Sentiment Classification Of Long Newspaper Articles Based On Automatically Generated Thesaurus With Various Semantic Relationships

Ivan Shchitov, Ksenia Lagutina, Nadezhda Lagutina, Ilya Paramonov

P.G. Demidov Yaroslavl State University

Yaroslavl, Russia

ivan.shchitov@e-werest.org, lagutinakv@mail.ru, lagutinans@gmail.com, Ilya.Paramonov@fruct.org

Abstract—The paper describes a new approach for sentiment classification of long texts from newspapers using an automatically generated thesaurus. An important part of the proposed approach is specialized thesaurus creation and computation of term's sentiment polarities based on relationships between terms. The approach's efficiency has been proved on a corpus of articles about American immigrants. The experiments showed that the automatically created thesaurus provides better classification quality than manual ones, and generally for this task our approach outperforms existing ones.

I. Introduction

Automatic sentiment analysis of texts is widely used in such subject areas as marketing and advertising, psychology, economics, political science, and many others [1]. The results of such analysis are useful in marketing and sociological research, building human-machine interface systems, forecasting events in politics and economics, and so on.

Most works on sentiment analysis process short reviews, news, or posts from social media [2]. Such documents usually represent short subjective opinions of users. Another type of texts with opinions includes long newspaper or analytical articles whose length is more than 500 words. Such articles contain more analytical information than short ones and have different structure, therefore their analysis requires specialized methods.

However, sentiment classification of large texts is insufficiently researched in scientific literature. Most investigations show relatively poor results and remark that the solution of this task requires new approaches and models of the subject area [3].

One of the way to model a subject area is to build a specialized thesaurus. Particularly, such a thesaurus in application to the sentiment analysis task can contain different relationships as semantic links between terms and term weights as marks of sentiment polarity. There is some evidence of the effectiveness of these lexical resources in opinion analysis [4].

The goal of our research is to improve performance of methods that classify long newspaper articles by sentiment polarity, taking into account thesaurus relationships. We propose a new approach that includes hybrid methods that automatically create a sentiment thesaurus using manually classified text corpus, and standard algorithms that automatically classify articles into two groups: positive and negative.

The rest of the paper is structured as follows. In Section II we overview investigations related to sentiment extraction and classification. Section III represents details of the proposed approach. In Section IV we describe the corpus of newspaper articles we use in our experiments and metrics for result estimation. Section V provides experiment results, their justification, and explanation. In conclusion we summarize the main results of the paper and propose directions for future work.

II. Related work

Classification of political texts is one of the most popular task in sentiment analysis of long newspaper articles. Grimmer and Stewart [3] discuss advantages and disadvantages of dictionary and machine learning methods for automatic solution of this problem. Dictionary methods strongly depend on lexical resources that are too specific and not validated properly. Supervised learning methods are more flexible, but they do not provide high quality: the authors' experiment with Russian public statements and Random Forest algorithm showed accuracy of 0.65.

Better results are achieved by Kaur and Chopra [5]. They classify Indian online news articles to positive, neutral, and negative using a hybrid approach that combines standard supervised algorithms Naive Bayes and Decision Table. The obtained accuracy equals 0.71.

Padmaja at al. [6] solve the close task. They detect only negative sentences in political newspaper articles. Authors apply three metods to this problem that mark words with polarity weights using SentiWordNet. Then these methods change weights depending on word context in three different ways. The final weights are taken as an input for supervised classifiers. The best quality is provided by the method that build dependency trees for words in sentences and apply these relationships to polarity computation. Its precision, recall, and F-measure are 0.79, 0.77, and 0.78 correspondingly that outperforms other approaches by 5–20 %. This research shows that subject area modeling and use of lexical resources can be very effective for sentiment analysis.

SentiWordNet is a popular thesaurus for solution of sentiment analysis tasks. The idea to supplement it with term weights that depend on subject area is successfully used in the research [7]. Authors mark objective words from SentiWordNet as positive or negative if they appear in positive or negative

sentences. Classification is performed on reviews dataset that has a large appraisal vocabulary, therefore the method show high quality with accuracy of 0.74–0.79.

Another approach to apply a lexical resource to sentiment analysis is to build a thesaurus based on a text corpus that is analysed. Bollegala at al. [8] extract opinion words from a dataset where text sentiments are known, and build one-way associations between them using information about word's cooccurrences and text sentiments. Besides, these associations have weights that become word feature for the binary classifier Maximum Entropy. The best accuracy of this method is about 0.85.

