Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Национальный исследовательский университет
«Высшая школа экономики»

Факультет Бизнес-информатика

Отделение Программной инженерии

Кафедра Управление разработкой программного обеспечения

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

На тему: Программа анализа тональности текстов на основе методов

 машинного обучения

|  |
| --- |
| Студентка группы №472ПИ\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Вишневская Н.И. /«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2013 г. |
| Руководитель ВКРдоцент каф. УРПО, к.т.н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Ахметсафина Р.З. /«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2013 г. |

Москва, 2013

**Аннотация**

В данной работе рассматривается проблема автоматической классификации текстов по тональности, описываются методы машинного обучения для решения этой проблемы. Приводится описание алгоритмов классификации: наивного Байесовского классификатора и алгоритма k ближайших соседей. Рассматриваются способы векторного представления документов обучающей и тестовой выборок, а также функции весов. Для каждого сочетания параметров рассчитываются метрики эффективности. Для оценки работы алгоритмов используется метод перекретной проверки. По результатам проверок, выбирается сочетание векторной модели языка, функции весов и классификатора с наибольшей оценкой точности.

Содержание

[Введение 4](#_Toc357687869)

[1. Обзор предметной области 7](#_Toc357687870)

[1.1. Задача сентимент-анализа 7](#_Toc357687871)

[1.2. Подходы к определению тональности текста 7](#_Toc357687872)

[1.3. Обзор литературы 9](#_Toc357687873)

[1.4. Постановка задачи 10](#_Toc357687874)

[2. Разработка проекта решения 11](#_Toc357687875)

[2.1. Этапы решения 11](#_Toc357687876)

[2.2. Метрики эффективности 11](#_Toc357687877)

[2.3. Векторная модель языка 12](#_Toc357687878)

[2.4. Классификаторы 14](#_Toc357687879)

[2.4.1. Наивный Байесовский классификатор 14](#_Toc357687880)

[2.4.2. Метод k ближайших соседей 17](#_Toc357687881)

[2.5. Создание обучающей выборки 18](#_Toc357687882)

[2.6. Тестирование работы алгоритма 18](#_Toc357687883)

[3. Реализация и анализ результатов 19](#_Toc357687884)

[3.1. Язык программирования и среда разработки 19](#_Toc357687885)

[3.2. Структура проекта программы 19](#_Toc357687886)

[3.2.1. MakingTrainingSet 19](#_Toc357687887)

[3.2.2. SentAnalysis 19](#_Toc357687888)

[3.3. Точность и полнота классификации 22](#_Toc357687889)

[3.4. Дальнейшие направления исследований 23](#_Toc357687890)

[Заключение 24](#_Toc357687891)

[Список источников 25](#_Toc357687892)

# Введение

Люди всегда нуждались в советах и рекомендациях, когда стояли перед выбором: это мог быть выбор школы, вуза, книги или фильма и т.д. До распространения глобальной сети Интернет, они спрашивали совета у своих знакомых традиционными способами. С появлением веб 2.0 платформ, таких как блоги, форумы, социальные сети – процесс получения рекомендаций значительно упростился.

С одной стороны, глобальная сеть вместе с набором поисковых систем (таких как Google) сильно упрощает задачу поиска интересующей информации о продукте или услуге. С другой стороны, Horrigan сообщает [1], что 58% интернет-пользователей считают онлайн-поиск сложным и утомительным занятием. Дело в том, что объемы информации в интернете на сегодняшний день так велики, что релевантные данные попросту теряются в море информационного шума.

Таким образом, возникает необходимость в создании инструмента, который поможет потребителям в принятии правильных решений относительно приобретения тех или иных товаров или услуг. Проще говоря, нужна система, которая будет анализировать мнения членов интернет-сообщества касательно различных предметов обсуждения.

Исследования в этой области проводятся в настоящее время, и на данный момент не существует оптимального решения проблемы автоматического анализа мнений.

Сентимент-анализ (или анализ тональности) – это анализ текстовой информации с целью выявления эмоциональной оценки автора. Сентимент-анализ является одним из направлений компьютерной лингвистики, изучающих мнения и эмоции в текстовых документах (в рецензиях на что-либо). Тональностью называется эмоциональная оценка, выраженная в тексте рецензии по отношению к некоторому объекту.

К задачам сентимент-анализа относятся:

* классификация документов на основе мнений;
* классификация предложений на основе мнений;
* анализ мнений на основе характеристик объекта;
* создание словаря мнений;
* поиск сравнений;
* поиск спама в отзывах;
* анализ полезности отзывов;

Проблема анализа мнений интернет-пользователей становится все более важной как с теоретической, так и с прикладной точек зрения. Сентимент-анализ привлекает крупные компании в связи со значительным распространением социальных медиа, ведь маркетологам всегда было необходимо мониторить СМИ в поисках упоминаний своих брендов. Lillian Lee отмечает [2]:

«Consider, for instance, the following scenario. A major computer manufacturer, disappointed with unexpectedly low sales, finds itself confronted with the question: “Why aren’t consumers buying our laptop?” While concrete data such as the laptop’s weight or the price of a competitor’s model are obviously relevant, answering this question requires focusing more on people’s personal views of such objective characteristics. Moreover, subjective judgments regarding intangible qualities — e.g., “the design is tacky” or “customer service was condescending” — or even misperceptions — e.g., “updated device drivers aren’t available” when such device drivers do in fact exist — must be taken into account as well.»

