КОМПЬЮТЕРНАЯ ЛИНГВИСТИКА И ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА

УДК 004.852

РАЗРАБОТКА СИСТЕМЫ АНАЛИЗА ТОНАЛЬНОСТИ ТЕКСТОВОЙ ИНФОРМАЦИИ

В. В. Гаршина, К. С. Калабухов, В. А. Степанцов, С. В. Смотров

Воронежский государственный университет

Поступила в редакцию 18.09.2017 г.

Аннотация. В статье проанализированы подходы для автоматического определения тональности текстовых данных, произведен сравнительный анализ методов и алгоритмов машинного обучения для решения задачи классификации тональности текста, приводится описание разработанного программного обеспечения для выделения тональности текстовых данных, реализующее подход на основе метода машинного обучения с учителем с оптимальным набором параметров для классификации.
Ключевые слова: анализ тональности текста, эмоциональная окрашенность, машинное обучение, классификация, текстовые данные.
Annotation. The article suggests an approach to automatic determination emotional coloration of text data, comparative analysis of methods and algorithms of machine learning for solving the problem of sentiment analysis of the text, description of the developed software for highlighting the key of text data is realized. It implements the approach based on the machine learning method with the teacher with the optimal set of parameters for classification.
Keywords: sentiment analysis of the text, emotional coloration, machine learning, classification, text data.

ВВЕДЕНИЕ

Непрерывное развитие социальных сетей, различного рода блогов в сети интернет, а также всевозможных ресурсов, где люди высказывают своё мнение в виде текстовых сообщений, привело к росту интереса к задачам, связанным с извлечением и анализом мнений пользователей в автоматическом режиме. К наиболее актуальным задачам, из числа анализа текстовых данных, относится определение эмоциональной окрашенности текста (мнения человека) определённой полярности, направленное к объекту высказывания. Применение методов, направленных для анализа тональности текстовой информации реализуется не только в качестве научно-исследовательских инструментов, но и в коммерческой деятельности.

Активное использование подходов и решений, применяемых при анализе тональности текстов, отмечается в следующих областях: разработка систем рекомендаций; аналитика общественного мнения; извлечение отношения пользователя к продукту. Данная область исследований является актуальным направлением. Можно указать ряд конференций международного уровня: «TACL», «ICAICT», «Диалог», «АИСТ», представляющих проблемы компьютерной лингвистики и анализа тональности текста, подтверждающие данный факт.

1. ПОДХОДЫ ДЛЯ ВЫДЕЛЕНИЯ ЭМОЦИОНАЛЬНОЙ ОКРАШЕННОСТИ ТЕКСТА

Для выделения бинарной эмоциональной окрашенности текста в автоматическом режиме используется подходы, основанные на лингвистических правилах и методах машинного обучения.

Алгоритмы, создаваемые на основе правил, позволяют учитывать семантические, структурные особенности различных слов и самого языка, но их реализация сталкивается с рядом проблем:

– Требуется формировать определённый корпус различных лингвистических правил, который обязан учитывать обширную часть различных конструктивных языковых особенностей. Данный аспект требует привлечения групп экспертов в области лингвистики.

– Узкая сфера применения набора правил в связи с тем, что формат написания различных сообщений в сети Интернет достаточно сильно отличается от принятых норм русского языка в литературной форме. Сообщения, публикуемые в социальных сетях, отличаются тем, что они содержат ошибки пунктуационного и орфографического характера, имеют место для применения различных опечаток и словесного сленга, своеобразной пунктуации, а также использование специальных символов и графических обозначений для усиления эмоциональной окрашенности текста (эмотиконов).

– Привязка к языку анализируемого текста всегда связана с уникальной языковой структурой и не может быть перенесена и применена для другого языка.

Использование подхода, основанного на лингвистических правилах, может обеспечить высокие показатели результативности лишь в тех случаях, когда анализируемые тексты будут грамматически верны, а так же если различные конструкции анализируемого языка будут покрыты корпусом правил.

Применение методов машинного обучения, подразумевает наличие некоторого набора входных данных, применяемых для обучения классификатора, и в свою очередь реализует алгоритмы обучения с учителем (применяются для обучения размеченные примеры) и без учителя (использующие методы автоматической классификации). Использование метода обучения без учителя подразумевает распределение текстов на определенное число кластеров, которое задаётся заранее. Впоследствии производится присвоение соответствующей метки, характеризующей полярность в режиме автоматического или ручного формата для каждого из кластеров. При применении метода обучения с учителем, требуется наличие текстового корпуса, который заранее размечается метками полярности, в свою очередь идентифицирующими полярность для каждого текста из корпуса, а определение тональности производится непосредственно автором текста, либо экспертом или их группой.

