Recognition of Sarcastic Sentences in the Task of Sentiment Analysis

Alexey Dolbin, Vladimir Rozaliev
Yulia Orlova, Sergey Fomenkov
Volgograd State Technical University
Volgograd, Russia
sizeof.void34@gmail.com, vladimir.rozaliev@gmail.com
yulia.orlova@gmail.com, saf@vstu.ru

***Abstract*—This article is devoted to the sarcasm recognition in the text written in a natural language. The main goal is to increase the accuracy of sentiment analysis. The sentiment level determination of a text that describes the appearance of a person was chosen as a domain area for the experiment. At first, references to the personality and elements that describes appearance from text are detected using the method of latent semantic analysis. The next step is to evaluate the attitude to a person in text using pre-labeled sentiment dictionary. At this stage, the method of recognising sarcastic sentences that contains a description of the appearance is used. The sentiment level should be re-evaluated in the person information model. The results of the experiment showed that the recognition of sarcasm based on the morphological features of words and the frequency characteristics of the sentences does not effectively increase the accuracy of sentiment level determination.**

***Keywords*—sentiment analysis, named entity recognition, text mining**

I. Introduction

Sentiment analysis of the text belongs to the category of information retrieval tasks. The importance of an effective solution to this problem grows over time, since the amount of information that needs to be processed by semantic analysis systems is continuously increasing. At the moment there are quite effective methods for sentiment analysis of the text, but there are a number of directions, the solution of which will make it possible to achieve greater accuracy of correct recognition. One such direction is the recognition of sarcasm. Sarcasm can be classified as an implicit approach to the expression of opposing emotions. However, even a person cannot always determine reliably whether this phrase is a sarcasm.

The task of automating the definition of sarcasm itself is of little practical value. Typically, you need a limited application area to apply the sentiment analysis. And most often the development is carried out in the following areas:

- Sentiment analysis of users reviews.
- Analysis of comments posted on social media resources [1].

The problem of recognizing sarcastic sentences in the text in natural language was considered in the context of searching for elements of a person's appearance and determining the sentiment class. This named entity was chosen not by chance, since it is a quite complex task to recognize it with high accuracy due to a large number of approaches to co-referencing through pronouns in the third person.

The aim of this work was to examine modern methods for determining the author's relationship to the described person by performing the sentiment analysis. The most obvious area of application of the development, considered in this article, is the analysis of comments on photos in social networks. Using the methods of machine learning, it is possible to construct a model that is able to recognize a positive or negative attitude to the appearance of the person depicted in the photograph. The main contribution of the authors of the article is the adaptation of existing methods of assessing the sentiment in the field of recognition of a person's appearance in the text in natural language [2].

II. Information Model

First of all, it was required to develop an information model of a person's appearance. This model must meet the following requirements:

- Extensibility.
- Visibility.
- Completeness of the description.

A frame presentation of knowledge is perfectly suitable for this description. Fig. 1 shows the final model of a person's appearance using the frame representation language notation. There are the main components on which it is possible to compose a complete description of a person's appearance in the frame slots. Slots for the model were compiled by the authors of the article. In this figure, "M" is the set of valid values for the description elements of the appearance for each slot. The specialty of the FRL notation is that it is permissible to join special procedures-demons to it. The only procedure is the determination of sentiment level with the subsequent resolution of sarcasm. It is worth noting that each non-empty slot must correspond to the sentences from which the facts were extracted for the frame. This is required

```
(frame Human_Appearance
(Height      ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Body        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Head        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Hair        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Face        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Forehead    ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Eyebrows    ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Eyes        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Eyelashes   ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Nose        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Lips        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Chin        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Teeth       ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Neck        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Shoulders   ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Chest       ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Back        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Legs        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
(Arms        ( value (M) ) ( IF_ADDED(sentiment_analysis) ) )
)
```

Figure 1. Simulation results for the network.

for further recognition of the presence of sarcasm in the text [3].

A person is one the most difficult named entity to recognize in the text. It s not so difficutl to determine person if text contains the name or surname of the entity. One common approach is to use contextual rules. Every rule represents a standard regular expression, which is constracted from the training sample as follows:

- Any mention of a person should be replaced with a special word PERSON.
- If the word in a training sample is an element of a human appearance, it should be set in its initial form.
- All other words should be replaced with its parts of speech.
- After processing the entire training sample, similar contextual rules should be combined using special characters. The '?' symbol means that this position can be omitted, the '+' symbol means that the position can be repeated one or more times in a row and the '|' symbol represents logical "or".
- Optionally, some phrases can be listed at the end of the training sample, which indicates that the sentence excludes the possibility of containing the entity.

