САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

ГОРОЖАНКИН Ярослав Павлович

Выпускная квалификационная работа

Разработка системы анализа мнений отзывов о фильмах

Уровень образования: бакалавриат

Направление 01.03.02 «Процессы управления»

Основная образовательная программа СВ.5005.2015 «Прикладная математика, фундаментальная информатика и программирование»

Научный руководитель:
старший преподаватель, кафедра технологии программирования, Попова Светлана Владимировна

Рецензент: старший преподаватель, кафедра космических технологий и прикладной астродинамики, Давыденко Александр Александрович

Санкт-Петербург

2019

Содержание

Введение

С появлением Web 2.0 различные платформы, такие как ВКонтакте[1], Facebook[2], Twitter[3], Instagram[4] позволяют гражданам делиться своими комментариями, мнениями, чувствами, суждениями по множеству тем, начиная от образования и заканчивая развлечениями. Эти платформы содержат огромное количество данных в виде твитов, блогов, обновлений статуса, сообщений и т.д. Анализ мнений направлен на определение полярности эмоций, таких как счастье, печаль, горе, ненависть, гнев, привязанность, а также на выделение мнения из текстов, обзоров, постов, которые доступны онлайн на этих платформах. Анализ мнений сложен из-за сленговых слов, орфографических ошибок, коротких форм, повторяющихся символов, использования диалектов, новых смайликов и т.д. Анализ мнений является одной из наиболее активных областей исследования и широко изучается в области интеллектуального анализа данных. Применяется практически во всех сферах бизнеса и социальной сферы.

Все больше и больше людей делятся своим личным опыт с незнакомцами благодаря интернету. Существует огромное множество ресурсов с отзывами на разные тематики, будь то книги, одежда или электронные предметы, пользователь первым делом прочитает мнения об этом конкретном продукте и затем только задумается о приобретении.

Дисциплину анализу мнений можно разделить на две большие части. Первая – извлечение тональности мнения, обычно подразумевает задачу классификации текста по тональности эмоции. Вторая – извлечение мнений, когда выделяется не только эмоция, но и объект этой эмоции и что именно вызвало эмоцию.

[1] https://vk.com/
[2] https://www.facebook.com/
[3] https://twitter.com/
[4] https://www.instagram.com/

В данной работе используются обзоры на фильмы. Производители могут собирать обзоры пользователей, независимо от того, являются ли они положительным или нет, относительно фильма и в следующих своих работах попытаться повысить качество произведения киноискусства. Так к примеру на сайтах ivi[5], tvzavr[6] и кинотеатр.ру[7] пользователи не могут выставлять свои собственные оценки фильму. Поэтому важно разработать классификатор по определению тональности мнений.

[5] https://www.ivi.ru/
[6] https://www.tvzavr.ru/
[7] https://www.kino-teatr.ru/

Постановка задачи

Целью данной дипломной работы ставится улучшение качества работы алгоритмов определения тональности для их практического внедрения.

Задачи:

- Рассмотреть возможные подходы и алгоритмы к построению моделей классификации отзывов фильмов по трем классам тональности: “негативные”, “нейтральные”, “позитивные”.
- Собрать набор данных кинорецензий для тестирования моделей.
- Сравнить различные способы обработки текстовых данных и их влияние на модель классификации.
- Разработать Web-сайт для поиска необходимого фильма и рассмотрения отзывов о нем.

Обзор литературы

В качестве предметной области, представляющей тестовые коллекции, была выбрана тема фильмов. В качестве тестовых данных была использована коллекция отзывов о фильмах с портала Imhonet.ru, которая была предоставлена Российским семинаром по оценке методов информационного поиска (РОМИП), подробнее о коллекции в параграфе 2.1. В данной работе указанная коллекция выбрана в качестве основной тестовой коллекции, поэтому обзор литературы основан на анализе публикаций, в которых представлены результаты для этой коллекции.

В работе[1] описаны подходы к классификации отзывов пользователей, основанные на использовании фрагментных правил (подробнее об фрагментных правилах в параграфе 1.2.3). Правила составляются вручную экспертами, также были протестированы процедуры машинного обучения. Тестировались следующие алгоритмы машинного обучения:

- Алгоритм к-ближайших соседей;
- Алгоритм построение деревьев решений C4.5;
- Алгоритм на основе машин опорных векторов;
- Байесовский классификатор на основе смеси многомерных нормальных распределений;
- Байесовский классификатор на основе смеси распределений фон Мизеса-Фишера;
- Центроидный классификатор Роччио.

Тестирование классификаторов проходило следующим образом:

1) Формирование векторного представления текстов в рамках модели «Bag Of Words».
2) Снижение размерности (селекция признаков по частоте) и вычисление весов признаков (TF-IDF).

Следует отметить, что классификация проводилась по двух балльной шкале оценки (положительных и отрицательных отзывов). Наиболее эффективным оказался подход, основанный на ручном построении правил экспертами. Результаты оценки качества по F-мере:

1) классификатор на основе правил - 0.68,
2) модифицированный классификатор на основе правил – 0.68,
3) классификатор на основе деревьев решений – 0.50,
4) классификатор к-ближайших соседей - 0.52.

Остальные классификаторы показали низкие результаты F-меры их оценки не были представлены.

В работе[2] решаются вопросы выбора оптимальной векторной модели представления текста и наиболее подходящего метода машинного обучения. Тестировались следующие алгоритмы машинного обучения:

- Наивный байесовский классификатор.
- Метод Rocchio.
- Метод k ближайших соседей.
- Метод машин опорных векторов (SVM).
- Метода на основе ключевых слов и его комбинации с SVM.

Векторное представление текста формируется на основе подхода TF-IDF без использования информации о принадлежности текста к какому-либо классу и с использованием этой информации. Из всех текстов исключались стоп-слова, удалялись слова длиной менее трех символов, все слова приводились к словарной форме. В данной работе обсуждались разные тематики: книги, фотокамеры и фильмы. Среди представленных тем, модели, построенные по фильмам, показали наихудший результат. Результаты оценки качества для трех бальной системы оценки были представлены только для трех методов. Ниже показаны результаты по F-мере:

1) метод опорных векторов – 0.265,
2) метод ключевых слов – 0.206,
3) комбинированный метод – 0.285.

В работе[5] рассматриваются два основных подхода к проблеме анализа настроения – лексический и машинное обучение. Кроме того, в данном исследовании использовалась дополнительная коллекция отзывов с сайта Кинопоиск[8]. Тестировались следующие алгоритмы машинного обучения:

- Метод максимальной энтропии (Maximum Entropy method),
- Метод, основанный на лексиконе,
- Метод машин опорных векторов (SVM).

Словарь для алгоритмов машинного обучения составлялся при помощи оценочной лексики на русском языке для мета-области товаров[6]. Классификация проводилась по двух, трех и пяти бальной системе оценок. Наиболее эффективным оказался подход максимальной энтропии. Результаты оценки качества по F-мере:

1) Метод максимальной энтропии – 0,683,
2) Метод, основанный на лексиконе – 0,659,
3) Метод машин опорных векторов – 0,660.

Обзор литературы показывает, что, не смотря на активные исследования проблемы автоматического определения тональности текста, в частности в рамках соревнований РОМИП и конференции Диалог, применимость рядов методов для решения данной задачи остается не изученной. В своей работе я исследую возможность применения алгоритма случайного леса для классификации текстов по тональности. Как следует из приведенного обзора данный метод не был подробно рассмотрен ранее.

[8] https://www.kinopoisk.ru/

Глава 1. Обзор предметной области

В данной главе будут рассмотрены теоретические подходы обработки естественного языка и анализа тональностей. Обзор существующих подходов решения.

1.1 Обработка естественного языка

Обработка естественного языка – направление машинного обучения и компьютерной лингвистики, направленное на изучение проблемы синтеза естественных языков и компьютерного анализа. Основными направлениями обработки естественного языка являются: распознавание речи, генерация естественного языка и понимание естественного языка.