Методы анализа тональности могут быть основой для системы, которая будет находить рецензии в сети (в блогах, социальных сетях, сайтах типа *kinopoisk.ru* или *market.yandex.ru*), а затем генерировать сокращенные версии мнения рецензентов, включающие в себя только тему рецензии и отношение рецензента. Это, теоретически, поможет аналитику сэкономить время на просмотр и анализ сотен, или даже тысяч схожих мнений. Такой подход может быть также использован для мониторинга ситуации на рынке в целом.

Еще одно применение методов сентимент-анализа – рекомендательные системы, дающие пользователю совет о выборе товара или услуги. Рекомендации, которые система будет давать, будут основаны на рецензиях реальных пользователей чего-либо.

Грамотный баннер-плейсмент – еще одно приложение для технологии сентимент-анализа. Это необходимо во избежание нежелательных сочетаний, таких как появление баннеров, рекламирующих кофейню, на сайте о вреде кофе.

Последнее, что стоит упомянуть касательно применения методов анализа тональности – государственная разведка, инструменты для выявления настроений народных масс. Это может быть полезно для изучения того, что думают избиратели о своих потенциальных политических лидерах или политиках в целом.

В настоящем исследовании будет рассмотрена задача сентимент-анализа мнений на уровне документа и классификация документов. В простейшем случае классификация текстов по тональности осуществляется на два класса (т.е. бинарная), обозначающие позитивные и негативные эмоциональные оценки. С увеличением количества классов, очевидно, уменьшается точность классификации.

Следует также указать проблемы, связанные с настоящим исследованием. Задача сентимент-анализа возникла сравнительно недавно и оптимального решения проблемы на данный момент не существует. Следует принять во внимание так же и то, что все задачи, так или иначе связанные с обработкой естественных языков – сложны и неоднозначны. Это в первую очередь касается методов машинного перевода, распознавания речи, а также сентимент-анализа. По этой причине, исследований в этой области не очень много, а русскоязычных работ практически нет.

# 1. Обзор предметной области

## Задача сентимент-анализа

Дан текст рецензии, предположительно посвященный единственной теме, с выраженной эмоциональной оценкой автора. Задача – определить данную рецензию с оценкой рецензента как принадлежащую одному из нескольких заранее определенных классов тональности. Тональностью в настоящем исследовании называется позитивное или же негативное отношение рецензента, так как рассматривается бинарная классификация.

## Подходы к определению тональности текста

Первый подход для решения проблемы сентимент-анализа основан на правилах. Этот подход базируется на лингвистике; в таком подходе большую роль играет семантика слов и правила построения предложений. Такой подход подразумевает наличие тонального словаря, содержащего слова или коллокации. Для каждого слова в тональном словаре отмечена тональность и, иногда, сила тональности (например, по шкале от одного до десяти, где «десять» - это сильная позитивная тональность). Тональный словарь может быть взят извне, или же сформирован статистически. Далее, согласно подходу, основанному на правилах, из каждого предложения или его части в изначальном тексте рецензии формируется синтаксическое дерево, содержащее цепочки из слов или коллокаций, зависящих друг от друга. Происходит определение объекта анализа и направления тональности. Лингвистичекие правила, по которым формируются синтаксические деревья, могут быть сформированы согласно предпочтениям исследователя. Далее, определенные слова или их комбинации сравниваются с оными из тональных словарей и, таким образом, отдельным предложениям присваивается направление тональности и, опционально, сила тональности. Тональность всего текста рецензии может формироваться на основе тональности его частей.

Второй подход основан на методах машинного обучения. В рамках этого подхода к решению проблемы сентимент-анализа, необходим набор текстов в качестве обучающей выборки. Машинное обучение с учителем нуждается в наборе заранее размеченных тектов рецензий. Отдельный экземпляр этого набора – пара из вектора признаков, который является представлением конкретного текста, и тональности этого текста. Тональность текста рецензии выявляется либо самим автором рецензии, либо экспертом (например, исследователем).

Под термином «вектор признаков» подразумеваетсявекторная модель семантики. Вектор признаков – это представление каждого текста рецензии как точки в многомерном пространстве. Такое представление текстов рецензий в виде векторов признаков необходимо для дальнейшей классификации, так как векторы можно сравнивать друг с другом, путем вычисления расстояний между ними. Близко лежащие друг к другу векторы соответствуют семантически схожим документам (текстам). Более формально, *dj* = (*w1j, w2j, …, wnj*)*,* где *dj* – векторное представление документа *j, wij* – вес термина *i* в документе *j* и *n* – число документов в выборке. Понятие «термина» может варьироваться – термин может отображать отдельное или ключевое слово, пару слов или даже фразу из текста. Каждый термин относится к отдельному измерению*,* количество параметров вектора равно n.Вес конкретного термина указывает на важность этого термина в документе, то есть если термин появляется в тексте, то он имеет ненулевой вес. Вес может быть вычислен различными способами, это будет обсуждено далее.

