Sentiment Classification of Russian Texts Using Automatically Generated Thesaurus

Ksenia Lagutina, Vladislav Larionov, Vladislav Petryakov, Nadezhda Lagutina, Ilya Paramonov
P.G. Demidov Yaroslavl State University
Yaroslavl, Russia
lagutinakv@mail.ru, vladlarionov998@gmail.com, petryakov.v@inbox.ru, lagutinans@gmail.com, Ilya.Paramonov@fruct.org

Abstract—**This paper is devoted to an approach for sentiment classification of Russian texts applying an automatic thesaurus of the subject area. This approach consists of a standard machine learning classifier and a procedure embedded into it, that uses thesaurus relationships for better sentiment analysis. The thesaurus is generated fully automatically and does not require expert's involvement into classification process. Experiments conducted with the approach and four Russian-language text corpora, show effectiveness of thesaurus application to sentiment classification.**

I. Introduction

Sentiment analysis is the task of detecting, extracting, and classifying opinions, sentiments, and attitudes concerning different topics, as expressed in texts [1]. Various solutions of these tasks can be used for e-commerce sites, news reports, blog forums, and social networks that users investigate to understand the public opinion and consumers preferences, political movements, social gathering events, product preferences, marketing campaigns, reputation monitoring, and so on [2].

Most authors, who investigate sentiment analysis problems, work with texts in English [3]. That is why, most of methods for sentiment analysis are created and tested only for one language, and they can work improperly for multilingual online content, as it is described in the research [4]. In the same paper, the authors point out that until now no significant universal electronic resources for sentiment analysis have been created. Main reasons of such a situation are variety of natural languages, essential differences between their vocabulary, morphology, syntax, and ambiguity of word meanings.

Thus, the solution of many tasks of sentiment analysis is impossible without development of approaches and methods that take into account specifics of national languages. This fact is especially underlined in the works devoted to the analysis of text sentiments in Chinese [5] and Indian [6].

Active research in the field of sentiment analysis of Russian texts began relatively recently, most of works have been published since 2012 [7]. One problem is a quite narrow range of used methods. For example, developers of sentiment analysis systems for Russian texts in the SentiRuEval-2015 competition, held during the Dialogue conference, applied only two types of methods: statistical ones based on word2vec and machine learning methods using neural networks [8]. Another problem is the almost complete absence of open Russian-language sentiment lexicons that significantly limits research in the described field [9]. Thus, the development of new methods and open tools for sentiment analysis of texts in Russian is very topical.

In this paper the authors propose an approach for sentiment classification of texts with the automatic construction and use of a thesaurus of a subject area. It is based on the Naive Bayes classifier. To suit Russian texts and improve classification quality, this algorithm is supplemented by calculations of event probabilities using data obtained from the automatically generated thesaurus. Based on this approach, a sentiment analysis tool has been developed. The evaluation of the proposed approach has been carried out for four different text corpora.

The paper is organized as follows. Section II describes state-of-the-art in sentiment classification of Russian texts. Section III contains the algorithm of automatic thesaurus generation. Section IV describes the approach of sentiment classification applying the thesaurus constructed from the classified texts. Section V shows results of experiments with several text corpora from different domains. Section VI discusses main advantages and limitations of the proposed approach and reveals directions of future research. Conclusion summarizes the paper.

II. Related work

In most works sentiment classification of Russian-language texts is performed using the same methods that are successfully applied for other languages: machine learning algorithms, neural networks, methods for information extraction from linguistic resources. The last ones vary for different languages, besides, in combination with other algorithms they allow to improve classification quality [7], therefore, they are quite promising approaches to investigate.

Ermakov and Ermakova [10] developed an approach to sentiment classification of words that takes into account word phonetic characteristics and applies them as vector features for the SVM algorithm. The F-measure of found word sentiments is higher for English words—0.85, for Russian ones this metrics equals only 0.72.