В обработке естественного языка применяется предобработка текста в формат удобный для дальнейшей работы. В этой работе использованы следующие этапы предобработки текса:

- Перевод всех букв к верхнему или нижнему регистру;
- Удаление цифр;
- Удаление пунктуации;
- Удаление стоп-слов;

  Стемминг - процесс выделения основы слова. Альтернатива для русского языка: лемматизация – приведение слова к одинаковой форме:
  - для существительных — именительный падеж, единственное число;
  - для прилагательных — именительный падеж, единственное число, мужской род;
  - для глаголов, причастий, деепричастий — глагол в инфинитиве несовершенного вида.
- Векторное представление слов - для документа создается вектор размерности словаря, в него записывается насколько часто слово встречается в документе.

1.2 Анализ тональности

Анализ тональности – класс методов анализа текстовых данных, предназначенный для определения эмоциональной окраски текста и в нахождении эмоциональной оценки авторов по отношению к объектам, речь о которых идет в тексте.

1.2.1 Виды шкал для определения тональности

В области анализа тональности текста как правило использую одну из следующих шкал разделения текстов по тональности:

1) Бинарная шкала [10]

   Два класса оценок: позитивная и негативная. Минус данного подхода в том, что не во всех случаях удается однозначно определить к какому классу относиться документ: текст может содержать признаки позитивной и негативной оценки одновременно.
2) Многополосная шкала [8]

   Расширение задачи классификации документов от оценки “положительный или отрицательны” в сторону трех и четырех бальной системе оценки.
3) Системы шкалирования [12]

   Словам ставится в соответствие числа по какой-то шкале, например, от -10 до 10 (от резко негативного до резко положительного). Текст анализируется инструментами обработки естественного языка, затем найденные термины изучаются с целью понимания значения этих терминов.

В данной работе используется многополосная трех бальная шкала.

1.2.2 Подходы к определению тональности текстов

В проблеме анализа тональности существует два основных подхода: лексический подход и подход машинного обучения. В лексическом подходе определение тональности основано на анализе отдельных слов, используются

эмоциональные словари[9]: в тексте ищутся эмоциональные лексические элементы из словаря, веса их тональности уже подсчитаны, и применяется некоторая агрегированная весовая функция для определения тональности текста на основе всех элементов.

Задача извлечения тональности текста с помощью машинного обучения рассматривается как общая проблема классификации текста[11] – деятельность по маркировке текстов на естественном языке тематическими категориями из предопределенного набора, в ней применяются заранее размеченные по тональности корпусы данных, на которых происходит обучение модели, которая в дальнейшем используется для классификации.

Формальная постановка задачи классификации текста:

Имеется множество классов $C = \{c_1, \dots, c_{|C|}\}$
Имеется множество документов $D = \{d_1, \dots, d_{|D|}\}$
Неизвестная целевая функция $F: C * D \to \{0, 1\}$
Необходимо построить классификатор F`, максимально близкий к F.

В задаче извлечения тональности из текста, классами являются сами тональности, к примеру “негативные”, “нейтральные” и “позитивные”.

У каждого подхода есть свои преимущества и недостатки. Лексическому подходу не нужны размеченные по тональности корпусы данных и процедура обучения, следовательно, решения, принятые классификатором, легко объяснимы. Однако необходимы огромные лингвистические ресурсы, такие как эмоциональный словарь. Так же термины словаря должны иметь вес, адекватный предметной области документа. Например, слово «большой» по отношению к размеру мобильного телефона является отрицательной характеристикой, но положительной по отношению к объему памяти жёсткого диска.

При машинном обучении словарь не требуется, и на практике методы демонстрируют высокую точность классификации. Но классификатор, обученный для одной области, в большинстве случаев не работает в других.

Как уже было сказано в обзоре литературы, в данной работе используется метод машинного обучения. Как следует из обзора литературы, для выбранной коллекции не было представлено результатов для одного из самых популярных алгоритмов машинного обучения – случайный лес, поэтому в данной работе выбран именно этот алгоритм машинного обучения, подробнее в параграфе 2.3.

1.2.3 Метод фрагментных правил

В обзоре литературы был упомянут метод фрагментных правил. Данный метод заключается в разбиении текста на последовательности слов – фрагменты при помощи правил. Например, простые правила $FirstUp – выделяет все слова в тексте с большой буквы, $Sentence — все предложения в документе, ‘обл*’ — все слова, начинающиеся на «обл» и т.д. Сложные правила, являются объединением простых правил они и выделяют фрагменты из текста. Полученные данные анализирую для нахождения часто используемых шаблонов которым присваивается позитивная или отрицательная оценка. Данному методу, так же, как и лингвистическому, необходимы лингвистические ресурсы для формирования правил экспертами.

1.2.4 Оценка качества анализа тональностей

Для того чтобы понять, насколько хорошо построенный алгоритм работает с данными, необходима численная метрика его качества. Для каждого класса отдельно составляется таблица классификации.

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

1) TP — истинно-положительное решение;
2) TN — истинно-отрицательное решение;
3) FP — ложноположительное решение;
4) FN — ложноотрицательное решение.

1) Полнота (Recall)

   Полнота системы – это доля найденных классификатором документов, принадлежащих классу относительно всех документов этого класса в тестовой выборке.

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

2) Точность (Precision)

   Точность системы в пределах класса – это доля документов, действительно принадлежащих данному классу относительно всех документов, которые система отнесла к этому классу.

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

3) F-мера (F-measure)

   F-мера представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю.

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

4) Cross-validation

Для оценки качества классификации используется метод кросс-валидации (cross-validation) – данные делятся на k частей. Затем на k-1 частях производится обучение модели, а оставшаяся часть используется для тестирования. Процедура повторяется k раз и в итоге каждая из k частей данных используется для тестирования.

1.2.5 Оценка качества анализа тональностей в многоклассовом случае

Так как в данной работе рассматривается трех классовая классификация приведем пример расчета метрик для данного числа классов.

	Оценка экспертов для первого класса	Оценка экспертов для второго класса	Оценка экспертов для третьего класса
Оценка системы для первого класса	a11	a12	a13
Оценка системы для второго класса	a21	a22	a23
Оценка системы для третьего класса	a31	a32	a33

Так для первого класса базовые показатели будут равны:

$$TP_1 = a_{11};\ TN = a_{22} + a_{23} + a_{32} + a_{33};\ FP = a_{12} + a_{13;}\ FN = a_{21} + a_{31}$$

Для второго:

$$TP_2 = a_{22};\ TN = a_{11} + a_{13} + a_{31} + a_{33};\ FP = a_{21} + a_{23};\ FN = a_{12} + a_{32}$$

И для третьего:

$$TP_3 = a_{33};\ TN = a_{11} + a_{12} + a_{21} + a_{22};\ FP = a_{31} + a_{32};\ FN = a_{13} + a_{23}$$

В данной работе используется подход макро-усреднения. Поэтому приведем формулы для его вычисления:

1) Вычисляем нужную метрику для каждого класса по отдельности. К примеру точность: $Precision_1 = \frac{TP_1}{TP_1+FN_1}$ и т.д.
2) Усредняем метрику: $Precision = \frac{Precision_1+Precision_2+Precision_3}{3}$

1.2.6 Обзор существующих решений

В настоящее время существует огромное число систем анализа тональности текста. Приведем некоторые из них:

1) «SentiStrength»[9] - система разработана для анализа коротких неструктурированных неформальных текстов на английском языке. Но она может быть сконфигурирована для работы с текстами других языков, в том числе и для русского. Алгоритм основан на поиске в тексте слов с максимальной тональностью для негативной и позитивной шкалы.
2) Компонент анализа тональности текста в составе систем «Аналитический курьер» и «X-files»[10] - реализует метод, основанный на словарях и правилах. Тональность оценивается по трёхбалльной шкале. Этапы работы системы:
    a. Обработка текста, выделение и классификация слов.
    b. Объединение слов в связанные друг с другом цепочки.
    c. Нахождение объектов тональности.
3) Компонент анализа тональности в составе системы RCO Fact Extractor[11] – система использует подход, основанный на правилах и учитывает синтаксическую структуру текста. Этапы работы системы:

[9] http://sentistrength.wlv.ac.uk/
[10] https://www.i-teco.ru/solutions/business_intelligence_products/analytical_courier/?sphrase_id=47148
[11] http://www.rco.ru/?page_id=3554

a. Распознавание упоминаний об объекте.
b. Отсев и синтаксический анализ конструкций, связанных с объектом.
c. Нахождение и классификация тех позиций, в которых выражается тональность.
d. Оценка общей тональности текста на основе тональности всех позиций.