Применение подходов, основываемых на методах машинного обучения, позволяет подстраиваться под языковые особенности, включать в учёт дополнительные признаки, производить обработку текстов, которые, с точки зрения принятых языковых правил, являются грамматически неверными. К минусам этих подходов можно отнести некоторый проигрыш в отношении качества обработки и интерпретирования различных конструкций языка с высоким уровнем сложности.

В данном исследовании к решению задачи определения эмоциональной окрашенности текстов применялся подход, основанный на методах машинного обучения с учителем.

2. ВЕКТОРНОЕ ПРЕДСТАВЛЕНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ

Векторное представление документа (модель Bag-of-Words) W является параметризированной функцией $W : words \to R^n$, отображающей документ из естественного языка в n-мерный вектор.

Векторное представление данных можно описать следующей формулой:

$$W(\text{'word and another word'}) = \\ = (w_{j,1}, w_{j,2}, \ldots, w_{j,n}), \quad (1)$$

где W – является векторным представлением анализируемого документа, $w_{j,i}$ – является

весом термина i в текстовых данных, а j, n – является общим количеством терминов в векторном пространстве.

Функция отображения W задается с помощью таблицы поиска через специальную матрицу θ, определяющей однозначные соответствия для каждого из слов и строки $W_\theta(word_n) = \theta_n$.

Данная модель предоставляет векторное представление информации на естественном языке, согласно которой текстовые данные задаются в форме неупорядоченного набора терминов, без какого-либо указания сведений о возможных между ними связях.

3. ПРИЗНАКИ ДЛЯ КЛАССИФИКАЦИИ

В качестве признаков классификации могут использоваться: термины (n-граммы), словоформы, а также различные символьные последовательности.

К n-граммам относятся последовательности слов, стоящих определённым образом друг за другом длинной равной N.

В случае, если $n > 1$, то благодаря n-граммам будет возможен учёт контекста слова. Среди n-грамм выделяются три вариативных категории: униграммы, где $n = 1$; биграммы, где $n = 2$; триграммы, где $n = 3$.

Если анализируемый текст будет состоять из n количества предложений, а также m количества уникальных терминов, то матрица θ будет состоять из m количества строк и n количества столбцов.

Для каждого термина в словаре будет иметься вес $w_{i,j}$, где i – является порядковым номером термина в словаре, а j – является порядковым номером предложения.

Качество классификации можно повысить за счет специальных процедур по предварительной текстовой обработке, направленных на улучшение входных данных:

– приведение всех символов, встречающихся в текстовых данных к нижнему регистру. Данная процедура позволяет в значительной степени уменьшить общее количество терминов в словаре, являющихся уникальными.

– удаление символов, которые не являются буквенными. Данная процедура значительно уменьшает количество уникальных терминов в словаре, в случае активного применения различной авторской пунктуации.

– удаление символов, которые являются повторными. Процедура позволяет заменить последовательности одинаковых символов, что в свою очередь уменьшает размерность словаря.

– процедура стемминга, предназначенная для выделения основы слова из набора входных текстовых данных.

Данные действия, производимые непосредственно перед процессом классификации текстовых данных, позволяют в значительной мере сократить словарь терминов, который генерируется при обучении.

4. МЕТОДЫ КЛАССИФИКАЦИИ

Рассмотрим методы машинного обучения, используемые в исследовании для задачи выделения эмоциональной окрашенности текстов: метод машин опорных векторов SVM; метод K ближайших соседей; наивный Байесовский классификатор с мультиномиальным распределением.

Наивный Байесовский классификатор основывается на теореме Байеса, с учётом строгого предположения и независимости. Если дан класс j, а также набор признаков $z_1, \ldots, z_n$, тогда запись теоремы будет осуществляться следующим образом:

$$P(j \mid z_1, \ldots, z_n) = \frac{P(z_1, \ldots, z_n \mid j) P(j)}{P(z_1, \ldots, z_n)}, \qquad (2)$$

где $P(h \mid k)$ – является вероятностью наступления события h при событии k, а $P(h)$ – является вероятностью наступления события h.

