13,3243	-0,740
любить	0,000778	0,0000585	13,2980	1	13,2980	0,739
интеллектуальный	0,000776	0,0000585	13,2717	1	13,2717	0,737
рассуждение	0,000302	0,0000585	13,2454		0,0000	0,000
неровный	0,000773	0,0000585	13,2192	-1	-13,2192	-0,734
эпохальный	0,000772	0,0000585	13,1929	1	13,1929	0,733
мастерский	0,00077	0,0000585	13,1666	1	13,1666	0,731
отстой	0,001536	0,000116918	13,1403	-1	-13,1403	-0,730
рекомендовать	0,004751	0,000362311	13,1140	1	13,1140	0,728
пена	0,000104	0,0000207	13,0746		0,0000	0,000
спасибо	0,000312	0,0000621	13,0549		0,0000	0,000
примитивно	0,00027	0,0000207	13,0352	-1	-13,0352	-0,724
капля	0,000104	0,0000207	13,0089		0,0000	0,000
бесподобный	0,000269	0,0000207	12,9827	1	12,9827	0,721
недоработанный	0,000268	0,0000207	12,9564	-1	-12,9564	-0,720

возразить	0,000104	0,0000207	12,9301		0,0000	0,000
ужасающий	0,000267	0,0000207	12,9038	-1	-12,9038	-0,717
кольцо	0,000208	0,0000414	12,8644		0,0000	0,000
по-другому	0,000208	0,0000414	12,8447		0,0000	0,000
ругаться	0,000265	0,0000207	12,8250	-1	-12,8250	-0,712
класс	0,000265	0,0000207	12,7987	1	12,7987	0,711
увлечься	0,000264	0,0000207	12,7725	1	12,7725	0,710
сродни	0,000104	0,0000207	12,7462		0,0000	0,000
шикарный	0,000263	0,0000207	12,7199	1	12,7199	0,707
обыденно	0,001708	0,000134573	12,6936	-1	-12,6936	-0,705
надежный	0,001181	0,0000932	12,6674	1	12,6674	0,704
светлый	0,000458	0,0000932	12,6411		0,0000	0,000
жалко	0,001475	0,000116918	12,6148	-1	-12,6148	-0,701
ужасный	0,001472	0,000116918	12,5885	-1	-12,5885	-0,699
неповторимый	0,000651	0,0000518	12,5622	1	12,5622	0,698
рекомендовать	0,000649	0,0000518	12,5360	1	12,5360	0,696
даваться	0,000146	0,0000311	12,5097		0,0000	0,000
разочарование	0,000388	0,0000311	12,4834	-1	-12,4834	-0,693
вкусный	0,000387	0,0000311	12,4571	1	12,4571	0,692
выгодно	0,000387	0,0000311	12,4309	1	12,4309	0,691
продумать	0,00127	0,000103518	12,2709		0,0000	0,000
уникальный	0,000497	0,0000414	12,0062	1	12,0062	0,667
классный	0,000497	0,0000414	12,0062	1	12,0062	0,667
ценить	0,000497	0,0000414	12,0062	1	12,0062	0,667
сопереживать	0,000497	0,0000414	12,0062		0,0000	0,000
волшебный	0,000701	0,0000585	11,9753		0,0000	0,000
мрачный	0,000701	0,0000585	11,9753	-1	-11,9753	-0,665
безуспешный	0,000701	0,0000585	11,9753	-1	-11,9753	-0,665
естественно	0,000701	0,0000585	11,9753		0,0000	0,000
бредятина	0,000701	0,0000585	11,9753	-1	-11,9753	-0,665
солидный	0,000701	0,0000585	11,9753	1	11,9753	0,665
интерестный	0,000701	0,0000585	11,9753	1	11,9753	0,665
внимательно	0,000701	0,0000585	11,9753		0,0000	0,000
неприятно	0,000701	0,0000585	11,9753	-1	-11,9753	-0,665
неблагодарный	0,000701	0,0000585	11,9753	-1	-11,9753	-0,665
безобразный	0,000699	0,0000585	11,9527	-1	-11,9527	-0,664
объяснить	0,001398	0,000116918	11,9611		0,0000	0,000
входить	0,000699	0,0000585	11,9527		0,0000	0,000
досадный	0,000699	0,0000585	11,9527	-1	-11,9527	-0,664
жутко	0,000699	0,0000585	11,9527	-1	-11,9527	-0,664