В алгоритмах, основанных на машинном обучении с учителем, происходит анализ размеченного набора текстов (обучающей выборки) и статистически формируется паттерн для использования при классификации новых входных векторов. Машинное обучение без учителя ставит проблему выявления паттерна на основании неразмеченной выборки. Из-за того, что тексты в выборке неразмечены, не существует средств оценки правильности работы алгоритма (то есть правильности решений, принятых классификатором). Поэтому вероятность ошибки намного больше, чем в методах типа «обучение с учителем», и эффективность таких алгоритмов значительно ниже.

Многие исследователи [3, 4] используют комбинацию двух подходов: лингвистического и статистического. Причина в том, что смешанный подход на практике показывает лучшие результаты.

Линвистический подход дает достаточно хорошие результаты, так как позволяет выявлять направление тональности для частей текста рецензии, а не только для текста в целом. Подход, основанный на правилах, тесно связан с семантикой слов, и поэтому демонстрирует более глубокий анализ текста. Но у этого подхода есть и серьезные недостатки. Например, линвистический подход сильно привязан к языку, на котором написаны тексты. Модель, разработанная для одного языка, совершенно неприменима для другого, в первую очередь из-за тональных словарей, сформированных для конкретного языка, а также из-за синтаксиса языка.

## Обзор литературы

Несмотря на трудности, связанные с проблемой анализа тональности, существует некоторое количество научных работ, посвященных этой неоднозначной теме. Достаточно часто появляются новые статьи и публикации конференций. Область лингвистики, связанная с обработкой естественных языков, развивается быстро вместе с технологическим прогрессом и увеличением спроса на технологию сентимент-анализа.

Котельников Е.В. и Клековкина М.В. в своем исследовании [5] представили методы автоматической обработки текстов на основе методов машинного обучения. В ходе своего исследования они пытались отыскать оптимальный вариант векторной модели представления текстов и наилучшего классификатора. Исследователи проанализировали несколько векторных моделей на основе подхода TF.IDF (TF — Term Frequency, IDF — Inverse Document Frequency). Были рассмотрены несколько методов машинного обучения, в том числе наивный Байесовский классификатор, метод опорных векторов (Support Vector Machine или SVM), метод ключевых слов и т.д. Экспериментальным путем было выявлено, что наилучшие результаты дает бинарная модель с косинусной нормализацией без обучения и метод, комбинирующий использование ключевых слов и SVM.

В исследовании Tang С. и соавт. [6] предлагаются два метода типа «обучение с учителем» (class association rules и наивный Байесовский классификатор) для выявления характеристик продукта или услуги и классификации этих характеристик для составления краткого резюме отзывов потребителей. Результаты их работы показывают, что каждый метод хорош по-своему, это зависит от конкретной метрики эффективности. Работа этих методов обычно достигает более 70% точности.

Как было указано выше, исследователи часто комбинируют подходы для достижения наилучших результатов. Например, научная работа Васильева В.Г., Давыдова С. и Худяковой М.В. [7] использует лингвистический подход, дополенный методами машинного обучения для коррекции отдельных правил классификации путем обучения.

Большинство исследователей в своих работах полагаются на лингвистический подход, и этому есть разумное объяснение. Алгоритмы, основанные на правилах, дают более точные результаты, так как работа этих методов тесно связана с семантикой слов, в отличие от методов машинного обучения, оперирующих со статистикой и теорией веороятности. Но, как уже было упомянуто, лингвистический подход обладает рядом серьезных недостатков. Согласно Tang и соавт. [6]:

«Most existing techniques rely on natural language processing tools to parse and analyze sentences in a review, yet they offer poor accuracy, because the writing in online reviews tends to be less formal than writing in news or journal articles. Many opinion sentences contain grammatical errors and unknown terms that do not exist in dictionaries.»

Лингвистический подход может предоставить относительно точные результаты, будучи реализованным для научных или журнальных статей или других, грамматически верных текстов. Принимая во внимание тот факт, что одно из главных применений анализа тональности – бизнес-разведка, становится понятно, что интернет-сообщество не может обеспечить исследователей грамматически правильными текстами, или даже текстами без орфографических ошибок. В этой связи, не стоит и говорить о грамматике и стиле письма рядовых пользователей социальных сетей.

Кроме того, подход, основанный на правилах, сильно привязан к конкретному языку. В связи с вышесказанным, в настоящем исследовании рассматривается подход, основанный на методах машинного обучения с учителем.

## Постановка задачи

Целью настоящего дипломного проекта является реализация программы для сентимент-анализа рецензий на основе методов машинного обучения с учителем. Для достижения этой цели необходимо решить следующие задачи:

1. Исследовать несколько векторных моделей представления текста (униграммы, биграммы) и фунций весов (бинарная, частотная). Выбрать то сочетание модели языка и весов, которое даст наилучший показатель точности на данной выборке.
2. Исследовать и реализовать несколько классифицирующих алгоритмов (в т.ч.наивный Байесовский алгоритми алгоритмk ближайших соседей). Выбрать алгоритм, дающий наибольший уровень точности на конкретной обучающей выборке.