Работа представленных выше систем основана на использовании словарей сентиментов и правил. Ручная разработка таких словарей и правил требует большой работы экспертов. В нашей работе мы исходим из того, что привлечение экспертов не всегда возможно, в связи с чем актуально исследование применимости методов машинного обучения для автоматического определения тональности текстов.

Глава 2. Постановка эксперимента и результаты

В этой главе речь пойдет об проектировании системы анализа тональности. Задача анализа тональности будет решаться при помощи метода машинного обучения.

2.1 Тестовые коллекции

Как уже было сказано выше, в качестве предметной области, представляющей тестовые коллекции, была выбрана тема фильмов. В качестве тестовых данных была использована коллекция отзывов о фильмах с портала Imhonet.ru, которая была предоставлена Российским семинаром по оценке методов информационного поиска (РОМИП). Коллекция была представлена в виде xml-кода, отрывок которого показан на Рис. 1.

```
- <row rowNumber="0">
    <value columnNumber="0">10</value>
    <value columnNumber="1">3</value>
    <value columnNumber="2">196076</value>
    <value columnNumber="3">23499</value>
    <value columnNumber="4">Замечательный фильм, очень рекомендую.</value>
  </row>
- <row rowNumber="1">
    <value columnNumber="0">8</value>
    <value columnNumber="1">3</value>
    <value columnNumber="2">218945</value>
    <value columnNumber="3">38132</value>
    <value columnNumber="4">Очень хороший фильм. Напоминает немного 6-е чувство. Джим
      Керри играет как всегда замечательно!</value>
  </row>
- <row rowNumber="2">
    <value columnNumber="0">7</value>
    <value columnNumber="1">3</value>
    <value columnNumber="2">190406</value>
    <value columnNumber="3">38756</value>
    <value columnNumber="4">Фильм неплохой, если бы я не читала книгу, оценила бы выше. Но
      Кинга в принципе экранизировать очень трудно так, чтобы передать все эмоции, которые
      вызывает книга. Фильм все же слабее.</value>
  </row>
```

Рис. 1. Пример обзоров из тестового набора данных

Где:

1) columnNumber=”0” – score - оценка, поставленная пользователем по 10 балльной шкале Если у отзыва стоит оценка 0, это значит, что он не оценен;

2) columnNumber=”1” – content_id - идентификатор контента (1 цифровые фотокамеры, 2 книги, 3 фильмы; используются только данные по фильмам);
3) columnNumber=”2” - element_id - идентификатор книги или фильма, о котором идет речь;
4) columnNumber=”3” - user_id - идентификатор пользователя, оставившего отзыв;
5) columnNumber=”4” - text - текст отзыва.

Корпус данных состоит из 15718 таких отрывков на тему фильмов. Распределение оценок отзывов в наборе данных показано на Рис. 2.

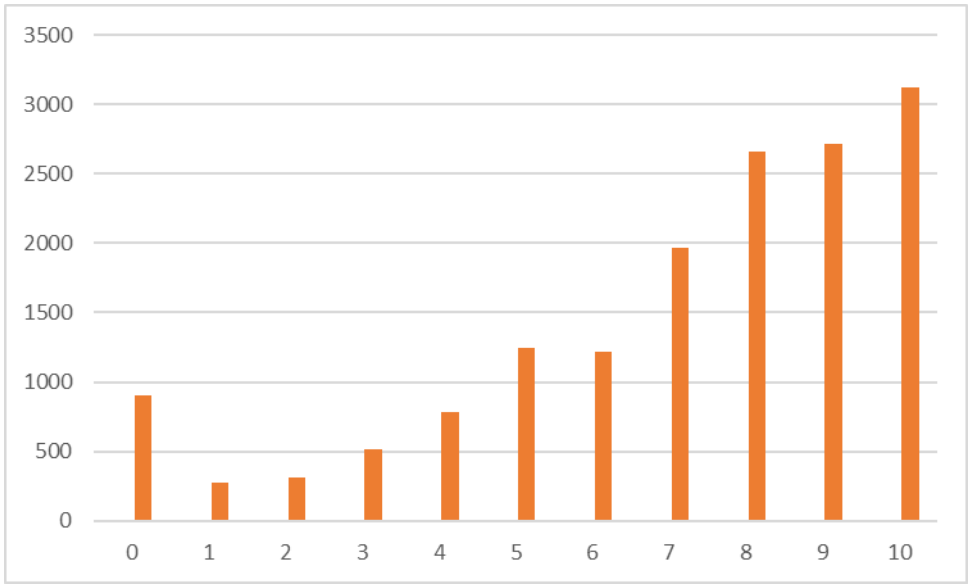

Рис. 2. Распределение оценок

Так же на языке программирования Java[12] версии 1.8, при помощи библиотеки jsoup[13] версии 1.11.3, был спроектирован Веб-краулер (описание работы краулера в параграфе 3.1) для сбора отзывов о фильмах с сайта Мегакритик[14]. Коллекция была сохранена при помощи СУБД PostgreSQL[15].

[12] http://jdk.java.net/
[13] https://jsoup.org/
[14] https://www.megacritic.ru/
[15] https://www.postgresql.org/

Отрывки таблиц жанров, фильмов и отзывов показаны на Рис. 3-5 соответственно.

film_id	genre
1	Ужасы
1	Триллеры
2	Фэнтези
2	Семейные
3	Комедии
4	Драмы
4	Триллеры
4	Мелодрамы
5	Боевики
5	Драмы
5	Военные
6	Комедии

Рис. 3. Таблица жанров

film_id	name	description	viewers_score	critics_score	img_link
1	Мы	Фильм Мы представляет соб…	3.2	7.5	Мы
2	Дамбо	Фильм Дамбо студии Дисней…	7.9	6.2	Дамбо
3	Пляжный бездельник	Фильм Пляжный бездельник…	2.3	7.1	Пляжный бездельник
4	Амбивалентность	Фильм Амбивалентность рас…	4.0	7.3	Амбивалентность
5	Балканский рубеж	Фильм Балканский рубеж ра…	8.0	6.2	Балканский рубеж
6	Трезвый водитель	Фильм Трезвый водитель пр…	2.9	5.9	Трезвый водитель
7	Королевский корги	Мультфильм Королевский ко…	3.9	5.4	Королевский корги
8	Волшебный парк Джун	Мультфильм Волшебный пар…	6.0	6.6	Волшебный парк Джун
9	В объятиях лжи	Фильм В объятиях лжи пред…	3.5	7.2	В объятиях лжи
10	Стертая личность	Фильм Стёртая личность (Ис…	5.5	6.5	Стертая личность
11	Пиковая дама: Зазеркалье	Фильм Пиковая дама: Зазер…	5.9	5.8	Пиковая дама: Зазеркалье
12	Рожденный стать королем	Фильм Рождённый стать кор…	5.5	5.0	Рожденный стать королем
13	Чайка	Фильм Чайка вдохновлён зна…	7.5	6.0	Чайка
14	Капитан Марвел	Фильм Капитан Марвел разв…	5.7	6.3	Капитан Марвел
15	Лови момент	Фильм Лови момент предста…	3.6	4.5	Лови момент
16	Ван Гоги	Фильм Ван Гоги представляе…	8.0	8.2	Ван Гоги
17	Гости	Фильм Гости представляет с…	5.0	6.3	Гости
18	Счастливого нового дня смер…	Фильм Счастливого нового д…	5.6	5.9	Счастливого нового дня смер…
19	Любовницы	Фильм Любовницы является…	3.0	5.3	Любовницы
20	Наркокурьер	Фильм Наркокурьер основан…	5.3	6.9	Наркокурьер
21	Кадавр	Фильм Кадавр (Одержимост…	7.5	3.9	Кадавр
22	Как приручить дракона 3	Мультфильм Как приручить…	8.1	7.7	Как приручить дракона 3
23	Тобол	Фильм Тобол представляет с…	5.8	4.5	Тобол
24	Омен: Перерождение	Фильм Омен: Перерождение…	3.0	5.5	Омен: Перерождение
25	30 безумных желаний	Фильм 30 безумных желаний…	9.5	5.3	30 безумных желаний
26	Алита: Боевой ангел	Фильм Алита: Боевой ангел…	7.7	6.5	Алита: Боевой ангел
27	Громкая связь	<unreadable data>	5.4	6.5	Громкая связь
28	Отрыв	Фильм Отрыв представляет…	4.8	5.8	Отрыв
29	Семь ужинов	Фильм Семь ужинов предста…	5.6	7.0	Семь ужинов
30	Айка	Фильм Айка покажет зрител…	8.5	7.5	Айка
31	Завод	Фильм Завод представляет с…	7.3	6.7	Завод
32	Идеальные незнакомцы	Фильм Идеальные незнаком…	7.5	7.3	Идеальные незнакомцы
33	Как я стал русским	Фильм Как я стал русским пр…	6.9	4.9	Как я стал русским
34	Девочка	Фильм Девочка является де…	5.6	7.4	Девочка
35	Лего Фильм 2	Мультфильм Лего Фильм 2 я…	4.0	7.2	Лего Фильм 2
36	Спасти Ленинград	Фильм Спасти Ленинград ос…	4.3	5.8	Спасти Ленинград
37	Фаворитка	Фильм Фаворитка представл…	5.9	7.8	Фаворитка