Almatarneh and Gamallo [9] underline that the best natural language resources for sentiment classification depend on the subject area and the most convenient and fast way to build it is automatical generation without expert help. Authors classify short reviews into positive and negative using a dictionary created on a specific dataset. The dictionary contains term polarities calculated on the basis of term frequency and review ratings. In result the F-measure equals 0.76–0.83.

Several approaches for term polarity computation take into account not only the term context but also semantic relationships between terms. Particularly, Kamps at al. [10] calculate distances in WordNet between terms with known polarity and terms without it. Each distance for a term equals the length of the shortest synonym chain between the term and the term with univocal polarity like "good" or "bad". The term weight is calculated as normalized difference between distances to positive and negative words. Authors compare results with several manually constructed dictionaries and get accuracy about 0.61–0.71.

Loukachevitch and Chetviorkin [11] combine two methods to compute a polarity weight. Firstly, authors use a supervised classifier with TF*IDF that find probabilities of term sentiments. Then, they assign the average of related term's weights for each term as the sentiment polarity. Comparison with ProductSentiRus+ lexicon shows F-measure of 0.69.

It should be mentioned that most of sentiment computation methods process different thesaurus relationships in the same way or use only one type of relationships. Nevertheless, relationships have different semantics, therefore it is reasonable to take them into account separately.

III. PROPOSED APPROACH FOR SENTIMENT CLASSIFICATION

A. Overview

In this paper we propose an approach to sentiment classification that is based on use of the sentiment thesaurus with various relationships. The approach includes two steps: thesaurus generation and text classification. On the first step we create a sentiment thesaurus fully automatically. On the second step we classify texts using standard algorithms SVM and Naive Bayes with word features that depend on terms' weights in the thesaurus. Both steps take as input a corpus of raw texts. We divide them into training and test sets and use this division at each step of the approach application. Texts from the training set should be initially marked as positive or negative by an expert.

The generated after the first step sentiment thesaurus contains a set of terms from a concrete subject area, semantic relationships between them, and weights for terms that marks sentiment polarity. The values from −1 to 0 mean negative terms, values from 0 to 1—positive ones.

B. Automatic thesaurus generation

To generate the sentiment thesaurus firstly we select words from a corpus that potentially have sentiment polarity: nouns, adjectives, verbs, and adverbs. All extracted terms become thesaurus terms.

Secondly, relationships between terms are constructed. For this step we apply our method from the previous work [12] that extracts several types of semantic relationships between terms. This method provides a highly connected thesaurus with a lot of associations, synonyms, and hyponym—hypernyms. Therefore, when we calculate term weights based on its relationships, the weight depends on multiple different term polarities that could probably better characterize the term.

Thirdly, terms' weights are calculated. We do it into two steps. On the first one we assign terms with weights depending on their appearance in the training sets. For this set we already know text sentiments and reflect this information in the following way.

We suppose that positive terms appears in same texts more often than in negative and the tendency for negative terms is similar. Based on this assumption, we calculate how many times a term appears in positive (p) and negative (n) texts in the training set. The term weight w is computed using the formula: $w = (p - n)/(p + n)$. So the weight is in the range from −1.0 ("absolutely negative" terms) to 1.0 ("absolutely positive" terms).

After this step the thesaurus still has terms without weights that appears only in test set texts, but such terms have relationships with weighted ones. We assume that if a term has sentiment polarity, its related terms in the thesaurus have the same polarity, possibly stronger or weaker. Taking into account this consideration, we compute remaining weights using thesaurus relationships, and each relationship has its own coefficient that reflects its semantics. We find a related term with a weight for the term in question, multiply this weight by the relationship coefficient, and assign the resulted weight to the term. In the case of existing several related terms with weights, we assign the average of synonyms weights multiplied by the relationship coefficient. If there are no synonyms, the same procedure is performed for hypernyms. When the term has neither synonyms, nor hypernyms, its weight depends on associations in the same way. In the end, all terms have weights and the thesaurus is constructed.

The following example illustrates weight computation for term with several different relationships. If the term "ground" has two synonyms and one hypernym (Fig. 1), its weight depends only on synonyms. We find their weights' average and multiply it by 0.5—the relationship coefficient: $w = (0.33 + 0.15)/2 \cdot 0.5 = 0.12$. So, the weight for the term "ground" equals 0.12.