# 2. Разработка проекта решения

## Этапы решения

Процесс разработки программы сентимент-анализа в рамках настоящего исследования, схож с процессом разработки любой другой программы на основе методов машинного обучения. Методы машинного обучения с учителем требуют наличия обучающей выборки. После того, как обучающая выборка сформирована, каждый экземпляр из выборки выражается как вектор признаков. Далее, определяется тональность (позитивное или негативное отношение автора) для каждого элемента выборки – таким образом, классификатор будет обучаться в дальнейшем. После формирования векторов, реализуется какой-либо алгоритм классификации. Полученный классификатор далее обучается на основе изначальной выборки и становится пригодным для осуществления классификации новых текстов.

Результатом действий, описанных выше, будет программа анализа тональности текстов на основе методов машинного обучения.

Главной задачей, поставленной в настоящей работе, является реализация алгоритма определения тональной направленности текста на основе методов машинного обучения.

Решение этой задачи можно разбить на этапы:

1. Выбор метрик для оценки эффективности алгоритмов.
2. Выбор признаков, по которым будет осуществляться классификация.
3. Выбор и реализация нескольких алгоритмов классификации.
4. Создание обучающей выборки.
5. Проверка эффективности алгоритмов.

## Метрики эффективности

В качестве метрик правильности классификации текстов были выбраны точность (precision) и полнота (recall). Точность в пределах класса – это доля текстов, действительно принадлежащих данному классу, относительно всех текстов, причисленных классификатором к этому классу. Полнота системы – отношение числа найденных классификатором текстов, принадлежащих классу, к числу всех текстов этого класса в тестовой коллекции.

В результате классификации текстов рецензий тестовой выборки, к классу позитивных (положительных) рецензий правильно отнесены TP текстов, неправильно – FP, к классу негативных (отрицательных) рецензий правильно были отнесены TN текстов, неправильно – FN. Иными словами:

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

В таблице ниже наглядно представлены результаты классификации.

Таблица 1. Результаты классификации:

|  |  |
| --- | --- |
| Документ i | Экспертная оценка |
| Положительная | Отрицательная |
| Оценка системы | Положительная | TP | FP |
| Отрицательная | FN | TN |

Тогда, относительно класса позитивных рецензий, точность *Precision* и полнота *Recall* определяются следующим образом:



## Векторная модель языка

Качество будущего классификатора зависит от векторной модели (модели представления текстов). Иными словами, выбранный набор признаков вектора имеет огромное влияние на точность будущего классификатора.

Одна из наиболее часто используемых моделей – модель n-грамм. В рамках модели, векторы признаков являют собой последовательности из n слов. n может быть любым целым числом больше нуля; если n равно единице, то такая модель называется «униграмм», если n равно двум – «биграмм», трем – «триграмм», и так далее.В простейшем случае модели униграмм, вектор признаков представляет собой набор всех слов их текста, – то есть каждое слово представляет собой термин.В случае биграмм термин образует пара слов из текста. Это означает, что пара слов рассматривает как признак вектора. Для примера рассмотрим предложение ‘*Jane prefers coffee to tea’.* Оно может быть представлено как вектор униграмм: *[Jane; prefers; coffee; to; tea]*. Кроме этого, оно может быть выражено как набор биграмм: *[[Jane prefers]; [prefers coffee]; [coffee to]; [to tea]].* Также может рассматриватьсякомбинация униграмм и биграмм отдельно: *[Jane; prefers; coffee; to; tea; [Jane prefers]; [prefers coffee]; [coffee to]; [to tea]].*

Символьные n-граммы должны быть рассмотрены отдельно. Символьная n-грамма означает n идущих подряд символов из текста. Например, символьные биграммы слова ‘word’ будут выглядеть следующим образом: ‘\_w’, ‘wo’, ‘or’, ‘rd’ and ‘’d\_. Символьные триграммы того же слова будут: ‘\_wo’, ‘wor’, ‘ord’ and ‘rd\_’. Это может казаться примитивным способом, однако, подобная векторная модель языка показывает неплохие результаты в некоторых работах [8, 9].

Первым делом, необходимо представить коллекцию документов в виде матрицы термин-документ. Строки будут обозначать отдельные документы (тексты), а колонки – словарь выборки, заключающий в себе все слова в коллекции документов.

Проиллюстрируем примером: даны два коротких документа; они являются примером простейшей обучающей выборки.

1. ‘*Jane likes coffee and tea’*
2. *‘Jane also likes cookies’*

На данном этапе может быть сформирован словарь выборки; он будет содержать все слова из текстов обучающей выборки. Далее, можно будет сформировать матрицу термин-документ: значение в каждой ячейке показывает, встречается ли определенный термин в опредленном документе (1) или нет (0).

Таблица 2: матрица термин-документ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *Jane* | *likes* | *coffee* | *and* | *tea* | *also* | *cookies* |
| 1ый текст | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 2ой текст | 1 | 1 | 0 | 0 | 0 | 1 | 1 |

Таким образом, получаются бинарные вектора: *d1* = [1,1,1,1,1,0,0] и *d2* = [1,1,0,0,0,1,1].