При использовании наивного предположения о независимости, получим следующее:

$$P(z_i \mid j, z_1, \ldots z_{i-1}, z_{i+1}, \ldots, z_n) = P(z_i \mid j). \qquad (3)$$

Для всех x_i записывается следующим образом по формуле:

$$P(j \mid z_1, \ldots, z_n) = \frac{\prod_{i=1}^{n} P(z_i \mid j) P(j)}{P(z_1, \ldots, z_n)}. \qquad (4)$$

Если $P(x_1, \ldots, x_n)$ – является константой, задаваемой входными данными, то будет возможным использование следующего классифицирующего правила:

$$P(j|z_1,\ldots,z_n) \propto \prod_{i=1}^{n} P(z_i|j)P(j) \qquad (5)$$

$$\Downarrow$$

$$\hat{j} = \operatorname{argmax}_j \prod_{i=1}^{n} P(z_i|j)P(j). \qquad (6)$$

Если n будет являться достаточно большим, возникнет ситуация перемножения большого количества элементов, из-за чего может проявиться проблема переполнения. Для того, чтобы исключить данную проблему, применяется функция логарифмирования, причём основание логарифма в данном случае не имеет ключевого значения:

$$\ln \hat{j} = \operatorname{argmax}_j \ln \sum_{i=1}^{n} P(z_i|j)P(j) = \\ = \operatorname{argmax}_j \left[\ln P(j) + \sum_{i=1}^{n} \ln P(z_i|j) \right]. \qquad (7)$$

Оценка вероятностей событий производится на входных текстовых данных в форме обучающей выборки по следующей формуле:

$$P(j) = \frac{L_j}{L}, \qquad (8)$$

где L – является суммарным количеством текстовых данных, находящихся в обучающей выборке, а L_j – является общим количеством данных в классе j.

Для оценки вероятности встречи термина в классе, для мультиномиального случая оценивается по формуле:

$$P(z_i|j) = \frac{Z_{ij}}{\sum_{i' \in Y} Z_{i'j}}, \qquad (9)$$

где Y – является словарём терминов в обучающей выборке, а $Z_{i'j}$ – является общим количеством встречаемостей термина i в классе j.

Мультиномиальное распределение является обобщением биномиального распределения, принятого использовать в случае проведение независимых испытаний с различными исходами.

Однако в формуле для оценки вероятности встречаемости термина в классе имеется значительный недостаток: если в процессе классификации встречается термин, отсутствующий в обучающей выборке, то будет считаться, что $Z_{ij} = 0$, тогда, следовательно $P(z_i|j) = 0$.

Следует учитывать то, что решение данной проблемы одним лишь путём обработки обширного числа терминов не будет являться возможным и допустимым. Связано это с тем фактором, что создать подобную обучающую выборку, содержащую различные: слова с опечатками, различные словоформы, сленговые выражения, а также неологизмы – является задачей близкой к невозможной.

Для решения данной проблемы используется аддитивное сглаживание по Лапласу:

$$P(z_i|j) = \frac{Z_{ij}+1}{\sum_{i' \in Y}(Z_{i'j}+1)} = \frac{Z_{ij}+1}{\|Y\| \sum_{i' \in Y} Z_{i'j}}. \qquad (10)$$

Исходя из данной формулы, следует то, что оценка вероятностей будет смещаться в сторону менее вероятных исходов.

Следовательно, итоговое решающее правило примет следующий вид:

$$\hat{j} = \operatorname{argmax}_{j \in J} \ln \frac{L_j}{L} + \sum_{i=1}^{n} \frac{Z_{ij}+1}{\|Y\| \sum_{i' \in Y} Z_{i'j}}, \qquad (11)$$

где J – является набором возможных классов для классификации.

Классификатор на основе метода машин опорных векторов SVM при обучении модели представляет каждый объект входных данных в виде вектора R^n.

Для того, чтобы разделить объекты на несколько классов с использованием метки класса y, принимающей значения 1 или –1, находится гиперплоскость, имеющая максимальное расстояние между опорными векторами.