Thus, the outputs are kind of regular expressions, which are applied to the text to define a named entity with it.

The next step is to resolve the reference of pronouns in the third person. Reference resolution of pronouns in the third form is one of the most common, but at the same time the simplest case. This task can be considered as a problem of binary classification. Therefore, it is a good opportunity to use support vector machines (SVM). The list of parameters for support vector machines which were used for training:

- Number of sentences between antecedent and anaphora.
- Whether the antecedent is in the nominative.
- Position of the anaphora in the sentence.
- Position of the antecedent in the sentence.
- Number of nouns and pronouns, which are located in sentences with antecedent and anaphora.
- Is the antecedent and anaphora case matches.
- Is the antecedent and anaphora genus matches.
- Is antecedent and anaphora both in a plural or singular form [3].

To fill the frame, a method of latent semantic analysis was used, or abbreviated LSA, as it has proved itself in the field of machine learning. Methods that do not use a pre-tagged training sample for the learning process show a smaller effectiveness in terms of recognition. The method of latent semantic analysis can be characterized as establishing the relationship between the vectors of the features of the analyzed documents to the words that serve as the keys. Thus, to use the method of semantic analysis of text in natural language, the slots of the frame should be used as search keys [3].

The latent-semantic algorithm is as follows:

- Create a list of all keywords that will be searched in the text.
- Create a frequency matrix *A*, in cells of which the count of how many times does this word occur in the text.
- Apply TF-IDF method on a frequency matrix to ensure that results are relevant [4].
- apply a singular matrix decomposition: algorithm divides the transformed frequency matrix *A* into three composite matrices *U*, *Vt* and *S* according to (1)

$$A = U \times S \times V^t \tag{1}$$

- Matrix *U* contains the coordinates of keywords and *Vt* – coordinates of documents.

Singular value decomposition of the matrix allows you to get rid of unnecessary noise, which significantly increases the efficiency of the method. The number of rows and columns that can be discarded before the subsequent analysis can be selected experimentally. It is now possible to obtain the nearest documents, which has the same semantic meaning as specified keyword, and then fill in the frame slots.

III. Vocabulary Based Sentiment Analysis

All approaches to determination of sentiment class are divided into three main groups:

- Compilation of a sentiment vocabulary.
- The use of various classifiers.
- The use of compiled contextual rules.

A rule-based approach shows the most accurate results, but it requires very high costs and colossal linguistic work for compilation. The main drawback of this approach is that it is extremely difficult to compose universal rules that are suitable for all domains. To achieve the most effective evaluation of the tonality, the rules are compiled for a specific application area.

In this experiment, an approach based on the valence dictionary was applied, since it shows a fairly high percentage of correct recognition. The task is greatly simplified if there is a source for compiling a dictionary of valences belonging to the domain under study. Such dictionary was compiled on the basis of the corpus of the Russian language OpenCorpora. all the phrases that are marked as "Qual" were chosen from this dictionary. Further, only those word forms that can be used for describing a person's appearance have been filtered out. To simplify the task of sentiment analysis, it was decided that the valences would correspond to available sentiment class. Table I shows an example of a sentiment dictionary.

Table I
PART OF COMPILED SENTIMENT DICTIONARY

Keyword	Valence
Friendly	2
Unfriendly	-2
Shy	0
Impartial	1
Mean	-1

For the study, the basic five sentiment classes were compiled:

- Negative.
- Strongly negative.
- Positive.
- Strongly positive.
- Neutral.

To determine the sentiment, the naive Bayes method was used. This method has proved itself in the field of machine learning. A naïve Bayesian algorithm is a classification algorithm based on the Bayes theorem with the assumption of independence of features. The classifier assumes that the presence of any feature in the class is not related to the presence of any other attribute. Let the *P(d|c)* be the probability of finding a document in all the documents of a given class. The basis of the naive Bayesian classifier is the corresponding theorem (2). In (2) *P(c)* is the probability of certain document can be found among all data set and *P(d)* – probability the document occurs throughout the whole corpus.

$$P(c|d) = \frac{P(d|c) \times P(c)}{P(d)} \quad (2)$$

Thus, the naive Bayes method is based on the problem of finding the maximum probability of a document belonging to a certain sentiment class. Thus, the sentiment level for each key element of a person's appearance can be determined by (3) [5].

$$P_{max} = \arg\max \left[P(c) \prod_{i=1}^{n} P(w_i|c)\right] \quad (3)$$

Classification using naïve Bayes is easy and fast and requires less training data. Also, it is better suited for classification based on categories (sentiment analysis with separate defined classes refers to such cases). However, if there is some value of a category characteristic in the data set that was not found in the training samples, then the model will assign a zero probability to this frame slot. Sentiment class for each key element of a person's appearance can be determined by (3), where *P(w|c)* is probability of occurrence of a certain term in a document.