Рис. 4. Таблица фильмов

comment_id	film_id	author	text	score	score_mlt
1	1	Вероника	Ждала фильм с нетерпением, бежала на саму премьеру с радостью. И что в итоге…	3.0	1.0
2	1	Diana	Фильм оставил только негативные впечатления. Очень его ждали и спешили на п…	2.0	2.0
3	1	Владимир	Впервые в жизни пишу отзыв о фильме. Только что вышел из зала. Бежал на пре…	6.0	3.0
4	1	Александр.	Фильм полный отстой. Даже не так - полное днище. Не знаю, на кого рассчитан д…	1.0	2.0
5	1	Stock	Это что-то с чем-то, надо правильно настроиться смотреть не как ужас, а как ком…	10.0	1.0
6	1	Джейсон поджигатель	Посмотрел "Прочь" - не смотри "Мы". Неинтересный фильм с дерьмовым поворото…	3.0	2.0
7	1	Валерий	Очень свежо, оригинально, еще надо немного мозгов иметь чтобы понять, что это…	10.0	1.0
8	1	Зритель	Дерьмо полное, больше нечего сказать. В пустую потраченные деньги и время, на…	1.0	2.0
9	1	Андрей	Ребята, первый раз в жизни пишу отзыв, потому что фильм просто невероятный о…	2.0	1.0
10	1	Дмитрий	Пооооолный отстой , не смотрите . Глупый сюжет , только треллер интересный сде…	1.0	3.0
11	1	Vera	Люди в красном Непонятная для российского зрителя черная картина с ярким соц…	3.0	2.0
12	1	Зритель	Если зритель привык смотреть типичные фильмы где сюжет понятен и понятна ко…	10.0	3.0
13	1	Макс	Фильм полное дерьмо, режиссёра поставить к позорному столбу можно смело, зач…	1.0	1.0
14	1	Зритель	Фильм не понравился. Слишком затянут сюжет. Задумка интересная, но фильм н…	3.0	2.0
15	1	Карина	Фильм туфта. Фигня полная. Впервые ушли с сеанса недосмотрев и половины. По…	1.0	2.0
16	1	Петр 1	Лучше посмотрите на сварку, чем на этот фильм! Спасибо за внимание!!!	1.0	3.0
17	1	Юлиана	Кто-то пишет про скрытый смысл, так он один, забрать ваше бабло. Никому не сов…	1.0	2.0
18	1	Anna-Russia	Фильм отвратительный. Просто невероятно глупый. Ну просто невероятно... Что э…	1.0	2.0
19	1	грета	Задумка была хорошей, но, видимо, не хватило фантазии и режиссер со сценарис…	3.0	3.0
20	1	Зритель	Фильм безобразный! Даже не знаю, какие эмоции он оставил. Мы досмотрели до…	1.0	1.0
21	1	Степан	По моему фильм оригинальный в плане публикации несущей характер показать, к…	5.0	3.0
22	1	Павел	Добрый день. К сожалению, в последнее время, премьеры не завлекают, кинотеа…	7.0	1.0
23	1	Алексей	Сходил на фильм с девушкой, фильм жуткий отстой. Длинное начало ни о чём, от…	1.0	1.0
24	1	Не мы	Я такого дерьма никогда не видел. Тошнотина, тягомотина, развидеть бы. Очень…	1.0	1.0
25	1	Елена	Полный бред! Не советую идти на этот фильм. Не тратьте время и деньги! Актеры…	2.0	1.0
26	2	Валерия	Очень хороший фильм. Желаю вам сходить всей семьёй. Добрый, показывает что…	10.0	3.0
27	2	Евгения	Сегодня ходили с детьми, мы просто в восторге. Добрый фильм. В некоторых мест…	10.0	1.0
28	2	Анастасия	Очень хороший фильм! Я в восторге! Очень трогательный фильм. Я очень рекоме…	10.0	3.0
29	2	Егор	Мне 10 лет и я, сходив на Дамбо не остался равнодушным. Фильм очень красивый…	8.0	1.0
30	2	Инга	Потрясающий фильм, хочется кричать от восторга! Пошли с ребенком, и наконец…	10.0	3.0
31	2	Алёна	Фильм снимали садисты, не иначе! Очень грустный, очень нудный, через 20 минут…	3.0	1.0
32	2	Татьяна	Фильм для всей семьи, семьи , которая дорожит близкими, любит и поддерживае…	10.0	3.0
33	2	KEWE	Фильм РАЗОЧАРОВАЛ. Не ожидал что такой знаменитый режиссер снимет такой,…	1.0	2.0
34	2	GFHD	Если вы работаете в ночную смену, лучше не смотрите, будете рыдать как я.	9.0	3.0
35	3	Анна	Ужасный фильм! Хуже я не смотрела, меня реально хватило на 1 час просмотра, н…	1.0	1.0
36	3	Мария	Мне жалко потраченных денег и времени. Ощущение после просмотра фильма, чт…	1.0	2.0

Рис. 5. Таблица отзывов

Где:

1) film_id – идентификатор фильма;
2) comment_id – идентификатор отзыва;
3) name – название фильма;
4) description – описание фильма;
5) viewers_score – рейтинг зрителей;
6) critics_score – рейтинг критиков;
7) img_link – название файла в котором храниться картинка фильма;
8) author – псевдоним автора отзыва;
9) text – текст отзыва;
10) score – оценка фильму поставленная автором отзыва;
11) score_mlt – оценка используемого метода машинного обучения для данного отзыва.

Корпус данных с сайта Мегакритик состоит из 21518 отзывов. Распределение оценок отзывов в наборе данных показано на Рис. 6.

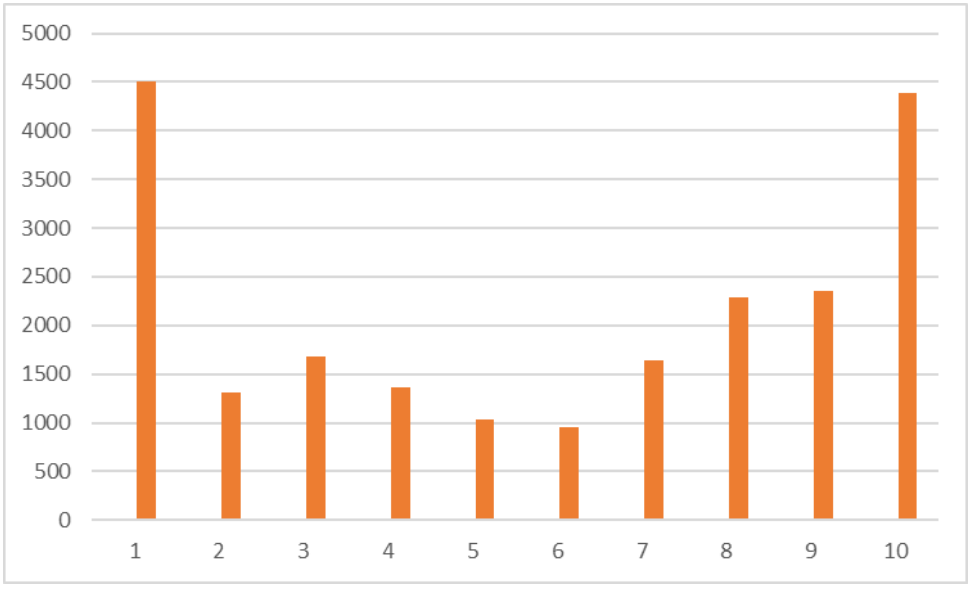

Рис. 6. Распределение оценок

На Рис. 7 продемонстрированы графики зависимости пропорции числа отзывов разной длины к числу отзывов. Среднее число слов в коллекции РОМИП и “Мегакритик” равно 55 и 45 соответственно.