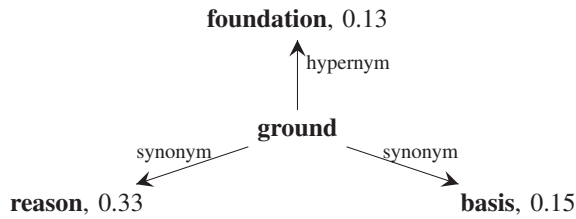Another example of weight computation is shown in Fig. 2, where a term has neither synonyms nor hypernyms. The

foundation, 0.13

↑ hypernym

ground

synonym ↙   synonym ↘

reason, 0.33   **basis**, 0.15

Fig. 1.   A thesaurus term with weighted synonyms and a hypernym

allegedly, −0.05   **October**, −0.01

↖ association   ↗ association

criminal charge

↙ association   ↘ association

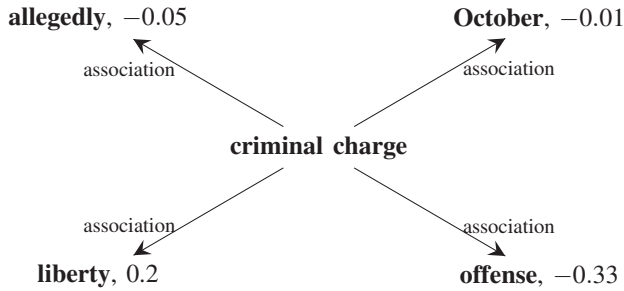**liberty**, 0.2   **offense**, −0.33

Fig. 2.   A thesaurus term with weighted associations

term "criminal charge" is linked by associative relationships with four terms: "liberty" whose weight equals 0.2, "offense" (−0.33), "allegedly" (−0.05) and "October" (−0.01). To calculate the weight of "criminal charge" we use the same rule and take into account only one type of relationships, associations. The term weights' average multiplied by 0.1 (he relationship coefficient) equals −0.00475 is the term weight for "criminal charge".

Relationship coefficients are assigned from following considerations. Synonyms and hypernyms relationships are contextual, so their semantic is a little bit different and their thesaurus weights should also be different. Associations' semantic differs much more and we reflect it in weights too. Based on this considerations, we choose the synonym's and hypernym's coefficients of 0.5. For associations we chose significantly smaller coefficient 0.1.

C. Calculation of feature vectors

After thesaurus generation we calculate feature vectors for each text using computed sentiment polarities.

The length of the vector equals the number of thesaurus terms. Each feature corresponds with a particular thesaurus term and is calculated as $w \cdot F$, where w is the term weight from the thesaurus and F is a some characteristic of the term depending on the text or thesaurus. Namely, in our research we consider the following characteristics: TF*IDF, index Gini, info gain, mutual information, and chi-square statistic [13]. For the sake of completeness we collect the formulas for their calculation here.

The formula for TF*IDF is

$$TF*IDF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \cdot \log \frac{|D|}{|d : t_i \in d|},$$

where i is the term index and j is the text index in the matrix, $n_{i,j}$ is how many times the term t_i appears in the text d_j, and D is the set of texts.

The formula for index Gini is

$$G(t) = \sum_{i=1}^{2} p_i^2(t),$$

where $p_i(t)$ is the probability that the term t belongs to the class i. In our research there are two classes: positive and negative and their probabilities for each term are reflected as thesaurus term weights, therefore $p_0(t) = w$ and $p_1(t) = 1 - w$.

The formula for info gain is

$$I(t,d) = -\sum_{i=1}^{2} P_i(d) \cdot \log(P_i(d)) + F(t) \cdot \sum_{i=1}^{2} p_i(t) \cdot \log(p_i(t)) +$$
$$+ (1 - F(t)) \cdot \sum_{i=1}^{2} (1 - p_i(t)) \cdot \log(1 - p_i(t)),$$

where $p_i(t)$ is the same as in the previous formula. If $p_i(t)$ is negative, we take $-\log|p_i(t)|$ instead of $\log(p_i(t))$. P_i is the global probability of the text d being in the class i and equals the fraction of positive or negative terms in the d. $F(t)$ is the number of texts with the term t.

The formula for mutual information is

$$M(t) = \sum_{i=1}^{2} P_i(d) \cdot \log \frac{p_i(t)}{P_i(d)},$$

where all elements are computed as for info gain.