благородный	0,000699	0,0000585	11,9527	1	11,9527	0,664
гнетущий	0,000699	0,0000585	11,9527	-1	-11,9527	-0,664
неверно	0,000699	0,0000585	11,9527	-1	-11,9527	-0,664
поэт	0,000699	0,0000585	11,9527		0,0000	0,000
урод	0,000699	0,0000585	11,9527	-1	-11,9527	-0,664
расстроиться	0,000606	0,0000518	11,7060	-1	-11,7060	-0,650
прекрасно	0,000717	0,0000621	11,5397	1	11,5397	0,641
необходимый	0,000717	0,0000621	11,5397		0,0000	0,000
хороший	0,000827	0,0000725	11,4051	1	11,4051	0,634

сага	0,000827	0,0000725	11,4051		0,0000	0,000
реально	0,000827	0,0000725	11,4051	1	11,4051	0,634
здорово	0,000937	0,0000828	11,3178	1	11,3178	0,629
увлекательный	0,002922	0,000258794	11,2893	1	11,2893	0,627
передать	0,001158	0,000103518	11,1827		0,0000	0,000
невыразительный	0,003896	0,000350754	11,1067	-1	-11,1067	-0,617
наглый	0,001299	0,000116918	11,1068	-1	-11,1068	-0,617
гадкий	0,001299	0,000116918	11,1068	-1	-11,1068	-0,617
откровенный	0,001299	0,000116918	11,1068		0,0000	0,000
потратить	0,001299	0,000116918	11,1068		0,0000	0,000
вульгарный	0,001378	0,000124221	11,0940	-1	-11,0940	-0,616
чудесный	0,001898	0,000175377	10,8220	1	10,8220	0,601
перо	0,009537	0,000890251	10,7122		0,0000	0,000
бездумный	0,00011	0,0000104	10,6000	-1	-10,6000	-0,589
несправедливость	0,00022	0,0000207	10,6512	-1	-10,6512	-0,592
посредственный	0,00011	0,0000104	10,6000	-1	-10,6000	-0,589
сомнительный	0,00011	0,0000104	10,6000	-1	-10,6000	-0,589
раздумье	0,00011	0,0000104	10,6000		0,0000	0,000
немыслимый	0,00011	0,0000104	10,6000	1	10,6000	0,589
улёт	0,00011	0,0000104	10,6000	1	10,6000	0,589
постановка	0,00022	0,0000207	10,6512		0,0000	0,000
проблематичный	0,00011	0,0000104	10,6000	-1	-10,6000	-0,589
качественый	0,00011	0,0000104	10,6000	1	10,6000	0,589
служащий	0,00011	0,0000104	10,6000		0,0000	0,000
пресный	0,00011	0,0000104	10,6000	-1	-10,6000	-0,589
безысходность	0,00011	0,0000104	10,6000	-1	-10,6000	-0,589
эффективный	0,00011	0,0000104	10,6000	1	10,6000	0,589
волей	0,000325	0,0000311	10,4343		0,0000	0,000
банальный	0,000216	0,0000207	10,4502	-1	-10,4502	-0,581
обходиться	0,000108	0,0000104	10,4000		0,0000	0,000
прикупить	0,000108	0,0000104	10,4000		0,0000	0,000
практичный	0,000108	0,0000104	10,4000	1	10,4000	0,578
одиннадцать	0,000107	0,0000104	10,3200		0,0000	0,000
гордость	0,000215	0,0000207	10,3699	1	10,3699	0,576
грязно	0,000107	0,0000104	10,3200	-1	-10,3200	-0,573
проработанный	0,000107	0,0000104	10,3200	1	10,3200	0,573
недостойный	0,000107	0,0000104	10,3200	-1	-10,3200	-0,573
любящий	0,000537	0,0000518	10,3607		0,0000	0,000
отталкивающий	0,000107	0,0000104	10,3200	-1	-10,3200	-0,573
бессвязный	0,000107	0,0000104	10,3200	-1	-10,3200	-0,573
пасхальный	0,000107	0,0000104	10,3200		0,0000	0,000
креативный	0,000107	0,0000104	10,3200	1	10,3200	0,573
скука	0,000107	0,0000104	10,3200	-1	-10,3200	-0,573
впечатлять	0,000107	0,0000104	10,2600	1	10,2600	0,570
приоритет	0,000107	0,0000104	10,2600		0,0000	0,000