Пусть $w \in R^n$ – является нормальным вектором к разделяющей гиперплоскости, а параметр $\frac{b}{\|w\|}$ – является расстоянием от гиперплоскости до начала координат. Тогда задачей метода будет являться нахождение таких параметров w и b, чтобы для нового объекта $\tilde{d}$ выполнялось следующее:

$$w * \tilde{d} > b \Rightarrow \tilde{y} = 1 \qquad (12)$$

$$w * \tilde{d} < b \Rightarrow \tilde{y} = -1, \qquad (13)$$

где $w * d_i = b$ – является уравнением гиперплоскости, полученное по обучающей выборке $i = 1,\ldots,m$. Для всех векторов d_i из обучающей выборки:

$$w * d_i - b \geq 1, \text{, при } y_i = 1 \qquad (14)$$

$$w * d_i - b \leq -1, \text{ при } y_i = -1 \qquad (15)$$

Получим неравенство, задающее расстояние между объектами разных классов:

$$-1 < w * d_i - b < 1. \quad (16)$$

Тогда метод можно представить в следующем виде:

$$y_i(w * d_i - b) \geq 1, \text{ при } \max_{b,w} \frac{2}{\|w\|}. \quad (17)$$

В том случае, когда объекты являются не линейно разделимыми, вводится набор дополнительных переменных $\xi_i \geq 0$:

$$\begin{cases} y_i(w * d_i - b) \geq 1 - \xi_i \\ \min_{w,b} \|w\| + \lambda \sum_{i=1}^{m} |\xi_i| \end{cases}, \quad (18)$$

где λ – является варьируемым параметром.

Классификатор на основе метода K-ближайших соседей является метрическим алгоритмом классификации.

При классификации анализируемому объекту присваивается тот класс, который наиболее часто встречается среди его k ближайших соседей (k – является нечётным числом больше нуля).

Соседи берутся из обучающей выборки, на которой осуществляется тренировка классификатора.

Для применения метода к выборкам с большой размерностью необходимо определить функцию дистанции. Как правило, берётся функция расстояния между точками в Евклидовом пространстве, либо косинусная близость.

Евклидово расстояние является геометрическим расстоянием между векторами в многомерном пространстве. Для вектора $a = (A_1, \ldots A_n)$ и вектора $b = (B_1, \ldots B_n)$, Евклидово расстояние *distance* рассчитывается следующим образом:

$$distance(a,b) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}, \quad (19)$$

где A_i и B_i – являются координатами векторов a и b соответственно, а n – размерность пространства.

5. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

В качестве средств реализации использовались следующие программные продукты и технологии: язык разработки: Python 3.4; среда разработки: JetBrains PyCharm 2017.1.; прикладные библиотеки обработки текстовых данных под Python.

Разработан программный модуль SentimentTextClassifier, реализующий метод машинного обучения с учителем для методов: машин опорных векторов SVM; K ближайших соседей; наивный Байесовский классификатор с мультиномиальным распределением.

Процесс обработки текста включает следующие этапы:

1. Процесс предобработки входных текстовых данных
2. Процесс извлечения из входных данных признаков, необходимых для представления информации в векторной форме.
3. Процесс обучения алгоритма классификации на обучающей выборке
4. Процесс классификации объектов исходя из решающего правила алгоритма
5. Процесс тестирования качества классификации алгоритма

В качестве данных, применяемых для обучения классификатора, использовался корпус коротких текстов Юлии Рубцовой [3], который содержит в себе около 112 тысяч записей (русскоязычных twitter-постов, разбитых на классы положительной и отрицательной тональности).

На рис. 1 представлена функциональная структура работы программного модуля анализа тональности текста, представляющая собой определённую последовательность действий при обучении модели, а также при использовании обученной модели.

Для повышения качества работы классификационного алгоритма и уменьшения размерности словарей терминов, реализовываются следующие этапы предварительной обработки: приведение текста к нижнему регистру; удаление стоп-слов; удаление всех символов, которые не являются буквенными; удаление повторяющихся символов; удаление слов, которые встречаются реже всего; процедура стемминга.

Первоначально выполняется процесс обучения классификатора, который включает в себя следующую последовательность действий:

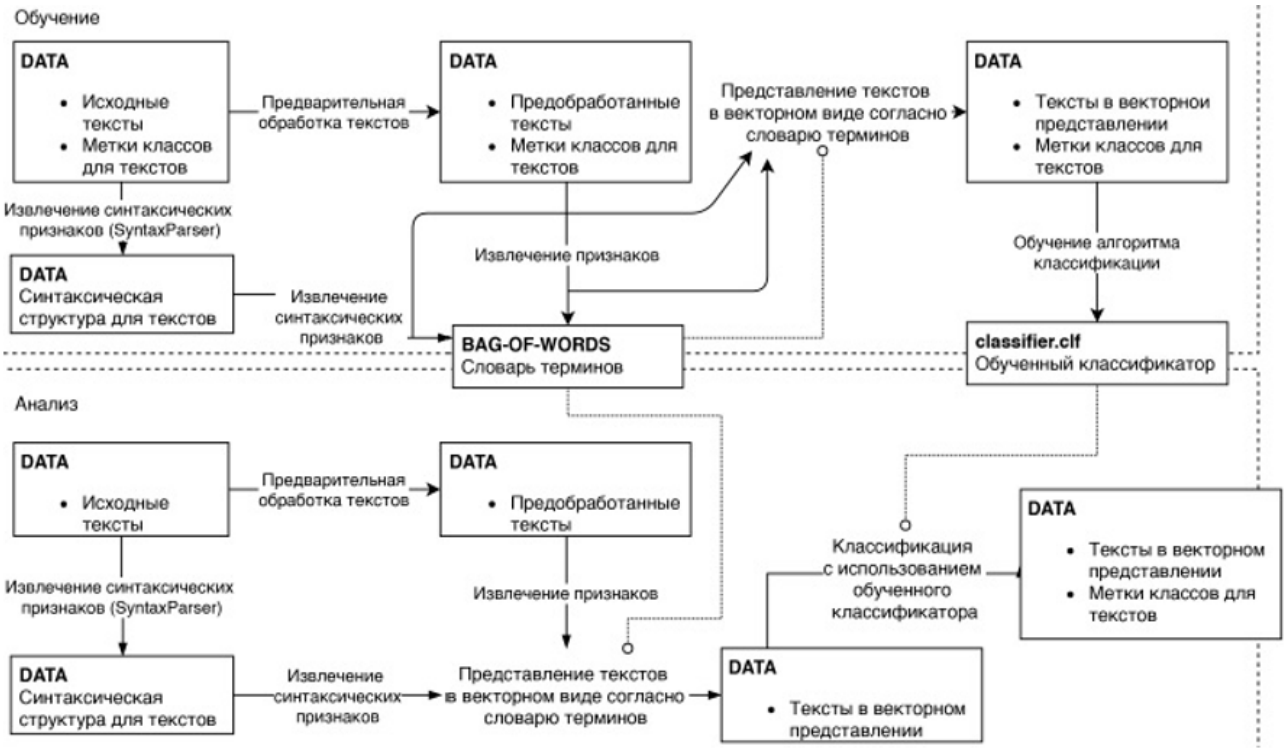

Рис. 1. Схема взаимодействия частей программного модуля анализа тональности

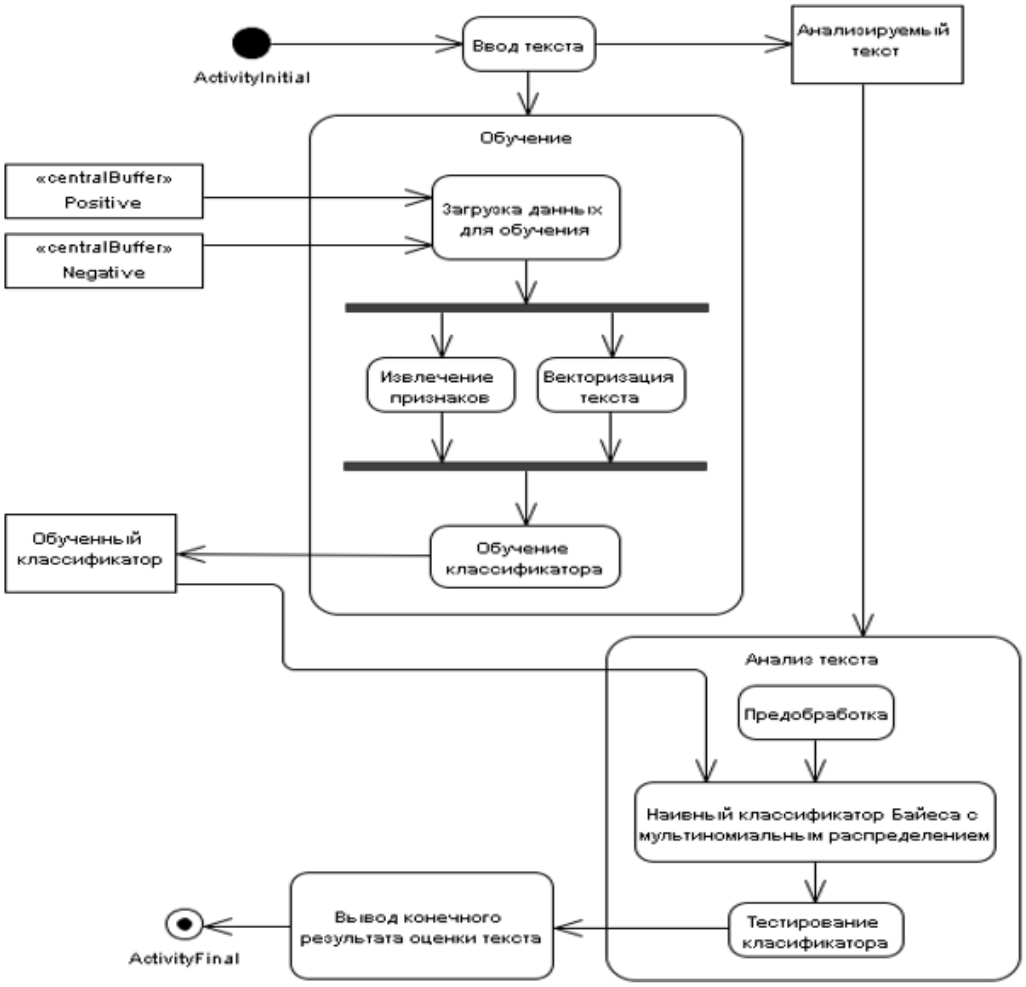

Рис. 2. Activity diagram

– считывание обучающих текстовых данных с соответствующими информационными метками об их тональности;

– извлечение признаков из обучающих данных;

– предварительная обработка текстовых данных;

– формирование словаря терминов и представление текстовых данных в векторной форме;

– обучение классификатора на основе преобразованных текстовых данных.

На втором шаге выполняется тестирование – анализ текстовых данных, для которых пользователю требуется получить оценку их эмоциональной окрашенности.

На рис. 2 представлена диаграмма деятельности, демонстрирующая последовательность действий при работе программного модуля анализа тональности текста.

6. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Тестирование качества классификации эмоциональной окрашенности текстов производилось с использованием оценок метрик эффективности, выражаемых в критериях: точность; полнота; F-мера. На основе данных понятий рассчитываются показатели точности и полноты по следующим формулам:

$$precision = \frac{AP}{AP + WP}, \qquad (20)$$

$$recall = \frac{AP}{AP + WN}, \qquad (21)$$