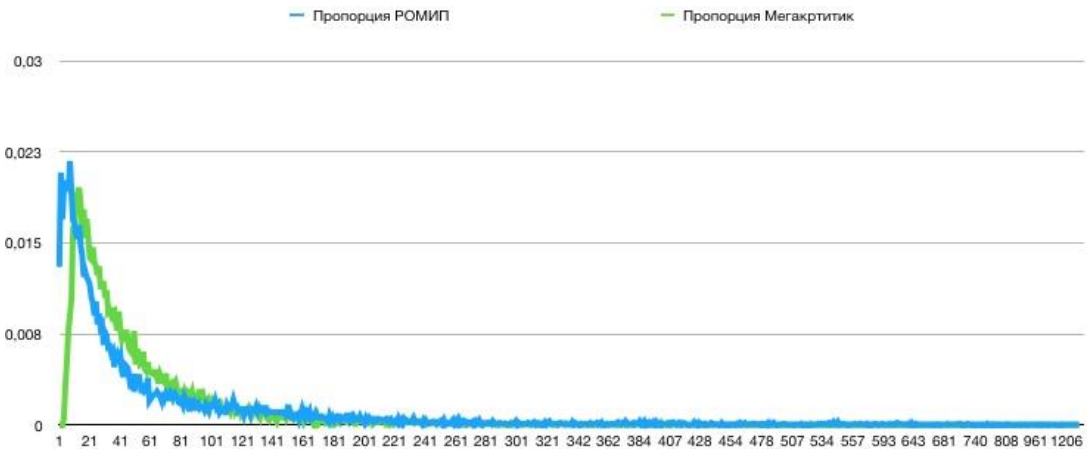

Рис. 7. Графики зависимости пропорции числа отзывов разной длины.

Так как исходные данные являются необработанным текстом, была проведена предварительная обработка для приведения документов к нормализованному виду.

1) Удалены все отзывы с неизвестной оценкой и пустым текстом.

2) При помощи MyStem[16] от Yandex произведен морфологический анализ и лемматизация.
3) Удалены союзы, местоименные наречия, местоименные прилагательные, местоименные существительные и предлоги.
4) Отдельно обработана частица “не” (удаляется пробел между частицей и словом, идущим после нее).

После всей первоначальной обработки в тестовых данных с портала Imhonet.ru осталось 14648 отзывов, а с портала Megacritic количество отзывов осталось прежним.

Пример предобработки отзыва:

До	“Посмотрели фильм в кинотеатре! Очень понравилась экранизация! Фильм прошёл легко, не затянуто, актеры сыграли на отлично! Развязка также не подвела! А в сравнении с оригиналом- достойно!”
После	“посмотреть фильм кинотеатр очень понравиться экранизация фильм проходить легко незатягивать актер сыграть отлично развязка также неподводить сравнение оригинал достойно”

2.2 Построение матрицы документ/термин

Полученные данные в параграфе 2.1 также отдельно обрабатываются при помощи библиотеки Weka[17]:

1) Обработка при помощи метода StringToWordVector.

   Данный метод преобразует строковые атрибуты в набор числовых

[16] https://tech.yandex.ru/mystem/
[17] https://www.cs.waikato.ac.nz/ml/weka/

атрибутов, представляющих информацию о появлении слова из текста, содержащегося в строках. Для каждого слова создается вектор из всех документов (отзывов). И в соответствие каждому документу ставится число встречаемости данного слова в данном документе.

Используется статистический метод TF-IDF для оценки важности слова в контексте документа, являющегося частью коллекции документов.

TF – Tern Frequency – частота слова - отношение числа вхождений некоторого слова t_i к общему числу слов документа. Таким образом, оценивается важность слова в пределах отдельного документа.

TF = $\log(1 + f_{ij})$.

IDF - Inverse Document Frequency — обратная частота документа — инверсия частоты, с которой некоторое слово встречается в документах коллекции. Учёт IDF уменьшает вес широкоупотребительных слов.

IDF = $f_{ij} \log\left(\frac{|D|}{|D_i|}\right)$.

Соответственно частоты слов преобразуются по формуле:

$TF * IDF = \log(1 + f_{ij})$*$f_{ij} \log\left(\frac{|D|}{|D_i|}\right)$, где:

f_{ij} – частота слова i в документе j,

D – число документов,

D_i – число документов с i-м словом.

Удалены стоп-слова, к таким можно отнести предлоги, суффиксы, причастия, междометия, частицы и т.п. Список стоп-слов взят сайта университета Невшателя[18].

2) Обработка при помощи метода AttributeSelection - оценка слов путем измерения прироста информации по отношению к классу.
   a) *InfoGainAttributeEval* - Оценивает ценность атрибута, измеряя прирост информации относительно класса.

$$InfoGain(Class, Attribute) = H(Class) - H(Class \mid Attribute).$$

[18] http://members.unine.ch/jacques.savoy/clef/

$H(X) = -sum(P_i * \ln(P_i))$ – Энтропия, где P_i - это вероятность класса i в наборе данных. Энтропия в основном измеряет степень «нечистоты». Чем ближе к 0, тем меньше примесей в нашем наборе данных. Следовательно, хороший атрибут – это атрибут, который содержит большую часть информации, то есть уменьшает наибольшую энтропию.

Пример принципа InfoGain продемонстрирован в приложении 1

3) Так как распределение оценок отзывов в наборе обучающих данных очень несбалансированное, что создает трудности при обучении модели. Используется метод Resample – каждый класс будет иметь одинаковое количество выборок. Если у какого-то класса количество документов (отзывов) меньше, то к данному классу будет добавлена случайная подвыборка документов из этого же класса.

Часть матрицы документ/термин, которая получается, после полной обработки всех отзывов показана на Рис. 8.

No.	1: понравиться	2: отличный	3: рекомендовать	4: деньги	5: советовать	6: полный	7: потратить	8: бред	9: хороший
	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	1.2612237...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.83135...
5	1.2612237...	1.81940...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	1.7704...	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	1.7704...	0.0	0.0	1.83135...
...	0.0	1.81940...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	1.2612237...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.15545...
...	1.2612237...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	2.0719242400...	0.0	2.059869...	0.0	0.0	0.0	0.0
...	1.2612237...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	1.859...	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	1.2612237...	0.0	2.0719242400...	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.31090...
...	1.2612237...	0.0	2.0719242400...	0.0	0.0	0.0	0.0	0.0	1.15545...
...	0.0	0.0	0.0	0.0	2.059869...	0.0	0.0	0.0	1.83135...
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	2.059869...	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	2.059869...	0.0	0.0	0.0	0.0
...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.15545...

Рис. 8. Пример матрицы документ/термин

2.3 Random Forest

Как следует из обзора литературы алгоритм случайного леса не был представлен для используемой коллекции. Поэтому в данной работе используется именно этот алгоритм.

“Случайный лес - это классификатор, состоящий из набора классификаторов с древовидной структурой: $\{h(x, \Theta_k), k = 1, \ldots\}$, где $\{\Theta_k\}$ являются независимыми одинаково распределенными случайными векторами, и каждое дерево получает единичный голос за самый популярный класс на входе x. (x – входной вектор).”[7]

Перед рассмотрением алгоритма построения случайного леса нужно понять, что такое дерево принятия решений.

Дерево решений - это инструмент поддержки принятия решений, в котором используется древовидная модель решений и их возможных последствий, включая случайные исходы событий, затраты ресурсов и полезность. Это один из способов отображения алгоритма, который содержит только условные операторы управления.

Алгоритм обучения деревьев решений:

1) Начинаем со всех имеющихся данных.
2) Выбираем атрибут, который дает «лучшее» разделение.
3) Создать дочерние узлы на основе разделения.
4) Повторяем для каждого потомка, используя дочерние данные, пока не будет достигнут критерий остановки: все примеры имеют один и тот же класс или объем данных слишком мал или дерево слишком велико.