нудность	0,000107	0,0000104	10,2600	-1	-10,2600	-0,570
расстроенный	0,000107	0,0000104	10,2600	-1	-10,2600	-0,570
меткий	0,000107	0,0000104	10,2600		0,0000	0,000
слеза	0,00096	0,0000932	10,3049		0,0000	0,000

несовершенный	0,000107	0,0000104	10,2600	-1	-10,2600	-0,570
напрасный	0,000212	0,0000207	10,2493	-1	-10,2493	-0,569
повсюду	0,000106	0,0000104	10,1800		0,0000	0,000
мысленно	0,000106	0,0000104	10,1800		0,0000	0,000
бесхитростный	0,000424	0,0000414	10,2300	-1	-10,2300	-0,568
положительный	0,000106	0,0000104	10,1800	1	10,1800	0,566
соврать	0,000106	0,0000104	10,1800		0,0000	0,000
живописный	0,000106	0,0000104	10,1800	1	10,1800	0,566
оптимистичный	0,000106	0,0000104	10,1800	1	10,1800	0,566
безвкусный	0,000106	0,0000104	10,1800	-1	-10,1800	-0,566
нестабильный	0,000106	0,0000104	10,1800	-1	-10,1800	-0,566
обсудить	0,000105	0,0000104	10,0800		0,0000	0,000
медленно	0,000629	0,0000621	10,1295		0,0000	0,000
превосходный	0,00021	0,0000207	10,1287	1	10,1287	0,563
близость	0,00021	0,0000207	10,1287		0,0000	0,000
славный	0,000105	0,0000104	10,0800	1	10,0800	0,560
дискомфорт	0,000105	0,0000104	10,0800	-1	-10,0800	-0,560
занудно	0,000105	0,0000104	10,0800	-1	-10,0800	-0,560
максимализм	0,000105	0,0000104	10,0800	1	10,0800	0,560
изолировать	0,000105	0,0000104	10,0800		0,0000	0,000
резко	0,000105	0,0000104	10,0800	-1	-10,0800	-0,560
твёрдо	0,000105	0,0000104	10,0800		0,0000	0,000
удовлетворительный	0,000105	0,0000104	10,0800	1	10,0800	0,560
многофункциональный	0,000105	0,0000104	10,0800	1	10,0800	0,560
остроумный	0,000105	0,0000104	10,0800	1	10,0800	0,560
самец	0,000105	0,0000104	10,0800		0,0000	0,000
милый	0,000105	0,0000104	10,0800	1	10,0800	0,560
небезызвестный	0,000105	0,0000104	10,0800	1	10,0800	0,560
жизнерадостный	0,000105	0,0000104	10,0800	1	10,0800	0,560
поражать	0,000105	0,0000104	10,0600	1	10,0600	0,559
епископ	0,000105	0,0000104	10,0600		0,0000	0,000
восторженность	0,000105	0,0000104	10,0600	1	10,0600	0,559
жуть	0,000105	0,0000104	10,0600	-1	-10,0600	-0,559
слепой	0,000105	0,0000104	10,0600		0,0000	0,000
толково	0,000105	0,0000104	10,0600	1	10,0600	0,559
величайший	0,000105	0,0000104	10,0600	1	10,0600	0,559
смешанный	0,000105	0,0000104	10,0600		0,0000	0,000
галлюцинация	0,000105	0,0000104	10,0600		0,0000	0,000
безыскусный	0,000105	0,0000104	10,0600	-1	-10,0600	-0,559
скудный	0,000208	0,0000207	10,0483	-1	-10,0483	-0,558
навязчивый	0,000104	0,0000104	10,0000	-1	-10,0000	-0,556
созерцание	0,000104	0,0000104	10,0000		0,0000	0,000
познание	0,000416	0,0000414	10,0492		0,0000	0,000
примитивный	0,000208	0,0000207	10,0483	-1	-10,0483	-0,558
ковать	0,000104	0,0000104	10,0000		0,0000	0,000
защитить	0,000104	0,0000104	10,0000		0,0000	0,000
достойный	0,000104	0,0000104	10,0000	1	10,0000	0,556
пропитаться	0,000104	0,0000104	9,9600		0,0000	0,000

комфортный	0,000207	0,0000207	10,0081	1	10,0081	0,556
дуэт	0,000104	0,0000104	9,9600		0,0000	0,000