Experimentally, it was found that the hierarchical classification gives better results than flat, because for each classifier, you can find a set of features that allows you to improve results. However, it requires a lot of time and effort for training and testing. Fig. 2 shows the final classifier based on the naive Bayes method.

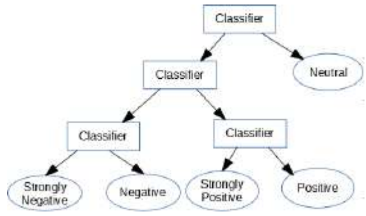

Figure 2. The Hierarchical Structure of Sentiment Classes.

IV. APPROACH TO SARCASTIC SENTENCES DETERMINATION

The issue of sarcasm recognition in a sentence requires the training of another classifier. To solve this issue, the method of k-nearest neighbors is used [6]. To classify each of the test sample objects, you must perform the following steps sequentially:

- Calculate the distance to each of the training sample objects.
- Choose *k* training sample objects, the distance to which is minimal.
- Class of the object being classified is the class most often encountered among the k nearest neighbors.

The following set of parameters for the vector of singularities was compiled:

- The presence of word forms, which are specific for sarcasm (such expressions include common words from the Internet slang).
- The presence of quotes in the text (if there are quotes, it is most likely that the text contains a certain degree of irony).
- High frequency of punctuation.
- The presence in the text of words that are most often used in conjunction with sarcasm for a particular language, which are taken from training samples [7].

For this case, the weight is given as a function of the distance to the nearest neighbors. In (4) *d(x, x(i))* is a function which determines the distance between elements in a vector space. Equation (5) finally determines whether or not the text being analyzed contains sarcasm, where *Zi* is a sum of weights for all of the available classes. If so, then the class of the slot must be changed to the opposite.

$$w(x(i)) = w(d(x, x_{(i)})) \quad (4)$$

$$C = \arg\max Z_i \quad (5)$$

Empirically, it was revealed that the classifier gives the best efficiency in terms of accuracy if it analyzes the nearest-neighbor number *K* equal to the number of sentiment classes.

To obtain more plausible results, you should filter out the most frequent words in the model. This step removes unnecessary noise that could affect the final result of the study. In addition, before using the K nearest neighbors method, the volume of aggregated sentiment information should be considered. In this study the results for unigrams and trigrams are provided [8].

V. Conducting The Experiment

To decide whether a sentiment recognition effectiveness is better or worse using the method of sarcasm determination, a numerical metric is needed. For most modern algorithms based on machine learning, metrics of accuracy and completeness of search are used. The accuracy of the search determines the proportion of documents that really belong to a given sentiment class across all documents of this class. The completeness of the search determines the ratio of the found classifiers of documents belonging to this class to all documents in the sample. Since in real practice of machine learning the maximum accuracy and completeness of search are unattainable simultaneously, the analysis of results using the F-measure will be the most acceptable. The F-measure is calculated using (6).

$$F = 2\frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

A training set consists of 500 samples was compiled: 150 of them were marked as "containing sarcasm" and 350 were marked as "not containing sarcasm". This ratio between classes was not chosen randomly, since the likelihood of evaluating the sentiment class of the text as positive or negative is much higher than sarcastic. The experiment was conducted on a sample of 100 texts, which are supposed to be a description to the different photos with no more than 200 words in length and contains only the information about person's appearance.

Table II
Experiment Results

	Recall	Precision	F
Unigrams, without sarcasm	0.80	0.82	0.810
Trigrams, without sarcasm	0.85	0.84	0.844
Unigrams, with sarcasm	0.86	0.68	0.760
Trigrams, with sarcasm	0.87	0.77	0.820

As can be seen from the obtained results (table II), the method of sarcasm recognition in the text slightly lowers accuracy due to a relatively large number of false positives. It can be concluded that lexical features and punctuation signs are not enough to train the classifier at a sufficient level. Most often, sentences have a complex structure, which cannot be treated as a "bag of words" and requires the use of contextual syntactic rules [9].