The formula for chi-square statistic is

$$\chi^2(t) = \sum_{i=1}^{2} P_i(d) \cdot \frac{|D| \cdot F^2(t) \cdot (p_i(t) - P_i(d))^2}{F(t) \cdot (1 - F(t)) \cdot P_i(d) \cdot (1 - P_i(d))},$$

where all elements are computed as for previous formulas.

D. Text classification

After feature computation we classify vectors into positive and negative classes. For this task we apply standard classification algorithms that process the term-to-document matrix. Previous research in sentiment analysis shows that the best classifiers are machine learning methods [3]. Thus, we chose two most common supervised algorithms SVM and Naive Bayes that are state-of-the-art in sentiment classification [2].

IV.   EVALUATION PROCEDURE AND USED TEXT CORPUS

We classified the corpus of articles about American immigrants that were chosen from The New York Times, The New York Post, and The Los Angeles Times newspapers. In total the corpus has 56 articles that contain 37 669 words and 234 310 characters, i.e. in average 673 words and 4 184 characters per text. The texts were initially marked as positive or negative by an experts in social sciences. We divided them into training and test sets, each of them contained 17 positive articles and 11 negative ones.

To compare the classification quality we used one automatically generated thesaurus based on the chosen corpus and two well-known thesauri created manually: SentiWordNet (http://sentiwordnet.isti.cnr.it) and SenticNet (http://sentic.net). These thesauri do not have relations between terms, they
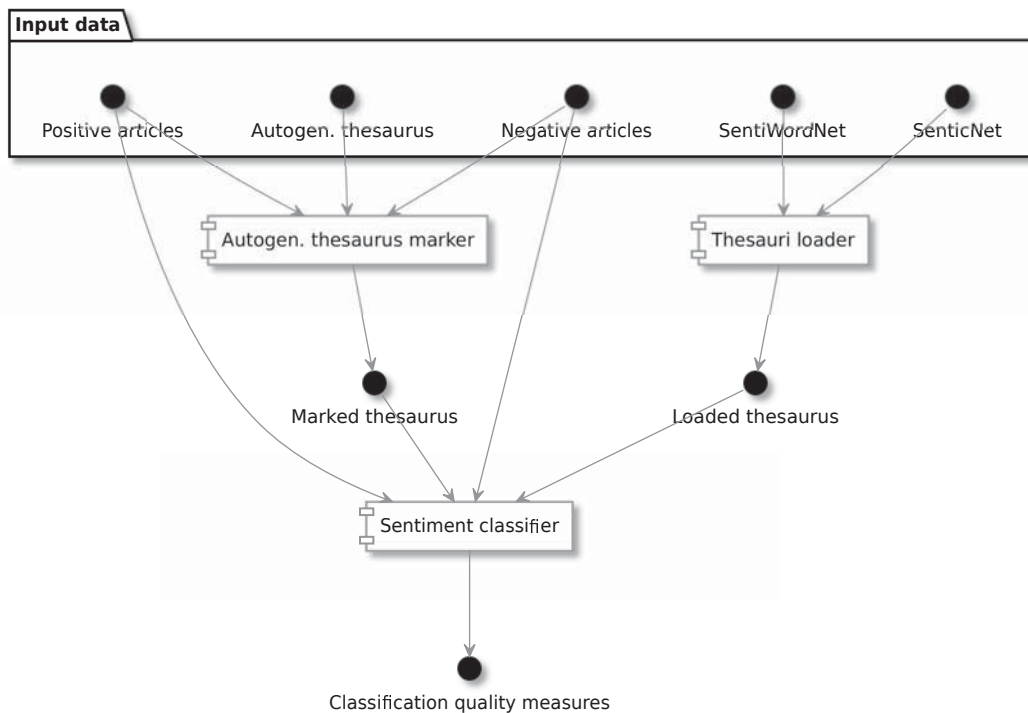
Fig. 3.   Classification procedure pipeline

contain only weights expressing terms' polarity. In turn, the automatically generated thesaurus contains terms, term weights, and term relations: synonyms, hyponyms, hypernyms, and associations.

The tool for the sentiment classification was implemented by the authors in the Python programming language using the NLTK suite of libraries (http://www.nltk.org). NLTK implements the well-known Naive Bayes and SVM classifiers that was chosen for experiments.