Chetviorkin and Loukachevitch [7] review state-of-the-art in the field of Russian-language sentiment classification including classification of texts. The best results are around 0.7–0.9 of F-measure and accuracy. They have been achieved by approaches that combine linguistic algorithms that use manual dictionaries and linguistic rules, and machine learning methods. The best results of 0.92 are achieved only for the narrow subject area of opinions about cameras, for other areas F-measure is significantly lower, around 0.75.

Koltsova et al. [9] propose a lexicon for sentiment analysis of political and social blogs in Russian. They describe a system for Russian-language sentiment analysis that includes a publicly available sentiment lexicon, a publicly available test collection with sentiment markup, and a crowdsourcing website that allows users to perform such markup of texts. The lexicon is aimed to detect sentiments in user-generated content (blogs, social media) related to social and political issues. Its prototype is based on Russian dictionaries and on the topic modeling performed on a large collection of blog posts. The Sentiment analysis task here is reduced to a relatively simple classification of texts into two classes: texts with prevailing negative emotions and prevailing positive emotions. Nevertheless, results of classification are quite low: precision equals 0.44 and recall equals 0.43.

Sakenovich et al. [11] compare several sentiment classification methods based on neural networks. The authors experiment with universal algorithms that do not depend on a language and do not use linguistic resources. Experiments with news in Russian and Kazakh show high results for standard metrics: the best precision, recall, and F-measure are about 0.84–0.86. They are achieved using recurrent neural networks.

Summarily, the best results in sentiment classification are reached using a combination of universal machine learning methods and language-specific dictionaries. Unfortunately, there are too few such linguistic resources for Russian language in open access. To overcome this issue, we propose an approach that can fully automatically generate and apply Russian thesaurus for a specific subject area. Such thesauri well describe area's peculiarities, so their use should improve sentiment classification of texts from particular areas.

III. AUTOMATIC RUSSIAN THESAURUS CONSTRUCTION

A. *Method for thesaurus construction*

The method of thesaurus generation is based on our previous work [12]. It has the same structure and utilizes several universal algorithms for term and relationship extraction, but uses additional methods and linguistic resources specific for Russian texts. Thesaurus construction includes the following steps:

1) Term extraction using an unsupervised algorithm.
2) Extraction of associations using statistical algorithms.
3) Extraction of synonyms using statistical and linguistic algorithms.
4) Extraction of hyponyms and hypernyms using linguistic algorithms.
5) Term filtering.

At the first step the method extracts all keywords that can be used as thesaurus terms from texts. After this step no new words are added. For this stage we used the unsupervised algorithm TextRank [13] whose quality of results for Russian texts was investigated in our previous work [14]. According to results of this study, supervised algorithms outperform others, but creating the material for training requires huge expenses and much of expert's work. Unsupervised algorithms, TextRank and Topical PageRank [15], show quite close quality characteristics. Actually, Topical PageRank is a variation of TextRank that more precisely selects keywords for specific texts but works slower. Nevertheless, the final set of keywords for all texts is the same for both algorithms. Generating the thesaurus, we need only such set without division by specific texts.

On the next three steps algorithms for search of semantic relationships between terms are consistently applied. If two algorithms propose different relationships for the same pair of terms, our method saves the relationship that was discovered later. For example, if two terms were associated on the second step and the method discovers a synonymous relationship between them on the third step, the association is deleted and they are marked as synonyms. Such behavior is based on the idea that synonym and hyponym-hypernym relationships are stronger than associative and more useful for natural language processing, as it was discovered in our previous research [16].

At the last step terms without relationships are removed, because such terms are useless for thesaurus application.

Relationship extraction by statistical algorithms is similar to our approach for English texts [12]. We use latent semantic analysis (LSA) [17] for associations and the Levenshtein distance [18] for synonym search. Besides, for associative relationships we apply word2vec (https://code.google.com/archive/p/word2vec/). This algorithm requires quite large amount of data for training to generate precise results. Fortunately, it already has them for different languages, including Russian, in open access.