С помощью модели n-грамм, можно получить набор векторов, которые в дальнейшем могут быть сравнены друг с другом, путем вычислений расстояний между ними. Чем ближе векторы друг к другу – тем более они походят друг на друга.

Под расстоянием между векторами может пониматься, к примеру, Евклидово расстояние:



или максимум модулей:



или другая функция расстояния.

Вектор признаков – это алгебраическая модель представления текстов:

*dj* = (*w1j, w2j, …, wnj*),

где *dj* - векторное представление документа *j, wij* – вес термина *i* в документе *j* и *n* – количество всех терминов в выборке. Для полноценного понимая принципа векторной модели текста, необходимо объяснить, как именно рассчитываются веса векторов. Существует несколько базовых функций весов. Pang и соавт. в своем исследовании [10] выявили, что бинарная функция взвешивания векторов более эффективна. Это значит, что наличие термина в документе важнее, чем его частота. Бинарные векторы представлены как последовательность нулей и единиц: если конкретный термин из словаря выборки встречается в тексте – вес термина будет равен 1, иначе – 0. Частотные векторы формируются на основе количества вхождений определенного терминав классе документов.

## Классификаторы

### Наивный Байесовский классификатор

Наивный Байесовский классификатор (Naïve Bayes Classifier, NBС) является одним из примеров использованияметодов векторного анализа. Данная модель классификации базируется на понятии условной вероятности принадлежности документа *d*классу *c*.

NBС – один из самых часто используемых классификаторов, из-за сравнительной простоты в имплементации и тестировании. В то же время, наивный Байесовский классификатор демонстрирует не худшие результаты, по сравнению с другими, более сложными классификаторами.

В основе наивного Байесовского классификатора лежит теорема (или формула) Байеса:



Для данной модели, документ – это вектор: *d =*{ *w1* *, w2* *,…, wn* }, где *wi* - вес *i*-ого термина, а *n* – размер словаря выборки. Таким образом, согласно теореме Байеса, вероятность класса *с* для документа *d* будет:

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

Таким образом, вычисляется условная вероятность для всех классов.

Наиболее вероятный класс *с\**, которому принадлежит документ *d* тот, при котором условная вероятность принадлежности документа *d* классу с максимальна:

$$с^{\*}=arg max\_{c} P \left(c \right| d).$$

По теореме Байеса:

$$с^{\*}=arg max\_{c} P \left(d \right| c)\*P (c)$$

и, так как

*d =*{ *w1* *, w2* *,…, wn* },

то

$$с^{\*}=arg max\_{c} P \left(w\_{1} ,w\_{2} ,…, w\_{n}  \right| c)\*P \left(c\right).$$

Знаменатель может быть опущен, так как для одного и того же документа *d* вероятность *P(d)* будет одинаковой, а это значит, что ее можно не учитывать.

Для наивного Байесовского классификатора определено существенное допущение – предполагается, что все признаки *x1 , x2 ,…, xn* документа *d* независимы друг от друга. Из-за этого допущения модель и получила название «наивная». Это очень серьезное упрощающее допущение и, в общем случае, оно неверно, но наивная Байесовская модель демонстрирует неплохие результаты, несмотря на это [10, 11]. Предполагается так же, что позиция термина в предложении не важна. Как следствие, условную вероятность

*P* (*w1* *, w2* *,…, wn*  | *c*)

для признаков *x1* *, x2* *,…, xn* , можно представить как

*P* (*w1* | *c*) \* (*w2* | *c*) \*...\* ( *wn* | *c*) = Пi *P*(*w*i | *cj*)

Таким образом, для нахождения наиболее вероятного класса для документа

*d =* { *w1* *, w2* *,…, wn* }с помощью наивного Байесовского классификатора, необходимо посчитать условные вероятности принадлежности документа *d* для каждого из представленных классов отдельно и выбрать класс, имеющий максимальную вероятность:

$$C\_{NB}=argmax\_{c} [ P\left(c\_{j}\right)\*\prod\_{i}^{}P\left(w\_{i} \right|c\_{j }) ] $$

Теперь необходимо оценить *P(cj)* и*P(wi | cj).* Оценить вероятность класса несложно: *P(cj)* является отношением количества докуметов класса *j* в обучающей выборке к общему количеству документов в выборке.

**,

Где *Dc* - количество документов класса *с*, а *D* – общее количество документов в выборке.

Для оценки условных вероятностей для признаков, используется формула:

$\hat{P} \left(w\_{i }\right| c\_{j} )= \frac{count (w\_{i }, c\_{j} )}{\sum\_{w\in V}^{}count (w, c\_{j}) }$ ,

где *P*(*wi* | *cj*) определяется как отношение количества терминов *wi*в классе *cj*общему количеству терминов в этом классе. *V* – словарь обучающей выборки.