The implemented sentiment classification tool contains three modules: manually created thesauri loader, automatically generated thesaurus marker and sentiment classifier. The first module contains functions for reading the SentiWordNet and SenticNet thesauri and saving their terms and term weights into data structures for further calculations. The marker takes the automatically generated thesaurus with positive and negative articles, calculates the term weights, and saves the thesaurus with weigths (marked thesaurus) for further using in the sentiment classifier. The sentiment classifier implements the feature vectors calculation for each text from the corpus using weights, classifies the texts and calculates the classification quality measures. The whole classification pipeline is shown in Fig. 3.

To evaluate the classification results we use the most popular quality measures: precision, recall, F-measure, and accuracy. The precision is the fraction of documents actually belonging to the given class among all documents that the algorithm assigned to the class. The recall is the fraction of documents found by the algorithm that belong to the given class among all documents of the class. The F-measure is the harmonic mean of the precision and recall. The accuracy is

the fraction of the retrieved documents for which the classifier made a correct decision.

V.   RESULTS

We conducted several experiments with two manually and one automatically generated thesauri. We used them with four term measures and two classifiers described in Section III. Also their results are compared with cases without thesauri. Computation of index Gini, info gain, mutual information, and χ^2 depend on a thesaurus as it is described in Subsection III-C, thus we do not calculate them for experiments without thesauri.

Table I shows results of classification without thesauri. P, R, and F mean precision, recall, and F-measure for positive (*pos*) or negative (*neg*) texts respectively. From the table we can see that standard algorithms marks almost all texts as positive, so most popular word features tf and TF*IDF do not fit to long political texts. Manual analysis of feature vectors shows that they contains too many zeros and due to this fact cannot be classified properly.

Tables II, III, and IV display results for the proposed approach that was conducted with three different thesauri: SenticNet, SentiWordNet, and automatically created one. The best results for each thesaurus are marked in bold.

The experiments confirm that TF*IDF is not suitable for our task. The highest quality is achieved by the automatic thesaurus with SVM and info gain or with Naive Bayes and mutual information. These combinations significantly outperform the others by almost all metrics. The accuracy of the proposed method is better by 10–15 % (0.75 in comparison with 0.643 and 0.607). The F-measure for positive texts (0.82)

TABLE I.   CLASSIFICATION QUALITY OF ALGORITHMS WITHOUT THESAURI

Classifier	Term characteristic	Accuracy	P_{neg}	R_{neg}	F_{neg}	P_{pos}	R_{pos}	F_{pos}
SVM	TF	**0.643**	**1.000**	0.091	0.167	0.630	**1.000**	**0.773**
Naive Bayes	TF	0.607	0.500	**0.273**	**0.353**	**0.636**	0.824	0.718
SVM	TF*IDF	0.571	0.000	0.000	0.000	0.59	0.941	0.727
Naive Bayes	TF*IDF	0.571	0.000	0.000	0.000	0.59	0.941	0.727

TABLE II.   CLASSIFICATION QUALITY OF ALGORITHMS WITH THE AUTOGENERATED THESAURUS

Classifier	Term characteristic	Accuracy	P_{neg}	R_{neg}	F_{neg}	P_{pos}	R_{pos}	F_{pos}
SVM	TF*IDF	0.607	0.000	0.000	0.000	0.607	**1.000**	0.756
Naive Bayes	TF*IDF	0.607	0.000	0.000	0.000	0.607	**1.000**	0.756
SVM	index Gini	0.679	0.667	**0.364**	0.471	0.682	0.882	0.769
Naive Bayes	index Gini	0.643	0.600	0.273	0.375	0.652	0.882	0.750
SVM	info gain	**0.750**	**0.833**	**0.455**	**0.588**	**0.727**	**0.941**	**0.820**
Naive Bayes	info gain	0.643	0.600	0.273	0.375	0.652	0.882	0.75
SVM	mutual information	0.607	0.500	0.182	0.267	0.625	0.882	0.732
Naive Bayes	mutual information	**0.714**	**0.800**	**0.364**	**0.500**	**0.696**	**0.941**	**0.800**
SVM	χ^2	0.643	0.571	**0.364**	0.444	0.667	0.824	0.737
Naive Bayes	χ^2	0.607	0.000	0.000	0.000	0.607	**1.000**	0.756