Основная задача при построении дерева - решить, какой атрибут выбрать для разделения данных, чтобы получить «лучшее» разделение. Ответом на эту задачу является метод InfoGain описанный в параграфе 2.2.

Алгоритм построения случайного леса может быть представлен в следующем виде:

1) Случайным образом отбираем (с повторениями) n наблюдений из исходных n наблюдений.
2) Обучаем дерево по случайной подвыборке.
3) Повторяем до получения определенного числа деревьев. В случае Weka число деревьев = 100.

Прогноз случайного леса:

1) Каждое дерево дает свой прогноз.
2) Усредняем и получаем финальный прогноз.

Случайные леса обладают целым рядом привлекательных качеств, что обусловило их широкое применение, а именно:

1. Случайные леса обеспечивают существенное повышение точности.
2. Сложная задача усечения полного дерева решений снимается, так как деревья в случайном лесу не усекаются;
3. Отсутствует проблема переобучения.
4. Простота применения: единственными параметрами алгоритма являются количество деревьев и количество признаков, случайно отбираемых для расщепления в каждой вершине дерева.
5. Легкость организации параллельных вычислений.

2.4 Построение классификатора

Для построения классификатора тональности был выбран язык программирования Java с использованием библиотеки Weka. Реализация проходила в соответствии с алгоритмами в параграфах 2.1 – 2.3.

В данной работе мы рассмотрим трех-классовую классификацию текста, поэтому нужно правильно выбрать отображение 10-ти кратной системы оценки. Задача определения эмоциональной оценки текста субъективна. Согласно [9] люди могут по-разному оценить один и тот же текст. Так в [3] было выбрано отображение: {1–6} → “1”, {7–8} → “2”, {9–10} → “3”. А в [5]: {1–4} → “1”, {5–6} → “2”, {7–10} → “3”. Где “1” – “негативные”, “2” – “нейтральные”, “3” – “позитивные”. Мы так же рассмотрим еще 2 вида отображения: {1–3} → “1”, {4–6} → “2”, {7–10} → “3” и {1–3} → “1”, {4–7} → “2”, {8–10} → “3”. На Рис. 9 приведены некоторые полученные характеристики данных.

	РОМИП			Мегакритик		
	позитивные	нейтральные	негативные	позитивные	нейтральные	негативные
1	5754	4590	4304	6737	3931	10850
2	10344	2446	1858	10668	1991	8859
3	10344	3225	1079	10668	3357	7493
4	8393	5176	1079	9029	4996	7493

Рис. 9. Характеристики данных.

Где:

1 – отображение вида {1–6} → –“негативные”, {7–8} → “нейтральные”, {9–10} → “позитивные”;
2 – отображение вида {1–4} → –“негативные”, {5–6} → “нейтральные”, {7–10} → “позитивные”;
3 – отображение вида {1–3} → –“негативные”, {4–6} → “нейтральные”, {7–10} → “позитивные”;

4 – отображение вида {1–3} → –“негативные”, {4–7} → “нейтральные”, {8–10} → “позитивные”.

Входные данные для классификатора - матрица документ/термин. Входные данные получены следующими образами с использованием методов описанных в параграфе 2.2:

1) Приведение данных к виду матрицы документ/термин при помощи алгоритма StringToWordVector с использованием метода TF-IDF.
2) При помощи алгоритма AttributeSelection с использованием метода InfoGain из матрицы удаляются термины с малым приростом информации относительно классов.
3) Применяется метод Resample чтобы избавиться от несбалансированности данных.

Протестированы две возможности:

1) Входные данные делят на тестовые, те на которых модель будет тестироваться и на обучаемые, те на которых модель обучается, после использования описанных выше методов.
2) Использование описанных выше методов происходит только на обучаемой выборке.

2.5 Результаты

На Рис. 10 продемонстрированы результаты экспериментов с использованием методов 10-кратной кросс-валидации и F-меры представленными в параграфе 1.2.4.

		РОМИП				Мегакритик			
		1	2	3	4	1	2	3	4
Обрабатывались только обучаемые данные	STWV	0.624	0.64	0.528	0.525	0.699	0.705	0.763	0.682
	STWV AS	0.617	0.649	0.517	0.52	0.699	0.709	0.757	0.691
	STWV AS R	0.6	0.642	0.502	0.507	0.716	0.743	0.766	0.702
Тестовые и обучаемые данные обрабатывались вместе	STWV	0.627	0.634	0.526	0.531	0.701	0.706	0.762	0.685
	STWV AS	0.623	0.646	0.52	0.525	0.698	0.709	0.757	0.687
	STWV AS R	0.816	0.814	0.769	0.773	0.889	0.909	0.931	0.890

Рис. 10. Результаты Экспериментов

Где:

- 1 – Отображение вида {1–3} → “негативные”, {4–7} → “нейтральные”, {8–10} → “позитивные”;
- 2 - Отображение вида {1–3} → “негативные” {4–6} → “нейтральные”, {7–10} → “позитивные”;
- 3 - Отображение вида {1–4} → “негативные”, {5–6} → “нейтральные”, {7–10} → “позитивные”;
- 4 - Отображение вида {1–6} → “негативные”, {7–8} → “нейтральные”, {9–10} → “позитивные”;
- STWV – StringToWordVector;
- AS – AttributeSelection;
- R – Resample.

2.5 Выводы

В результате построенных моделей и сравнения полученных данных можно сделать следующие выводы:

- Полученные в этой работе значения F-меры по коллекции РОМИП, при использовании методов StringToWordVector, AttributeSelection и Resample для всех видов отображения из параграфа 2.4, являются высокими относительно работ других авторов. Все остальные методы из параграфа 2.4 показали значения ниже относительно работ других авторов.
- Для разных данных важно подбирать правильное отображение 10 бальной системы оценки. Это может быть связано как со временем сбора коллекции, сайтом с которого собиралась коллекция, так и с изначальным распределением самих данных. Так к примеру, для коллекции Мегакритик третий вид отображения показывает наилучшие результаты для всех видов обработки. А для коллекции РОМИП лучшим по большинству видов обработки был второй тип отображения;
- Не все методы обработки данных всегда положительно сказываются на качестве финальной модели. К примеру одновременное использование методов StringToWordVector и AttributeSelection в большинстве случаев показывает результаты ниже, чем просто использование метода StringToWordVector.
- Совместная обработка данных на которых строится и тестируется модель не во всех случаях дает большой прирост к качеству модели, а в некоторых случаях и ухудшает это качество. К примеру, для коллекции РОМИП в наилучшем отображении, при использовании методов StringToWordVector и AttributeSelection результаты для совместной обработки хуже, чем, когда обрабатывается только обучаемые данные.

Глава 3. Разработка и создание Web-сайта

Глава посвящена разработке и создании Web-сайта для поиска необходимого фильма и просмотра отзывов о нем. Для отображения элементов пользовательского интерфейса и взаимодействия с сервером на стороне клиента использовался фреймворк Vaadin[19]. Для организации быстрого поиска использовалась поисковая машина ElasticSearch[20]. Данные об отзывах хранятся в базе данных PostgreSQL. Данные сначала были сохранены в базе данных. Затем на этих данных был построен классификатор из параграфа 2.4 и обработаны отзывы. Схема сайта представлена на Рис. 11