Существует небольшая проблема, связанная с этой формулой. Если в тестовом наборе встретится слово, которое не встречается в наборе обучающих документов, то вероятность *P*(*wi* | *cj*) этого слова для любого из классов будет равна нулю. Поскольку

*P*(*d* | *cj*) ≈ Пi *P*(*wi* | *cj*), то и вероятность принадлежности документа любому из классов также будет равна нулю, что, конечно, неправильно. Для решения этой проблемы обычно используют так называемое аддитивное сглаживание (add-1 smoothing или сглаживание Лапласа). Идея add-1 сглаживания заключается в том, что к частотам появления всех терминов из словаря искуственно добавляется единица. Получается, что термины, которые не присутствовали в документах обучающей выборки, получают незначительную, но не нулевую вероятность появления и, тем самым, дают возможность определить документ в какой-либо из классов.

$$\hat{P} \left(w\_{i }\right| c )= \frac{count \left(w\_{i }, c \right)+ 1}{\sum\_{w\in V}^{}\left(count \left(w, c\right)+ 1\right)}= \frac{count \left(w\_{i }, c \right)+ 1}{(\sum\_{w\in V}^{}\left(count \left(w, c\right)\right)+ |V|}$$

Здесь $|V|$- количество слов в словаре обучающей выборки.

Как было сказано выше, документ представляется в виде вектора для классификации: *d =* { *w1 , w2 ,…, wn* }, где *w1 , w2 ,…, wn* - веса для каждого из терминов словаря выборки. *wi* может быть количеством вхождений термина *xi*в документ *d*, или же может быть задано бинарно. Для бинарного вектора число вхождений термина *wi*не имеет значения, важен лишь факт появления *wi*в документе *d*.

Можно заметить, что для относительно больших текстов, вероятность *P*(*cj*| *d*) представляет собой произведение большого количества очень маленьких дробей. Для того чтобы избежать потери точности, можно заменить произведение вероятностей суммой логарифмов вероятностей.

###  Метод k ближайших соседей

Метод k ближайших соседей - k Nearest Neighbors или k NN.

Метод ближайших соседей – еще один алгоритм классификации текстов. Для его реализации нужна обучающая выборка размеченных рецензий. Для определения класса рецензии из тестовой выборки, нужно определить расстояние от вектора этой рецензии до векторов из обучающей выборки. Определить k объектов обучающей выборки, расстояние до которых минимально (k задается экспертом или выбирается согласно оценкам эффективности). Класс входного вектора – это класс, которому принадлежат больше половины из соседних k векторов. В качестве функции расстояния было выбрано Евклидово расстояние:



## Создание обучающей выборки

Для создания решений, основанных на методах машинного обучения, требуется размеченная обучающая выборка. От подбора выборки напрямую зависит качество классифицирующего алгоритма. Для максимизации значений оценок эффективности, обучающая выборка составляется из текстов той же предметой области, для которой будет применяться классификатор.

В качестве обучающей и тестовой выборки, были отобраны 800 положительных и 800 отрицательных рецензий на фильмы с сайта *kinopoisk.ru*. В качестве тональности рецензий брался выбранный автором тип рецензии (положительный или отрицательный).

## Тестирование работы алгоритма

Для тестирования точности алгоритмов классификации, был использован метод перекрестной проверки (скользящий контроль, кросс-валидация, cross-validation). В рамках этой процедуры фиксируется некоторое множество разбиений исходной выборки на две подгруппы: обучающую и контрольную. Для каждого разбиения выполняется настройка [алгоритма](http://www.machinelearning.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) по обучающей подгруппе, затем оценивается его эффективность на векторах контрольной подгруппы. Оценкой перекрестной проверки называется средние по всем разбиениям величины точности и полноты на контрольных подгруппах.

Если объекты выборки независимы, то средние значения оценок эффективности перекрестной проверки дадут несмещенные оценки эффективности.

# 3. Реализация и анализ результатов

##  3.1. Язык программирования и среда разработки

 В качестве языка разработки был выбран С#, в качестве среды разработки – Microsoft Visual Studio 2010.

##  3.2. Структура проекта программы

### 3.2.1. MakingTrainingSet

Для агрегирования рецензий была разработана вспомогательная программа **MakingTrainingSet**, способная собирать тексты рецензий с сайта *kinopoisk.ru* (вместе с их тональностью), и записывать их в пригодном для основной программы формате. Также, тексты могут быть взяты не с сайта *kinopoisk.ru*, а из других источников. В таком случае, исследователю необходимо вручную скопировать текст рецензии и вставить в соответвующее окно на форме программы, определив также тип тональности, и сохранить полученный документ в выбранный файл с обучающей выборкой.

Из текстов обучающей и тестовой выборок удалялись все знаки препинания и лишние пробелы. Весь текст переводился в нижний регистр. Перед текстами с отрицательной тональностью добавлялся знак «-».

### 3.2.2. SentAnalysis

Основная программа **SentAnalysis** позволяет исследователю загрузить обучающую выборку, выбрать алгоритм классификации, настроить параметры алгоритма для работы.

С помощью элементов графического интерфейса пользователя, можно выбрать алгоритм классификации (метод k ближайших соседей или наивный Байесовский классификатор). Для метода k ближайших соседей исследователем выбирается собственно k. По умолчанию k = 3, и если выбранное исследователем k нецелое или значение k больше половины количества текстов обучающей выборки, k становится равным своему значению по умолчанию.