восхищаться	0,000104	0,0000104	9,9600	1	9,9600	0,553
агрессивность	0,000104	0,0000104	9,9600		0,0000	0,000
странно	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
жопа	0,000104	0,0000104	9,9600		0,0000	0,000
могущество	0,000104	0,0000104	9,9600		0,0000	0,000
безнадежный	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
неправильный	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
унылый	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
раздолбать	0,000104	0,0000104	9,9600		0,0000	0,000
пришел	0,000104	0,0000104	9,9600		0,0000	0,000
миллиард	0,000207	0,0000207	10,0081		0,0000	0,000
очаровательный	0,000104	0,0000104	9,9600	1	9,9600	0,553
кале	0,000104	0,0000104	9,9600		0,0000	0,000
совершенный	0,000104	0,0000104	9,9600	1	9,9600	0,553
поединок	0,000104	0,0000104	9,9600		0,0000	0,000
подвергать	0,000104	0,0000104	9,9600		0,0000	0,000
несомненный	0,000104	0,0000104	9,9600	1	9,9600	0,553
гадость	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
противный	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
поискать	0,000104	0,0000104	9,9600		0,0000	0,000
приличный	0,000104	0,0000104	9,9600	1	9,9600	0,553
жёстко	0,000207	0,0000207	10,0081		0,0000	0,000
переработка	0,000104	0,0000104	9,9600		0,0000	0,000
понравиться	0,000104	0,0000104	9,9600	1	9,9600	0,553
депрессивный	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
отходить	0,000104	0,0000104	9,9600		0,0000	0,000
разочаровать	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
проникновенный	0,000104	0,0000104	9,9600		0,0000	0,000
подпортить	0,000104	0,0000104	9,9600		0,0000	0,000
пустой	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
нашуметь	0,000207	0,0000207	10,0081		0,0000	0,000
смешной	0,000104	0,0000104	9,9600	1	9,9600	0,553
непродуманный	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
изысканный	0,000104	0,0000104	9,9600	1	9,9600	0,553
культовый	0,000104	0,0000104	9,9600	1	9,9600	0,553
максимально	0,000207	0,0000207	10,0081		0,0000	0,000
малолетний	0,000104	0,0000104	9,9600		0,0000	0,000
недостаточно	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
злобный	0,000104	0,0000104	9,9600	-1	-9,9600	-0,553
убедительный	0,000103	0,0000104	9,9200	1	9,9200	0,551
крепко	0,000103	0,0000104	9,9200		0,0000	0,000
слабоватый	0,000103	0,0000104	9,9200	-1	-9,9200	-0,551
шумный	0,000103	0,0000104	9,9200	-1	-9,9200	-0,551
этика	0,000103	0,0000104	9,9200		0,0000	0,000
изобретательный	0,000103	0,0000104	9,9200		0,0000	0,000
всего-то	0,000103	0,0000104	9,9200		0,0000	0,000
профессиональный	0,000103	0,0000104	9,9200	1	9,9200	0,551

трусость	0,000103	0,0000104	9,9200	-1	-9,9200	-0,551
незаурядный	0,000206	0,0000207	9,9679	1	9,9679	0,554
мудро	0,000103	0,0000104	9,9200	1	9,9200	0,551
скучный	0,000103	0,0000104	9,9200	-1	-9,9200	-0,551
температура	0,000103	0,0000104	9,8800		0,0000	0,000
уныло	0,000206	0,0000207	9,9277	-1	-9,9277	-0,551
недостаток	0,000103	0,0000104	9,8800	-1	-9,8800	-0,549