A set of linguistic algorithms for Russian thesaurus generation differs more. Let us discuss this methods in more details.

B. *Relationship extraction by linguistic algorithms*

Linguistic algorithms extract semantic relationship using reliable existing resources that contain linguistic information verified by experts, such as the Synmaster dictionary of synonyms (http://usyn.ru/blog.php?id_blog=11), or the Russian thesaurus RuThes [19], or apply linguistic rules and patterns, for example, morpho-syntactic rules [20].

Synmaster contains about 1 200 000 words and from one to 20 synonyms for each word. RuThes contains 55 000 concepts (or terms), 158 000 words and phrases, 210 000 relationships between these concepts that include synonymous and hyponym-hypernymous relationships. It is the biggest thesaurus of Russian.

Our method extracts synonyms from both resources, hyponyms, and hypernyms from RuThes relying on a single principle: it adds only those semantic relationships that bound terms already existing in the automatic thesaurus, and does not handle new terms.

The method of morpho-syntactic rules considers terms connected by hierarchical relations, if one of them includes the second as a suffix string or part of a verbose term. In both cases the first is a hyponym, the second is a hypernym, because it is more general. For example, “Russian language” is a hyponym for the term “language”.

C. *Evaluation of the thesaurus*

After relationship extraction and singular term filtering the method of thesaurus construction results a specialized thesaurus generated with no expert's involvement.

The quality of the thesaurus can be proven in two ways. The first one is comparison with an existing thesaurus that is already considered as qualitative. Unfortunately, there are no such thesauri in open access for most of subject areas. The second way is applying the thesaurus to a natural language processing task, for example, sentiment classification. If results of classification are more relevant, when we apply a thesaurus, than we do not use it, we can conclude that the thesaurus allows to improve sentiment classification, therefore, it is helpful and qualitative for the solution of this task.

IV. Approach for sentiment classification with an automatically generated thesaurus

We propose the classification approach that allow to find out text sentiments basing on thesaurus relationships. It uses the idea introduced in our previous work about sentiment classification [21], that semantically close words like synonyms, hypernyms, hyponyms, and associations, have close sentiments. Therefore, if we have sentiments of several words, we can calculate sentiments of their thesaurus neighbors. In such a way we get word features that can be used in a standard classifier.

The core of the approach is based on the following algorithm:

1) Automatic thesaurus generation.
2) Calculation of a posteriori probabilities for a particular word belonging to each class, i.e., word sentiments.
3) Classification of sentiments of texts using word sentiments.

The automatic thesaurus is generated as described in the previous section. After creation of this resource we calculate sentiments for all words contained in the corpus basing on thesaurus relationships between them. Also we calculate sentiments using numbers of word occurrences, and such an algorithm is used as a baseline for our research. Finally, we classify texts by the Multinomial Naive Bayes algorithm [22] that is a variation of one of the best classifiers for sentiment classification, the Naive Bayes method [3].

A. Classifier

We use the Multinomial Naive Bayes model [22], where a text is represented as a bag of words, disregarding grammar and word order but keeping multiplicity. In this model each text is represented as a vector of pair of words and numbers, each number is a number of word occurrences in texts.

Let us introduce the following notation:

- $T = (t_1, t_2, \ldots, t_n)$ is a text corpus;
- for each word w in a text t, $q(w,t)$ is the number of occurrences of this word in text T_j;
- $V = (w_1, w_2, \ldots, w_m)$ is a vocabulary composed of all the words from the corpus;
- c_1, c_2 are classes of positive and negative texts (a priori or a posteriori) respectively;
- $M_i(w)$ is a number of occurrences of the word w in the class c_i;
- p_i is a priori probability of class c_i;
- $g_i(w)$ is a posteriori probability that the word w belongs to a class c_i;
- $score_i(t)$ is a posteriori probability that a text t belongs to a class c_i.