где $precision$ – является показателем точности; $recall$ – является показателем полноты, для классифицированных системой объектов; AP (Authentic Positive), ответы являющиеся подлинно-положительными, которые классифицируются системой и экспертом как положительные; WP (Wrong Negative), ответы являющиеся ошибочно-положительными, которые классифицируются системой как положительные, а экспертом как отрицательные; WN (Wrong Negative), ответы являющиеся ошибочно-отрицательными, которые классифицируются системой как отрицательные, а экспертом как положительные; AN (Authentic Negative), ответы являющиеся подлинно-отрицательными, которые классифицируются системой, а также экспертом как отрицательные.

Показатель точности позволяет оценить достоверность положительных ответов, которые были классифицированы системой. Показатель полноты позволяет понять то, что все ли достоверные были возвращены системой. Чем выше показатели точности и полноты, тем качественнее будет результат классификации.

F-мера – отношение показателей полноты и точности:

$$F = (\beta^2 + 1)\frac{precision \times recall}{\beta^2 precision + recall}, \qquad (22)$$

где β – является коэффициентом, отвечающим за вес, который отдаётся показателям точности и полноты. При $0 < \beta < 1$ приоритет будет отдаваться показателю точности, а при $\beta > 1$ приоритет будет отдаваться показателю полноты. При $\beta = 1$ показатели полноты и точности будут иметь одинаковые веса:

$$F = 2\frac{precision \times recall}{precision + recall}. \qquad (23)$$

Для получения данных оценок использовался метод эмпирического тестирования 10-fold cross validation, позволяющий оценить