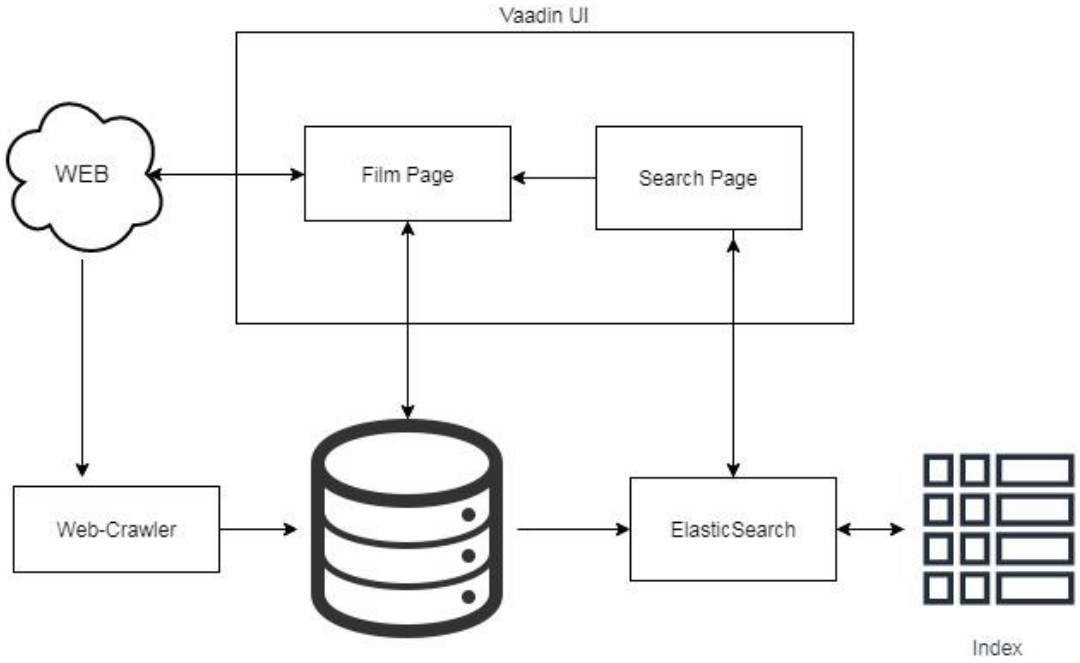

Рис. 11. Схема сайта

3.1 Web-crawler

Web-crawler или поисковый робот - это программы, цель которых автоматический поиск документов в Web'е, их индексация в Информационно Поисковых Систем. Они также обновляют информацию документов, уже находящихся в индексе.

В данной работе был разработан поисковый робот ради одной цели – собрать отзывы о фильмах с сайта Megacritic. Поэтому он не может

[19] https://vaadin.com/
[20] https://www.elastic.co/

обновлять информацию, эта проблема решаема, но затрагивать мы ее не будем. Далее будет описана архитектура поискового робота.

Web-crawler состоит из двух классов: WebCrawler и CrawlerLeg

1. WebCrawler или “голова” – управляет CrawlerLeg, хранит множество посещенных фильмов(ссылки) и множество фильмов для посещения, передает CrawlerLeg ссылку для обработки. Получает от CrawlerLeg всю информацию об фильме и сохраняет в базу данных
2. CrawlerLeg или “рука” - скачивает HTML файл, находящийся по получаемой от WebCrawler ссылке, находит в данном файле ссылки на следующие фильмы, передает WebCrawler название, описание, жанры, оценки критиков, оценки пользователей, отзывы пользователей и изображения фильмов.

3.2 ElasticSearch

ElasticSearch - это поисковая система, основанная на библиотеке Lucene[21]. Она предоставляет распределенную полнотекстовую поисковую систему с веб-интерфейсом HTTP и JSON-документами без схемы, разработана на Java. ElasticSearch осуществляет индексацию тех полей в базе данных, которые будут использоваться при поиске, к примеру в разрабатываемом в данной работе сайте будут индексироваться названия, описания и жанры фильмов. После того, как данные оказываются в индексе, они становятся доступными для осуществления поиска. Находит документы, которые соответствуют любому полю, но ранжирует эти документы по наибольшему весу искомого термина в документе. Все остальные настройки оставлены по умолчанию.

Вес термина состоит из трех частей – Tern Frequency (частота слова), Inverse Document Frequency (обратная частота документа) и field-length norm (нормальная длина поля). Про частоту слова и обратную частоту документа

[21] http://lucene.apache.org/

уже было рассказано в параграфе 2.2, но в ElasticSearch они имеют другие формулы:

$$TF = \sqrt{f_t}$$

$$IDF = 1 + \log\left(\frac{D}{D_t + 1}\right)$$

Нормальная длина поля вычисляется по формуле:

$$norm = \frac{1}{\sqrt{n_t}}$$

Где:

f_t – частота термина в документе

D – число документов в индексе

D_t – число документов, содержащих термин.

n_t – количество терминов в отдельном поле (название, описание или жанры фильма).

Эти три фактора - частота термина, частота обратного документа и нормальная длина поля - вычисляются и сохраняются в индексе. Вместе они используются для расчета веса одного термина в конкретном документе.

Пример вывода поисковой системы по запросу “Мстители” показан на рис. 12.

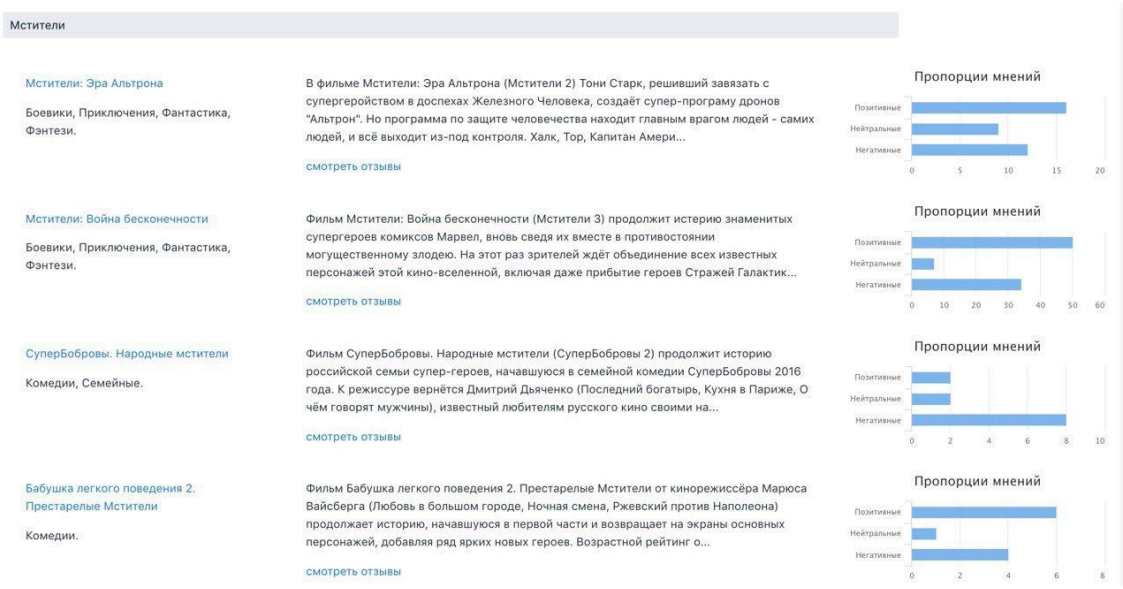

Рис. 12. Пример вывода по запросу “Мстители”

3.3 Страница фильма.

На странице фильма выводится вся информация о самом фильме, отзывы о нем и пропорции отзывов. Пример страницы фильмов показан на рис. 13 и пример отзывов, что располагаются на той же странице, но чуть ниже на рис. 14. Так же отзывы можно сортировать по тому, как они были классифицированы.