To train the classifier, we need to calculate the following values: a priori probabilities of each class:

$$p_i = \frac{|c_i|}{|T|},$$

and a prirori probabilities of words:

$$g_i(w) = \frac{M_i(w) + 1}{\sum_{j=1}^{|V|} (M_i(w_j) + 1)}.$$

The a posteriori probability that the text t belongs to the class c_i is described by the formula:

$$score_i(t) = \ln p_i + \sum_{w \in t} q(w,t) \ln g_i(w).$$

The classification of the text t is performed according to the class with the highest $score_i(t)$.

The algorithm of text sentiment classification using the multinomial model is presented in detail in algorithms 1 and 2. It consists of a training phase and a classification phase. On the first phase, the input parameters are the set of training texts and the set of classes. The algorithm creates a dictionary of stemmed words V, founds probabilities p_i and probabilities $g_i(w)$. The output is a trained model with configured parameters.

Algorithm 1 Training algorithm

```
for each class $c_i$ do
  for each text $t_j$ from $c_i$ do
    for each word $w$ from $t_j$ do
      stem $w$;
      add $w$ in $V$;
      $M_i(w) \leftarrow M_i(w) + 1$, the counter for $w$;
    end for
  end for
end for
for each class $c_i$ do
  calculate numbers of texts $|c_i|$ in the class $c_i$;
  calculate a prior probability $p_i \leftarrow |c_i|/|T|$;
  calculate $s_i$, the sum of all words occurrences in $c_i$;
  $s_i \leftarrow \sum_{t=1}^{|V|} (M_i(w_t) + 1)$;
  for each word $w$ from $V$ do
    calculate the probability $g_i(w) \leftarrow (M_i(w) + 1)/s_i$;
  end for
end for
Output data: $p$, $g$
```

On the second phase the algorithm 2 counts probabilities that the text t belongs to the class c_1 or c_2 and chooses the sentiment for the text. It is applied to all texts from the test set. Finally, we get sentiments for all texts.

Algorithm 2 Classification algorithm

```
t — a text for for classification;
for each word $w$ from $t$ do
    stem $w$;
    add $w$ in $V_t$;
end for
for each class $c_i$ do
    $score_i(t) \leftarrow \ln(p_i)$;
    for each word $w$ from $V_t$ do
        if $w$ entries $g[c_i]$ then
            $score_i(t) \leftarrow score_i(t) + \ln(g_i(w))$;
        end if
    end for
end for
if $score_1(t) > score_2(t)$ then
    $t$ belongs to the class $c_1$;
else
    $t$ belongs to the class $c_2$;
end if
Output data: the sentiment polarity of $t$
```

B. Calculation of word sentiments using thesaurus

The classifier described in the previous subsection does not use a thesaurus or another linguistic resource. We integrated into its algorithm the procedure that uses the automatically generated thesaurus. It should allow to take into account peculiarities of the subject area and define classes for texts more precisely.

The procedure consists from the word sentiments calculation based on the idea that semantically close words has similar sentiments. We compute them multiplying known sentiments by coefficients that depends of a type of a semantic relationship. These relationships allow to increase the probability that a word belongs to a class, i.e. has the particular sentiment.

The algorithm 3 shows the procedure handling synonyms from the thesaurus, calculations for associations, hyponyms, hyperonyms are the same. Here g is a vector of two vectors g_1 and g_2 of pairs, each pair contains a word and a probability that the word belongs to the class c_i. It is partially filled in for words contained in the training set, as it is described in the algorithm 1.