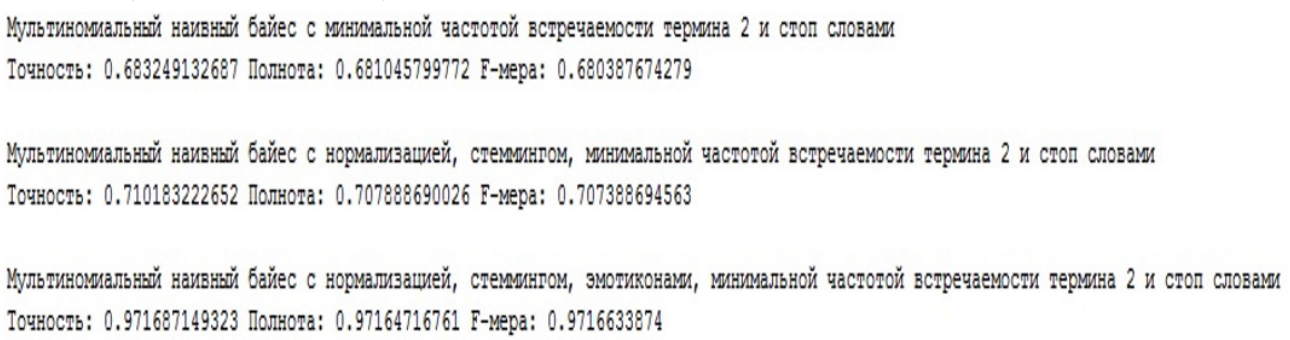

Рис. 4. Пример тестирования классификатора

Таблица 1

Наивный Байсоевский классификатор

Параметры эксперимента				10-fold cross validation		
Биграммы	Синтаксис	Обработка	Стемминг	Точность	Полнота	F-мера
–	–	–	–	84,05 %	78,45 %	81,15 %
–	–	–	+	89,50 %	87,99 %	89,23 %
–	–	+	–	90,69 %	84,19 %	87,32 %
–	–	+	+	92,66 %	**90,20 %**	**91,41 %**
–	+	–	–	86,15 %	78,71 %	82,26 %
–	+	–	+	91,58%	87,41 %	89,45 %
–	+	+	–	90,83 %	83,78 %	87,16 %
–	+	+	+	**93,29 %**	89,42%	91,30 %
+	–	–	–	84,27 %	78,42 %	81,24 %
+	–	–	+	90,66 %	86,65 %	88,61 %
+	–	+	–	90,62 %	83,73%	87,04 %
+	–	+	+	89,21 %	87,27 %	88,23 %
+	+	–	–	85,24 %	78,20 %	81,57 %
+	+	–	+	90,16 %	86,43 %	88,26 %
+	+	+	–	90,49%	86,09 %	86,63 %
+	+	+	+	93,05 %	88,36 %	90,65 %

Таблица 2

Машина опорных векторов SVM

Параметры эксперимента				10-fold cross validation		
Биграммы	Синтаксис	Обработка	Стемминг	Точность	Полнота	F-мера
–	–	–	–	74,48%	75,04%	74,76%
–	–	–	+	82,41%	84,31%	83,35%
–	–	+	–	79,75%	80,74%	80,24%
–	–	+	+	81,41%	81,86%	81,63%
–	+	–	–	75,18%	76,42%	75,79%
–	+	–	+	83,14%	85,18%	84,15%
–	+	+	–	80,12%	81,03%	80,57%
–	+	+	+	84,24%	87,36%	86,77%
+	–	–	–	73,85%	75,58%	74,71%
+	–	–	+	82,98%	83,95%	83,46%
+	–	+	–	79,67%	80,00%	80,83%
+	–	+	+	81,43%	79,78%	80,60%
+	+	–	–	75,18%	76,45%	75,81%
+	+	–	+	83,34%	84,56%	83,95%
+	+	+	–	79,90%	80,80%	80,35%
+	+	+	+	84,61%	86,67%	86,63%

Таблица 3

Метод К-ближайших соседей

Параметры эксперимента				10-fold cross validation		
Биграммы	Синтаксис	Обработка	Стемминг	Точность	Полнота	F-мера
–	–	–	–	55,99%	81,69%	66,48%
–	–	–	+	62,32%	77,89%	69,25%
–	–	+	–	62,84%	72,92%	67,51%
–	–	+	+	64,50%	78,92%	71,00%
–	+	–	–	55,47%	82,86%	66,49%
–	+	–	+	61,44%	77,20%	68,44%
–	+	+	–	62,08%	71,53%	66,48%
–	+	+	+	67,87%	75,58%	71,52%
+	–	–	–	57,97%	79,69%	67,14%
+	–	–	+	64,12%	77,89%	70,35%
+	–	+	–	65,53%	71,54%	68,41%
+	–	+	+	66,30%	79,20%	72,19%
+	+	–	–	56,48%	81,16%	66,64%
+	+	–	+	64,45%	74,40%	70,07%
+	+	+	–	64,80%	70,43%	67,50%
+	+	+	+	64,87%	73,53%	68,93%