Conclusion

As a result of the experiment, it can be concluded that the resolution of the task of recognizing sarcasm in a text containing a description of the appearance of a person cannot be effectively resolved only using the methods of machine learning with supervision. As a further study, it requires the development of contextual rules based on the syntactic structure of the text. At this stage, the F-measure estimation showed that the method slightly reduces effectiveness due to a relatively large number of false positives. It may be worth considering a deeper approach to analyzing emotions, as suggested by the authors [10]. There are alternative approaches to solving the problem of determining the tonality class of the analyzed text. For example, the use of a neural network for text analysis can significantly expand the boundaries of tonality classes. This is achieved by applying a suitable output function, as a result of which the output is the probability with which a text fragment belongs to each class. This work was partially supported by RFBR (grants 17-07-01601, 17-29-07021, 18-07-00220, 18-47-343007, 18-47-342002, 19-07-00020).

References

[1] A. Ritter, S. Clark, Mausam and O. Etzioni, "Named entity recognition in tweets: an experimental study", Proceeding of the conference on empirical methods in natural language processing, Association for Computational Linguistics, pp. 1524–1534, July 2011.

[2] A. Dmitriev, A. Zaboleeva-Zotova, Y. Orlova and V. Rozaliev, "Automatic identification of time and space categories in the natural language text," Proceedings of the IADIS Internation Conference, pp. 23–25, October 2013.

[3] A. Dolbin, V. Rozaliev, Y. Orlova, "Recognition of a person named entity from the text written in a natural language," Journal of Physics: Conference Series, vol. 803, 5 p., 2017.

[4] B. Gebre, M Zampieri, P. Wittenburg and T. Heskes, "Improving native language identification with tf-idf weighting," 8th NAACL Workshop on Innovative Use of NLP for Building Educational Applications, pp. 216–223, 2013

[5] H. Shimodaira, "Text classification using naïve bayes," Learning and Data Note, vol. 7, pp. 1–9, 2014.

[6] D. Davidov, O. Tsur, A. Rappoport, "Semi-supervised recognition of sarcastic sentences in Twitter and Amazon," Computational Natural Language Learning, pp. 107–116, July 2010.

[7] P. Carvalho, L. Sarmento, M.J. Silva, E. Oliveira, "Clues for detecting irony in user-generated contents: oh...!! it's "so easy"," 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, pp. 53–56, November 2009.

[8] A. Reyes, P. Rosso, "Mining subjective knowledge from customer reviews: a specific case of irony detection," 2nd workshop on computational approaches to subjectivity and sentiment analysis, pp. 118–124, June 2011.

[9] R. Gonzalez-Ibanez, S. Muresan, N. Wacholder, "Identifying sarcasm in Twitter: a closer look," 49th Annual Meeting of the Association for Computational Linguistics, vol. 2, pp. 581–586, June 2011.

[10] L. Volkova, A. Kotov, E. Klyshinsky, N. Arinkin, "A Robot Commenting Texts in an Emotional Way," Communications in Computer and Information Science, vol. 754, pp. 256–266, August 2017.

РАСПОЗНАВАНИЕ ПРЕДЛОЖЕНИЙ СОДЕРЖАЩИХ САРКАЗМ В ЗАДАЧЕ АНАЛИЗА ТОНАЛЬНОСТИ

Долбин А.В., Розалиев В.Л.,
Орлова Ю.А., Фоменков С.А.

Данная статья посвящена распознаванию сарказма в тексте, написанном на естественном языке. Основная цель - повысить точность анализа тональности текстов. Определение уровня тональности текста, описывающего внешность человека, было выбрано в качестве предметной области для эксперимента. На первом этапе в тексте распознаются личности и элементы описания их внешнего вида при помощи метода латентно-семантического анализа. На следующем этапе определяется отношение к внешнему виду человека с использованием размеченного словаря валентности. На данном этапе используется метод распознавания саркастических предложений, которые содержат описание внешнего вида личностей. В результате чего уровень тональности переоценивается в информационной модели внешнего вида человека. Результаты эксперимента показали, что распознавание сарказма на основе морфологических особенностей слов и частотных характеристик предложений не позволяет эффективно повысить точность определения уровня тональности.

Received 10.01.19