Algorithm 3 Classification with synonyms

```
Input data:
$S$ — a a vector of pairs, each pair contains a word and a vector of synonyms of this word;
$h$ — a coefficient of a semantic relationship that is varied from 0 to 1;
for each word $w_i$ from $S$ do
    for each word $w_j$ from the vector of synonyms of the word $w_i$ do
        for each class $c_k$ do
            $g_k(w_i) \leftarrow g_k(w_i) + (h \cdot g_k(w_j))$;
        end for
    end for
end for
Output data: $g$
```

We apply this procedure consistently to synonyms, hypernyms, hyponyms, and associations. In the end the algorithm returns sentiments for all words contained in the test set from the text corpus. The final word sentiment depends on all relationships that the word has in the thesaurus.

After the sentiment calculation we apply the algorithm 2. In the end it provide word sentiments calculated taking into account thesaurus relationships.

To estimate results of sentiment classification, we chose standard statistical metrics: accuracy, precision, recall, and F-measure. The accuracy is the fraction of texts for which the classifier made a correct decision. The precision is the fraction of texts actually belonging to the given class among all texts that the algorithm assigned to this 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.

V. Experiments

We experimented with the approach, proposed in the previous section, and four different Russian-language corpora.

The corpus with opinions about banks was taken from the site http://www.banki.ru. It contains 279 texts, where 114 ones are positive and 165—negative. In average, each text contains 159 words or 1 045 characters.

The corpus with opinions about mobile operators was taken from the resource http://www.banki.ru/telecom/responses/list/mobile. It contains 696 texts, where 353 ones are positive and 343—negative. In average, each text contains 198 words or 1 302 characters.

The corpus of tweets (http://linis-crowd.org) consists of short texts on different topics. It contains 4 320 texts, where 2 160 ones are positive and 2 160—negative. In average, each text contains 157 words or 1 044 characters.

The corpus of microblogs (http://study.mokoron.com) also consists of very short texts on different topics. It contains 213 412 texts, where 108 713 ones are positive and 104 699—negative. In average, each text contains 11 words or 65 characters.

These corpora were input data for the sentiment analysis tool, developed on top of the proposed approach. For preprocessing of texts we applied the Porter stemming algorithm [23]. For text processing during thesaurus generation we used the NLTK framework. The developed tool implements both algorithms for sentiment classification: the Multinomial Naive Bayes classifier and the same classifier with built-in automatic thesaurus generation.

For the procedure with the thesaurus 3 we varied coefficients of relationships (h) from 0.1 to 1.0 with the step 0.1. In all cases if $h > 0.5$ classification results are better than for smaller h, and they are close to each other in absolute values. So we can conclude that all types of semantic relationships affect the classification quality in the same way.

To estimate results of sentiment classification, we chose standard statistical metrics: accuracy, precision, recall, and F-measure.

TABLE I. SENTIMENT CLASSIFICATION OF RUSSIAN-LANGUAGE OPINIONS ABOUT BANKS

Algorithm	Accuracy	P_{neg}	R_{neg}	F_{neg}	P_{pos}	R_{pos}	F_{pos}
without thesauri	0.880	0.859	0.884	0.871	0.913	0.867	0.889
with the automatic thesaurus	0.942	0.919	0.970	0.944	0.969	0.921	0.944

TABLE II. SENTIMENT CLASSIFICATION OF RUSSIAN-LANGUAGE OPINIONS ABOUT MOBILE OPERATORS

Algorithm	Accuracy	P_{neg}	R_{neg}	F_{neg}	P_{pos}	R_{pos}	F_{pos}
without thesauri	0.787	0.867	0.813	0.839	0.724	0.789	0.755
with the automatic thesaurus	0.861	0.919	0.866	0.892	0.766	0.851	0.807

TABLE III. SENTIMENT CLASSIFICATION OF RUSSIAN-LANGUAGE TWEETS

Algorithm	Accuracy	P_{neg}	R_{neg}	F_{neg}	P_{pos}	R_{pos}	F_{pos}
without thesauri	0.643	0.735	0.684	0.709	0.573	0.637	0.603
with the automatic thesaurus	0.697	0.763	0.723	0.742	0.600	0.735	0.661