Таблица 4

Результаты экспериментов

Алгоритм	Параметры		10-fold cross validation		
	Обработка	Стемминг	Точность, %	Полнота, %	F-мера, %
MNB	+	+	95,18	91,52	93,43
SVM	+	+	86,37	89,54	87,31
kNN	+	+	69,14	77,36	87,72

обобщающую способность алгоритмов, обучающихся по прецедентам и включающий в себя следующие этапы:

– разбиение обучающей выборки на k – количество частей;

– обучение алгоритма на тестовом подмножестве;

– тестирование качества на контрольном подмножестве;

– получение усредненных результатов показателей качества классификации.

В табл. 1, 2, 3, 4 приводятся результаты экспериментов по оценке качества работы классификатора при использовании различных параметров классификации.

Исходя из полученных результатов, наиболее высокие показатели по критериям точности, полноты и F-меры демонстрирует наивный Байесовский классификатор с мультиномиальной моделью распределения, используемый в качестве классификационного алгоритма для разработанного программного обеспечения.

Также по полученным результатам можно сделать вывод, что применение процедур по предварительной текстовой обработке в значительной мере улучшает результаты классификации.

СПИСОК ЛИТЕРАТУРЫ

1. *Сбоев А. Г.* Продвинутые нейросетевые модели для решения задачи определения тональности / А. Г. Сбоев, И. Е. Воронина, Д. В. Гудовских, А. А. Селиванов// Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2016. – № 4. – С. 178–183.

2. *Loukachevitch N.* SentiRuEval: testing object-oriented sentiment analysis systems in Russia / N. Loukachevitch, E. Kotelnikov , Y. Rubtsova // Proceedings of International Conference Dialog-2015, Moscow, Russia. – 2015. – 313 с.

3. *Рубцова Ю. В.* Построение корпуса текстов для настройки тонового классификатора / Ю. В. Рубцова // Программные продукты и системы. – Новосибирск: Научно-исследовательский институт «Центрпрограммсистем», 2015 – № 109 – С. 72–78.

4. *Котельников Е. В.* Автоматический анализ тональности текстов на основе методов машинного обучения / Е. В. Котельников, М. В. Клековкина // Компьютерная лингвистика и интеллектуальные технологии: по материалам ежегодной Международной конференции «Диалог». – М : РГГУ, 2012. – С. 15–21.

5. *Адаскина Ю. В.* Сентиментный анализ твитов на основе синтаксических связей / Ю. В. Адаскина, П. В. Паничева, А. М. Попов // Компьютерная лингвистика и интеллектуальные технологии: по материалам ежегодной Международной конференции «Диалог». – Москва : РГГУ, 2015. – С. 25–35.

Гаршина В. В. – канд. техн. наук, доцент, доцент кафедры Технологий обработки и защиты информации факультета компьютерных наук Воронежского государственного университета.
E-mail: garshina@cs.vsu.ru

Калабухов К. С. – магистр направления подготовки Информационные системы и технологии факультета компьютерных наук Воронежского государственного университета.
E-mail: kirill_ks@mail.ru

Степанцов В. А. – канд. техн. наук, доцент, доцент кафедры Технологий обработки и защиты информации факультета компьютерных наук Воронежского государственного университета.
E-mail: mrstep@yandex.ru

Смотров С. В. – магистр 2-года обучения направления Информационные системы и технологии факультета компьютерных наук Воронежского государственного университета.
E-mail: smotnashruss2009@mail.ru

Garshina V. V. – candidate of Technical Sciences, Associate professor, Department of Processing Technology and Information Security, Computer Sciences Faculty, Voronezh State University.
E-mail: garshina@cs.vsu.ru

Kalabukhov K. S. – Undergraduate, Department of Information systems, Computer Sciences Faculty, Voronezh State University.
E-mail: kirill_ks@mail.ru

Stepantsov V. A. – candidate of Technical Sciences, Associate professor, Department of Processing Technology and Information Security, Computer Sciences Faculty, Voronezh State University.
E-mail: mrstep@yandex.ru

Smotrov S. V. – Undergraduate, Department of Information systems, Computer Sciences Faculty, Voronezh State University.
E-mail: smotnashruss2009@mail.ru