пожелание	0,000103	0,0000104	9,8800		0,0000	0,000
смеяться	0,000103	0,0000104	9,8800	1	9,8800	0,549
типичный	0,000103	0,0000104	9,8800	-1	-9,8800	-0,549
неприглядный	0,000103	0,0000104	9,8800	-1	-9,8800	-0,549
поплакать	0,000103	0,0000104	9,8800		0,0000	0,000
завлекать	0,000103	0,0000104	9,8800	1	9,8800	0,549
избитый	0,000103	0,0000104	9,8800	-1	-9,8800	-0,549
процветать	0,000103	0,0000104	9,8800		0,0000	0,000
проникать	0,000103	0,0000104	9,8800		0,0000	0,000
рассмотрение	0,000206	0,0000207	9,9277		0,0000	0,000
виртуозно	0,000103	0,0000104	9,8800	1	9,8800	0,549
испоганить	0,000103	0,0000104	9,8800	-1	-9,8800	-0,549
попутно	0,000103	0,0000104	9,8800		0,0000	0,000
стаж	0,000103	0,0000104	9,8800		0,0000	0,000
блистать	0,000103	0,0000104	9,8800	1	9,8800	0,549
провальный	0,000103	0,0000104	9,8800	-1	-9,8800	-0,549
поучительный	0,000103	0,0000104	9,8800	1	9,8800	0,549
уцелеть	0,000103	0,0000104	9,8800		0,0000	0,000
вожделение	0,000103	0,0000104	9,8800		0,0000	0,000
стыдный	0,000103	0,0000104	9,8800	-1	-9,8800	-0,549
легендарный	0,000103	0,0000104	9,8800	1	9,8800	0,549
сюра	0,000103	0,0000104	9,8800		0,0000	0,000
лаконичный	0,000103	0,0000104	9,8800	1	9,8800	0,549
второстепенный	0,000103	0,0000104	9,8800	-1	-9,8800	-0,549
двойственный	0,000103	0,0000104	9,8800		0,0000	0,000
вериться	0,000514	0,0000518	9,9190		0,0000	0,000
расстроить	0,000103	0,0000104	9,8800	-1	-9,8800	-0,549
разовый	0,000103	0,0000104	9,8800		0,0000	0,000
поражаться	0,000103	0,0000104	9,8800	1	9,8800	0,549
мыслитель	0,000103	0,0000104	9,8800		0,0000	0,000
пошлый	0,000103	0,0000104	9,8800	-1	-9,8800	-0,549
справка	0,000102	0,0000104	9,8000		0,0000	0,000
аромат	0,000102	0,0000104	9,8000		0,0000	0,000
тусклый	0,000102	0,0000104	9,8000	-1	-9,8000	-0,544
манный	0,000102	0,0000104	9,8000		0,0000	0,000
сказочно	0,000102	0,0000104	9,8000	1	9,8000	0,544
экранизировать	0,000102	0,0000104	9,8000		0,0000	0,000
бравый	0,000102	0,0000104	9,8000	1	9,8000	0,544
фальшивый	0,000404	0,0000414	9,7678	-1	-9,7678	-0,543
недоработка	0,000101	0,0000104	9,7400	-1	-9,7400	-0,541
уютный	0,000101	0,0000104	9,7200	1	9,7200	0,540
индивидуум	0,000101	0,0000104	9,7200		0,0000	0,000

наполнить	0,000607	0,0000621	9,7677		0,0000	0,000
неизгладимый	0,000101	0,0000104	9,7200	1	9,7200	0,540
списать	0,000202	0,0000207	9,7670		0,0000	0,000
оформить	0,000101	0,0000104	9,7200		0,0000	0,000
недописанный	0,000101	0,0000104	9,7200	-1	-9,7200	-0,540
полнокровный	0,000101	0,0000104	9,7200		0,0000	0,000
струна	0,000101	0,0000104	9,7200		0,0000	0,000
захватывать	0,000101	0,0000104	9,7200	1	9,7200	0,540
однообразный	0,000101	0,0000104	9,7200	-1	-9,7200	-0,540
оратор	0,000202	0,0000207	9,7670		0,0000	0,000
представительница	0,000101	0,0000104	9,7200		0,0000	0,000
добрый	0,000303	0,0000311	9,7520	1	9,7520	0,542
трепетный	0,000101	0,0000104	9,7200	1	9,7200	0,540
пребывание	0,0001	0,0000104	9,6400		0,0000	0,000
тешить	0,0001	0,0000104	9,6400		0,0000	0,000