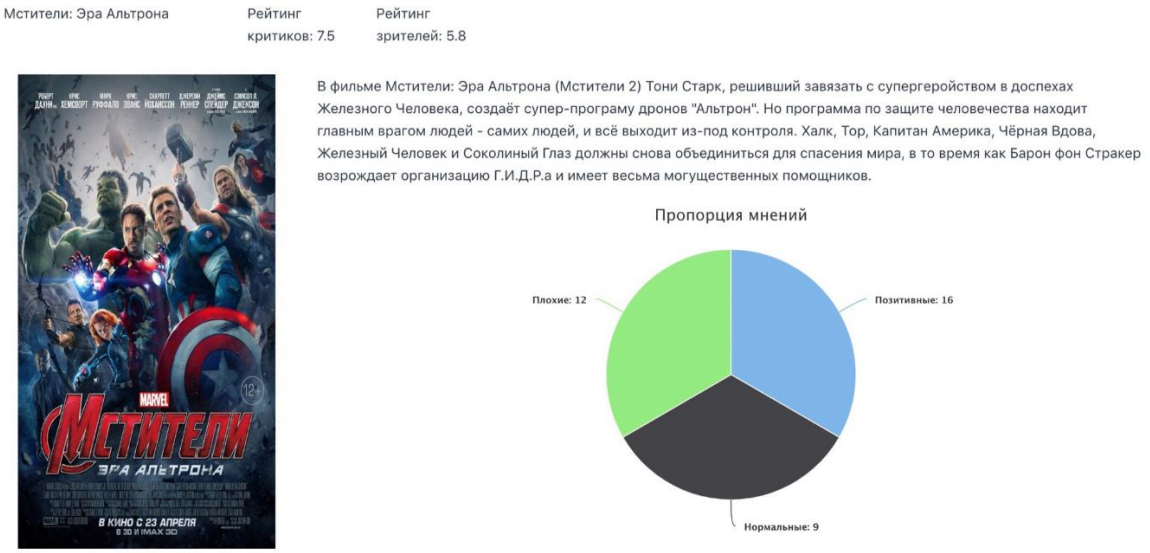

Рис. 13. Пример страницы фильма

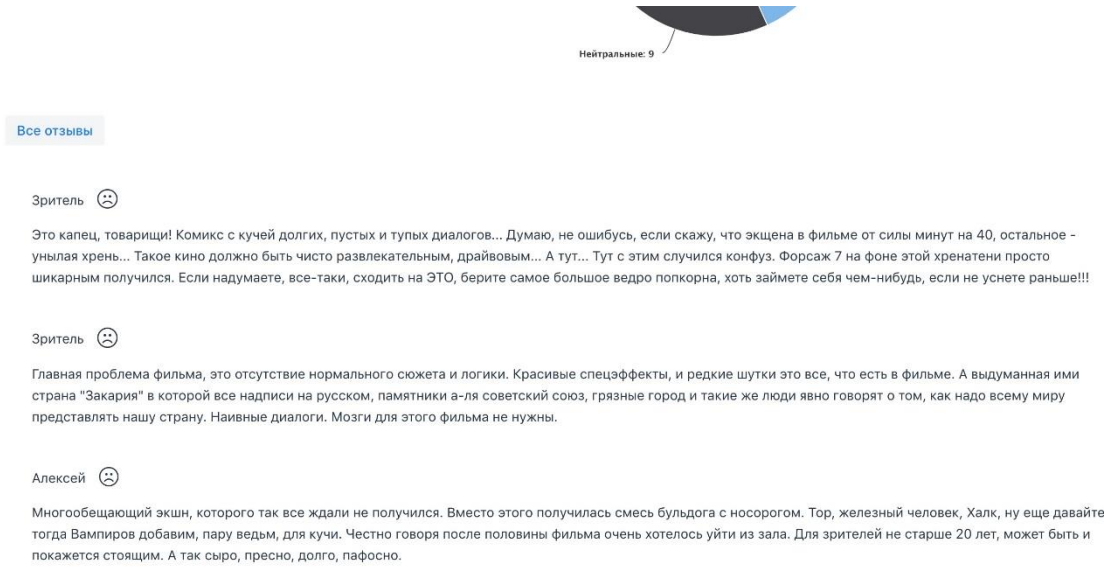

Рис. 14. Пример отзывов на странице фильма

Здесь смайликами обозначено то, как модель классифицировала отзыв.

Заключение

Все поставленные в работе задачи выполнены. Целью работы было улучшение качества работы алгоритмов определения тональности для их практического внедрения. Для достижения этой цели в работе были выполнено следующее:

- Рассмотрены возможные подходы и алгоритмы к построению моделей классификации отзывов фильмов по трем классам тональности: “негативные”, “нейтральные”, “позитивные”.
- Собраны наборы данных кинорецензий для тестирования моделей.
- Сравнены различные способы обработки текстовых данных и их влияние на модель классификации.
- Разработан Web-сайт для поиска необходимого фильма и рассмотрения отзывов о нем.

Список литературы

1. Васильев В. Г., Худякова М. В., Давыдов С. Классификация отзывов пользователей с использованием фрагментных правил // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог». Выпуск 11. Том 2. Бекасово: РГГУ, 2012. С. 66-76.
2. Котельников Е.; В., Клековкина М. В. Автоматический анализ тональности текстов на основе методов машинного обучения // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог». Выпуск 11. Том 2. Бекасово: РГГУ, 2012. С. 27-36
3. Четверкин И. И. Тестирование подхода к классификации отзывов об объектах из различных предметных областей — РОМИП 2011 // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог». Выпуск 11. Том 2. Бекасово: РГГУ, 2012. С. 15-26
4. Чистяков С. П. СЛУЧАЙНЫЕ ЛЕСА: ОБЗОР // Труды Карельского научного центра РАН № 1. 2013. С. 117–136.
5. Blinov P. D., Klekovkina M. V., Kotelnikov E. V., Pestov O. A. Research of lexical approach and machine learning methods for sentiment analysis. // Компьютерная лингвистика и интеллектуальные технологии. По материалам ежегодной Международной конференции «Диалог». Выпуск 12. Том 2. Бекасово: РГГУ, 2013. С. 51-61.
6. Ilia Chetviorkin; Natalia Loukachevitch. Extraction of Russian Sentiment Lexicon for Product Meta-Domain // Proceedings of COLING 2012: Technical Papers, P. 593–610.
7. Leo Breiman: Random Forests. Machine Learning. 45(1):5-32.

8. Bo Pang, Lillian Lee. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales // In Proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL): журнал. University of Michigan, USA, 2005. P. 115–124.
9. Pang, B., Lee, L. Opinion Mining and Sentiment Analysis // Foundations and Trends® in Information Retrieval. Vol. 2. 2008. P. 1-135.
10. Peter Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews // Proceedings of the Association for Computational Linguistics. 2002. P. 417–424.
11. Sebastiani F. (2002), Machine learning in automated text categorization, ACM Computing Surveys, Vol. 34, P. 1–47.
12. Thelwall Mike, Buckley Kevan, Paltoglou Georgios, Cai Di, Kappas Arvid. Sentiment strength detection in short informal text // Journal of the American Society for Information Science and Technology: журнал. 2010. P. 2544–2558.

Приложение

1 Пример принципа InfoGain.

Температура	Скорость ветра	Класс
Высокая	Низкая	Играть
Низкая	Низкая	Играть
Высокая	Низкая	Играть
Низкая	Высокая	Отменить
Низкая	Низкая	Играть
Высокая	Высокая	Отменить
Высокая	Низкая	Играть

Тогда $H(\text{Класс}) = -\left(\frac{5}{7} * ln\left(\frac{5}{7}\right) + \frac{2}{7} * ln\left(\frac{2}{7}\right)\right) = 0.598$

Давайте рассчитаем для нашего примера количество информации, переносимой атрибутом температуры.

$$InfoGain(\text{Класс}, \text{Температура}) = H(\text{Класс}) - H(\text{Класс} \mid \text{Температура}).$$

Чтобы получить $H(\text{Класс} \mid \text{Температура})$, нам нужно разделить набор данных в соответствии с этим атрибутом. Получатся 2 таблицы:

Температура	Скорость ветра	Класс
Высокая	Низкая	Играть
Высокая	Низкая	Играть
Высокая	Высокая	Отменить
Высокая	Низкая	играть

Температура	Скорость Ветра	Класс
Низкая	Низкая	Играть
Низкая	Высокая	Отменить
Низкая	Низкая	Играть

Каждая таблица здесь имеет свою энтропию. Нам нужно сначала рассчитать энтропию каждого разделения.

$$H(\text{верхняя}) = -\left(\frac{3}{4}ln\left(\frac{3}{4}\right) + \frac{1}{4}log\left(\frac{1}{4}\right)\right) = 0.562$$

$$H(\text{нижняя}) = -\left(\frac{1}{3}ln\left(\frac{1}{3}\right) + \frac{2}{3}ln\left(\frac{2}{3}\right)\right) = 0{,}636$$

H (класс | температура) тогда равен сумме энтропии обоих таблиц, взвешенной по доле случаев, взятых из изначального набора данных:

$$H(\text{Класс} \mid \text{Температура}) = 4/7 * H(\text{верхняя}) + 3/7 * H(\text{нижняя}).$$

Теперь у нас есть все, чтобы рассчитать InfoGain. В этом примере это 0.004. Это означает, что температурный элемент уменьшает глобальную энтропию только на 0,004, вклад функции в уменьшение энтропии (прирост информации) довольно мал. Это очевидно, если взглянуть на примеры в наборе данных, поскольку на первый взгляд видно, что температура не сильно влияет на конечный класс, в отличие от характеристики ветра.