TABLE III.   CLASSIFICATION QUALITY OF ALGORITHMS WITH SENTICNET

Classifier	Term characteristic	Accuracy	P_{neg}	R_{neg}	F_{neg}	P_{pos}	R_{pos}	F_{pos}
SVM	TF*IDF	**0.607**	0.000	0.000	0.000	0.607	**1.000**	**0.756**
Naive Bayes	TF*IDF	**0.607**	0.000	0.000	0.000	0.607	**1.000**	**0.756**
SVM	index Gini	0.464	0.300	0.278	0.286	0.556	0.588	0.571
Naive Bayes	index Gini	0.429	0.308	0.364	0.333	0.533	0.471	0.500
SVM	info gain	0.393	0.286	0.364	0.320	0.500	0.412	0.452
Naive Bayes	info gain	0.500	0.400	0.545	0.462	**0.615**	0.471	0.533
SVM	mutual information	0.464	0.250	0.182	0.211	0.55	0.647	0.595
Naive Bayes	mutual information	0.429	0.308	0.364	0.333	0.533	0.471	0.500
SVM	χ^2	0.393	0.200	0.182	0.190	0.500	0.529	0.514
Naive Bayes	χ^2	0.393	**0.393**	**1.000**	**0.564**	0.000	0.000	0.000

is also the largest. Precision and recall for positive text are one of the best: recall equal to 1.0 is achieved only when a classifier selects very small number of negative texts.

Manual thesauri provide in some cases better P_{neg}, R_{neg}, and F_{neg} than the automatic one. Although these metrics are higher, P_{pos} and R_{pos} are worse. For example, the combination of SentiWordNet, SVM, and χ^2 provides recall of 0.909 and F-measure of 0.606 for negative articles, but the same metrics for positive are 0.294 and 0.435, which are too small.

The use of existing sentiment thesauri is less effective than the automatic one in all cases. The best F_{pos} for SentiWordNet is 0.773, and it is worse by 5 % in comparison with 0.82 for the proposed approach. Besides, accuracy of 0.643 for SentiWordNet, Naive Bayes, and index Gini is the largest between all combination for manual thesauri, and it is lower by 5–9 % than in cases with our thesaurus. Combinations with SenticNet shows much worse results than even with SentiWordNet: the best accuracy, F_{pos}, and F_{neg} are 0.607, 0.756, and 0.564 correspondingly.

Besides, if we compare our results with measure values from other research described in Section II, we can see that accuracy of 0.75 achieved by the proposed approach is better than 0.71 from [5], where authors also classify newspaper

articles. Among all related works the best accuracy is 0.85 (see [8]), but it is achieved for short movie reviews, not long articles. Our best F-measure 0.82 is also fits state-of-the-art in sentiment analysis—in other works [6], [9] it is about 0.78–0.83.

Summarily, the proposed approach with the automatic thesaurus, SVM, and info gain characteristic significantly increases classification quality. Most positive texts are extracted properly (recall is 0.941), about a half of negative ones also are on the right place (recall is 0.455), and number of errors is the lowest (accuracy is 0.75).

From the experiment results we conclude that classification quality increases because of the use of automatically generated thesaurus containing different semantic relationships that we use to determine term sentiments. It is the main advantage of our sentiment thesaurus, because such lexical resources as SenticNet and SentiWordNet do not have any relationships between terms.

Also, most popular natural language resources are general purpose thesauri, so their set of terms is limited by common words and does not include a specific lexicon. Unlike them, the proposed thesauri has specialized vocabulary and relationships. Besides, this thesaurus is a good model of the subject area

TABLE IV. CLASSIFICATION QUALITY OF ALGORITHMS WITH SENTIWORDNET

Classifier	Term characteristic	Accuracy	P_{neg}	R_{neg}	F_{neg}	P_{pos}	R_{pos}	F_{pos}
SVM	TF*IDF	0.571	0.000	0.000	0.000	0.593	0.941	0.727
Naive Bayes	TF*IDF	0.571	0.000	0.000	0.000	0.593	0.941	0.727
SVM	index Gini	0.464	0.300	0.278	0.286	0.556	0.588	0.571
Naive Bayes	index Gini	**0.643**	**1.000**	0.09	0.167	0.630	**1.000**	**0.773**
SVM	info gain	0.429	0.308	0.364	0.333	0.533	0.471	0.500
Naive Bayes	info gain	0.214	0.077	0.091	0.083	0.333	0.294	0.313
SVM	mutual information	0.536	0.333	0.182	0.235	0.591	0.765	0.667
Naive Bayes	mutual information	0.500	0.286	0.182	0.222	0.571	0.706	0.632
SVM	χ^2	0.536	0.455	**0.909**	**0.606**	**0.833**	0.294	0.435
Naive Bayes	χ^2	0.427	0.400	**0.909**	0.556	0.667	0.118	0.200

that takes into account area's particular qualities, because it is constructed on the concrete text corpus and, therefore, reflects its structure, semantic meaning of terms and relationships, and sentiment polarities.