молодец	0,000201	0,0000207	9,6866	1	9,6866	0,538
вялый	0,000101	0,0000104	9,6800	-1	-9,6800	-0,538
поднимать	0,000301	0,0000311	9,6718		0,0000	0,000
снести	0,0001	0,0000104	9,6400		0,0000	0,000
маяться	0,0001	0,0000104	9,6400		0,0000	0,000
обаятельный	0,000201	0,0000207	9,6866	1	9,6866	0,538
умница	9,98E-05	0,0000104	9,6000	1	9,6000	0,533
незаконченный	9,98E-05	0,0000104	9,6000	-1	-9,6000	-0,533
неизменный	9,98E-05	0,0000104	9,6000		0,0000	0,000
покойник	9,98E-05	0,0000104	9,6000		0,0000	0,000
приятно	0,001797	0,000186332	9,6455	1	9,6455	0,536
абсолют	9,98E-05	0,0000104	9,6000		0,0000	0,000
неуместный	9,98E-05	0,0000104	9,6000	-1	-9,6000	-0,533
потускнеть	9,98E-05	0,0000104	9,6000	-1	-9,6000	-0,533
гармоничный	9,98E-05	0,0000104	9,6000	1	9,6000	0,533
сбиться	9,98E-05	0,0000104	9,6000		0,0000	0,000
побояться	9,98E-05	0,0000104	9,6000		0,0000	0,000
нежелание	0,0002	0,0000207	9,6464	-1	-9,6464	-0,536
полноценный	9,98E-05	0,0000104	9,6000	1	9,6000	0,533
органиченный	9,98E-05	0,0000104	9,6000	-1	-9,6000	-0,533
размышлять	0,0003	0,0000311	9,6316		0,0000	0,000
послевкусие	0,000499	0,0000518	9,6379		0,0000	0,000
продвинутый	9,98E-05	0,0000104	9,6000	1	9,6000	0,533
медицина	0,0002	0,0000207	9,6464		0,0000	0,000
грозный	9,98E-05	0,0000104	9,6000		0,0000	0,000
ухмылка	9,98E-05	0,0000104	9,6000		0,0000	0,000
слабоватый	0,000399	0,0000414	9,6472	-1	-9,6472	-0,536
экспрессивный	9,98E-05	0,0000104	9,6000	1	9,6000	0,533
довесок	9,98E-05	0,0000104	9,6000		0,0000	0,000
восхищать	9,98E-05	0,0000104	9,6000	1	9,6000	0,533
тошнить	9,98E-05	0,0000104	9,6000	-1	-9,6000	-0,533
подсадить	9,98E-05	0,0000104	9,6000		0,0000	0,000
экстаз	9,98E-05	0,0000104	9,6000	1	9,6000	0,533
излагаться	9,98E-05	0,0000104	9,6000		0,0000	0,000

ориентир	9,98E-05	0,0000104	9,6000		0,0000	0,000
полезно	9,98E-05	0,0000104	9,6000	1	9,6000	0,533
лишний	9,98E-05	0,0000104	9,6000	-1	-9,6000	-0,533
нафталин	9,98E-05	0,0000104	9,6000		0,0000	0,000
цениться	9,98E-05	0,0000104	9,6000	1	9,6000	0,533
присоединяться	9,98E-05	0,0000104	9,6000		0,0000	0,000
занудный	9,98E-05	0,0000104	9,6000	-1	-9,6000	-0,533
мягко	9,98E-05	0,0000104	9,6000		0,0000	0,000
вроде	0,002804	0,000292295	9,5947		0,0000	0,000
удовольствие	0,004643	0,000496884	9,3441	1	9,3441	0,519
радостный	0,004144	0,000445125	9,3091	1	9,3091	0,517
попадаться	0,000543	0,0000585	9,2787		0,0000	0,000
бесполезный	0,000543	0,0000585	9,2787	-1	-9,2787	-0,515
надеяться	0,001086	0,000116918	9,2852		0,0000	0,000
кроме	0,001628	0,000175377	9,2852		0,0000	0,000
неловко	0,000543	0,0000585	9,2787	-1	-9,2787	-0,515
свобода	0,000543	0,0000585	9,2787		0,0000	0,000
безобразие	0,000543	0,0000585	9,2787	-1	-9,2787	-0,515
словарь	0,000543	0,0000585	9,2787		0,0000	0,000
высосать	0,000543	0,0000585	9,2787		0,0000	0,000
позиция	0,000543	0,0000585	9,2787		0,0000	0,000
болезнь	0,000543	0,0000585	9,2787	-1	-9,2787	-0,515
вредный	0,000543	0,0000585	9,2787	-1	-9,2787	-0,515