После выбора обучающей выборки и классифицирующего алгоритма, выбирается векторная модель текста и веса векторов. Векторная модель текста представлена двумя вариантами: униграммы и биграммы. Функция веса также представлена в двух вариантах: бинарная и частотная.

Для любого из методов составляется словарь, включающий в себя все термины из всех текстов обучающей выборки. Для модели униграмм термин подразумевает 1 слово, для модели биграмм – пару рядом стоящих слов. При этом не происходит предварительного стемминга текстов (т.е. слова не приводятся к общему виду): слова «кошка» и «кошке» будут различными словами.

Для метода k ближайших соседей, из каждого текста обучающей выборки формируется вектор, размерностью которого является длина словаря, а элементами являются количества вхождений конкретного термина из словаря (для частотной функции веса) или индикатор присутствия термина в тексте (для бинарной функции веса). Входной текст также преобразуется к нужной векторной форме. Далее, с помощью одного из методов класса **SentAnalysis,** рассчитывается Евклидово расстояние от каждого из векторов обучающей выборки до входного вектора, выбираются k векторов обучающей выборки, имеющих наименьшее расстояние до входного вектора. Тональностью входного текста будет тональность, определенная для большей части ближайших k векторов обучающей выборки. Для более точной классификации следует выбирать нечетное значение k.

Для наивного Байесовского классификатора, для каждого термина из входного текста высчитывается вероятность его появления для каждого из двух классов (на основании теоремы Байеса об условной вероятности). На основании этого, рассчитывается вероятность принадлежности всего текста каждому классу по отдельности. Большее значение вероятности принадлежности текста классу определяет наиболее вероятный класс входного текста.

Кроме этого, **SentAnalysis** позволяет высчитывать значения оценок эффективности (точности и полноты) для специально размеченных тестовых выборок – для этого предусмотрен специальный функционал. Рецензии тестовой выборки, в данном случае, должны быть представлены в таком же виде, что и рецензии обучающей выборки, то есть иметь предварительную обработку и разметку тональности. Для определения оценок эффективности алгоритма с конкретными параметрами, выполняется алгоритмическая классификация текстов тестовой выборки и сравнение результатов классификации текстов с экспертной оценкой. Экспертной оценкой на проводимых тестах являлась тональная оценка автора конкретной рецензии.

**

Рисунок 1. Диаграмма классов

Класс **Document.cs** является классом для хранения документа для метода k ближайших соседей. Помимо самого документа, Document хранит его векторное представление и тональную направленность.

Класс **Termin.cs** необходим для наивного Байесовского классификатора. Termin хранит сам термин, а также вероятность его появления для положительного и отрицательного классов тональности.

Класс **NearNeigbours.cs** содержит методы для классификации текстов с помощью метода k ближайших соседей.

Класс **NaiveBayes.cs** содержит методы для выполнения классификации с помощью наивного Байесовского классификатора.

Класс-форма **Form1.cs** содержит элементы графического интерфейса пользователя и необходим для удобства подбора параметров классификации. Также содержит метод предварительной обработки входных текстов-рецензий.

##  3.3. Точность и полнота классификации

Для тестирования алгоритмов определение тональности текстов рецензий был использован метод кросс-валидации (или, по-другому, перекрестной проверки). Процедура кросс-валидации выполняется следующим образом:

1. Фиксируется множество разбиений обучающей выборки на собственно обучающую и тестовую.
2. Для каждого разбиения происходит обучение алгоритма на обучающей подвыборке и тестирование на тестовой.
3. Результатом кросс-валидации алгоритма являются средние значения оценок эффективности для тестовых подвыборок.

Для 800 положительных и 800 отрицательных кинорецензий было сделано 4 подразбиения (1200 рецензий – обучающая выборка, 400 рецензий – тестовая). В каждой обучающей и тестовой подвыборке содержалось одинакое количество положительных и отрицательных рецензий. Для каждой группы рассчитывались оценки эффективности, а затем высчитывались их средние значения. Таким образом, получились усредненные оценки эффективности. Результаты усредненных оценок эффективности для каждого из алгоритмов представлены ниже, в таблицах 3 и 4.

Таблица 3. Наивный Байесовский классификатор

|  |  |  |
| --- | --- | --- |
| **Векторная модель** | **Униграммы** | **Биграммы** |
| **Веса векторов** | **Точность, %** | **Полнота, %** | **Точность, %** | **Полнота, %** |
| **Бинарные векторы** | 84,5 | 88,25 | 85,3 | 91,5 |
| **Частотные векторы** | 85,2 | 87,4 | **86,2** | **89,4** |

 Наивный Байесовский классификатор показал очень хорошие результаты, наиболее эффективной конфигурацией в плане точности оказалось сочетание биграмм и частотной функции взвешивания – точность 86%. Хотя, на рассматриваемой выборке, для разных параметров классификации, оценка точности варьируется слабо – поэтому эффективным можно считать любое сочетание.