Another advantage of our approach is that it allows to extract a large number of terms and relationships between them and spread known sentiment polarity of one terms to others using all types of relationships. Such approach increases probability that the term has proper sentiment in the case when we do not initially know its polarity, because we determine term sentiment using lots of relationships. This idea is confirmed by experiments, when we increased sentiment classification accuracy by 10 % and F-measure by 5 % applying the proposed approach.

VI. CONCLUSION

In this paper we proposed an approach for sentiment classification of long newspaper articles based on specialized thesaurus construction and use. The approach consist of two stages: automatic creation of a sentiment thesaurus using the concrete text corpora and text classification using information from the thesaurus.

After experiments with our approach and different thesauri, term features and classifiers we found out that the best results are achieved when we use the automatical thesaurus, info gain characteristic, and SVM classifier. This combination of parameters significantly outperforms all cases with manually constructed thesauri or without thesauri. Although our results are lower than the highest ones for sentiment classification, the best results of existing algorithms are achieved for short texts. In classification of long articles quality provided by the proposed approach, is state-of-the-art.

High quality of the results allows to assume that automatic generation and use of specialized thesauri is a prominent approach in sentiment analysis. The further investigation can concern its application to close tasks like opinion extraction, subjectivity classification, and so on.

ACKNOWLEDGMENT

REFERENCES

[1] S. K. Yadav, "Sentiment analysis and classification: a survey," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 3, no. 3, pp. 113–121, 2015.

[2] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14–46, 2015.

[3] J. Grimmer and B. M. Stewart, "Text as data: The promise and pitfalls of automatic content analysis methods for political texts," *Political analysis*, vol. 21, no. 3, pp. 267–297, 2013.

[4] N. V. Lukashevich and I. I. Chetviorkin, "Combining corpus and thesaurus information for extracting sentiment words," *Sistemy i Sredstva Informatiki [Systems and Means of Informatics]*, vol. 25, no. 1, pp. 20–33, 2015.

[5] H. Kaur and V. Chopra, "Design and implementation of hybrid classification algorithm for sentiment analysis on newspaper articles," *Proceedings of International Conference on Information Technology and Computer Science*, pp. 57–62, 2015.

[6] S. Padmaja, S. S. Fatima, and S. Bandu, "Evaluating sentiment analysis methods and identifying scope of negation in newspaper articles," *International Journal of Advanced Research in Artificial Intelligence*, vol. 3, no. 11, pp. 1–6, 2014.

[7] C. Hung and H.-K. Lin, "Using objective words in SentiWordNet to improve sentiment classification for word of mouth," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 47–54, 2013.

[8] D. Bollegala, D. Weir, and J. Carroll, "Cross-domain sentiment classification using a sentiment sensitive thesaurus," *IEEE transactions on knowledge and data engineering*, vol. 25, no. 8, pp. 1719–1731, 2013.

[9] S. Almatarneh and P. Gamallo, "Automatic construction of domain-specific sentiment lexicons for polarity classification," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2017, pp. 175–182.

[10] J. Kamps, M. Marx, R. J. Mokken, M. De Rijke *et al.*, "Using WordNet to measure semantic orientations of adjectives." in *LREC*, vol. 4. Citeseer, 2004, pp. 1115–1118.

[11] N. Loukachevitch and I. Chetviorkin, "Refinement of Russian sentiment lexicons using RuThes thesaurus," in *Proceedings of the 16th All-Russian Conference on Digital Libraries: Advanced Methods and Technologies, Digital Collections*, 2014, pp. 61–65.

[12] K. Lagutina, E. Mamedov, N. Lagutina, I. Paramonov, and I. Shchitov, "Analysis of relation extraction methods for automatic generation of specialized thesauri: Prospect of hybrid methods," in *Proceedings of the 19th Conference of Open Innovations Association FRUCT. Jyvaskyla, Finland, 7-11 November 2016*, 2016, pp. 138–144.

[13] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining text data*. Springer, 2012, pp. 163–222.