Таблица 4. Метод k ближайших соседей (для k = 3)

|  |  |  |
| --- | --- | --- |
| **Векторная модель** | **Униграммы** | **Биграммы** |
| **Веса векторов** | **Точность, %** | **Полнота, %** | **Точность, %** | **Полнота, %** |
| **Бинарные векторы** | **63,6** | **68,5** | 52,3 | 80 |
| **Частотные векторы** | 61,75 | 71,5 | 51 | 96,1 |

 Были получены довольно низкие результаты для метода k ближайших соседей. Лучший результат – 63% точности для сочетания униграмм и бинарной функции взвешивания. Это может быть объяснено тем, что для формирования векторов использовался весь словарь обучающей выборки, заключающий в себе примерно 30000 уникальных терминов для униграмм. Словарь терминов-биграмм гораздо больше, т.к. уникальных биграмм в выборке очень много. Вследствие этого, каждый конкретный вектор имел много измерений, был сильно разрежен. В этом случае, все вектора в n-мерном пространстве почти одинаково далеки друг от друга, что дает весьма неточный результат в поиске ближайших соседей. Предположительно, результат станет лучше после сокращения словаря, определения ключевых слов.

## 3.4. Дальнейшие направления исследований

Для повышения эффективности программы, возможно, следует дополнить существующее решение элементами лингвистики. Лингвистические подходы позволяют определять тональность частей текста, тем самым существенно увеличивая точность классификации для отдельного текста – ведь, как правило, в рецензиях упоминаются и положительные и отрицательные стороны объекта тональности.

Кроме этого, стоит исключать из составляетмого словаря слова, не имеющие эмоциональной окраски, т.е. незначащие. Вполне возможно, что лучшие результаты даст извлечение ключевых слов (признаков).

Далее, для уменьшения словаря, также стоит проводить стемминг (приведение слов к их стеммам - основам). Также, можно высчитывать важность слов: например, согласно нотации TF.IDF. Важными словами для конкретного текста будут считаться слова, часто встречающиеся в данном тексте и редко – в остальных текстах выборки. Малозначимые слова можно убирать из словаря, тем самым, опять же, уменьшая его.

# Заключение

В ходе данного исследования была реализована программа анализа тональности текстов на основе методов машинного обучения.

Были решены следующие задачи:

1. Изучена проблема анализа тональности, проанализированы подходы ее решения.
2. Релизовано два алгоритма классификации текстов по тональности (наивный Байесовский классификатори алгоритмk ближайших соседей).
3. Выбраны метрики и вычислены оценки эффективности алгоритмов.
4. Проведено тестирование эффективности алгоритмов методом кросс-валидации.
5. Для каждого алгоритма было выбрано сочетание векторной модели языка (биграммы, униграммы) и функции весов (бинарная, частотная), которое дало наилучший показатель точности на данной выборке.

Для повышения оценок эффективности алгоритмов, предположительно, следует дополнить их элементами лингвистического анализа. Исследования [3, 4, 7] показывают, что лучшие результаты достигаются путем комбинирования лингвистического и статистического подходов.

# Список источников

[1] Horrigan, J. A. Pew Internet & American Life Project Report *//* Online shopping*.* – 2008.

[2] Lee, L. “I’m sorry Dave, I’m afraid I can’t do that”: Linguistics, statistics, and natural language processing circa 2001 // Committee on the Fundamentals of Computer Science: Challenges, Computer Science Opportunities, and National Research Council Telecommunications Board, editors, Computer Science: Reflections on the Field, Reflections from the Field. Washington: The National Academies Press. – 2004. – С. 111–118.

[3] Karkaletsis, V., Klenner, M., Rentoumi, V., Petrakis, S., Vouros, G. United we stand: improving sentiment analysis by joining machine learning and rule based methods.// 7th International Conference on Language Resources and Evaluation (LREC 2010), Malta.

[4] Prabowo, R. & Thelwall, M. [Sentiment analysis: A combined approach.](http://citeseer.ist.psu.edu/viewdoc/download;jsessionid=21C7C04158501E322A1371EAEF97B405?doi=10.1.1.102.7125&rep=rep1&type=pdf) // Journal of Informetrics. – 2009. - № 3(2). – С. 143-157.

[5] Котельников Е.В., Клековкина М.В. Автоматический анализ тональности текстов на основе методов машинного обучения. РОМИП 2011.

[6] Tang, X. Yang, C., Wong, Y., Wei C. Understanding Online Consumer Review Opinions with Sentiment Analysis using Machine Learning // Pacific Asia Journal of the Association for Information Systems. – 2010. - № 3(2). – С. 73-89.

[7] Худякова М.В., Давыдов С., Васильев В.Г. Классификация отзывов пользователей с использованием фрагментных правил. РОМИП 2011.

[8] Aisopos, F., Papadakis, G. & Varvarigou, T. Sentiment Analysis of Social Media Content using N-Gram Graphs. Scottsdale: WSM 2011*.*

[9] Houvardasi I., Kanaris I., Kanaris K., & Stamatatos E. Words vs. character n-grams for anti-spam filtering. // International Journal on Artificial Intelligence Tools. – 2006. - № 20(10). – С. 1-20.

[10] Pang L.Thumbs up? Sentiment Classification using Machine Learning Techniques // Proceedings of EMNLP (2002).

[11] Domingos, P. & Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss // Machine Learning. - 1997. - № 29. – С. 103-­137.