TABLE IV. SENTIMENT CLASSIFICATION OF RUSSIAN-LANGUAGE MICROBLOGS

Algorithm	Accuracy	P_{neg}	R_{neg}	F_{neg}	P_{pos}	R_{pos}	F_{pos}
without thesauri	0.633	0.621	0.663	0.658	0.611	0.609	0.610
with the automatic thesaurus	0.700	0.674	0.729	0.700	0.728	0.675	0.700

Tables I, II, III, and IV shows results of experiments with Russian corpora. P, R, and F mean precision, recall, and F-measure for positive (*pos*) or negative (*neg*) texts respectively. Coefficients of relationships equal 1.0.

We can see that in all cases the algorithm with the automatically generated thesaurus significantly outperforms one without the thesaurus. Accuracy is better by 5–6 % for all corpora, F-measure for positive texts is better by 6–9 %, Precision and recall for positive texts are also higher for the algorithm with the thesaurus.

The algorithm without the thesaurus performs slightly better in precision for negative microblogs, but recall for them is higher for the algorithm with the thesaurus. Judging from the F-measure that is almost equal for negative texts from this corpus, no algorithm outperforms another one in this case.

Negative texts are classified better than positive ones: for tweets and microblogs F-measure is better by 10 % in most cases, it is around 0.6 for positive and around 0.7 for negative. Positive and negative opinions about banks are processed with the same quality.

Besides, texts with opinions about banks and mobile operators that are united with the same topic, are classified better than other corpora in absolute values: all metrics are about 0.85–0.97, whereas quality characteristics for tweets and microblogs that contain texts with various topics, are from 0.57 to 0.76.

Comparing results of our approach with similar methods described in Section II, we can see that our algorithm shows the classification quality comparable with best state-of-the-art results for narrow subject areas: its best accuracy and F-measure equal 0.94, and in the survey [7] this metrics achieve 0.92.

Thus, from experiment results we can conclude that using the automatically generated thesaurus improves quality of sentiment classification of short texts.

VI. DISCUSSION

An important feature of the proposed approach is that it considers words as separate text units, so the structure of sentences is not taken into account. Relationships between words are used only at the stage of the thesaurus application. Synonyms, hyponyms, hyperonyms, and associations are actually considered as similar words. It allows to create a fully automatic sentiment analysis tool. The expert participates only in the generation of the training set for the supervised algorithm.

The analysis of experiments shows that the proposed approach provides good results for narrow subject areas. Perhaps, it happens due to the fact that in this case sentiment of the text is defined by the use of certain terms. The structure of sentences is less important.

Corpora of tweets and microblogs contain texts from broader subject areas. Results of experiments with them are lower. A possible direction to improve the performance of the approach is taking into account structure of the text. In addition to separate words, it seems reasonable to distinguish and consider word combinations of different lengths in the procedure of calculation of frequencies or probabilities. Such phrases can be found by statistical methods according to frequencies of occurrence or indicated by experts. Searching and applying of linguistic patterns is also a way to use properties of text structure in algorithms.

Possible improvements described above require increase of expert's working time. It is especially true for design of patterns. Although methods to automate this process exist [24], their implementation in existing solutions requires additional research.

VII. CONCLUSION

The article describes the approach of sentiment classification of texts in Russian language. It embeds the procedure of the thesaurus application into the machine learning algorithm, Multinomial Naive Bayes classifier. The thesaurus is generated and applied fully automatically. Experiments conducted with different Russian-language texts corpora, show that the algorithm with the thesaurus improves quality of sentiment classification in comparison with the algorithm without the thesaurus. Besides, the best results are obtained in the case of classification of texts from narrow subject areas.

The future directions for investigations are taking into account a structure of sentences and applying of linguistic patterns.

References

[1] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[2] B. Awrahman and B. Alatas, "Sentiment analysis and opinion mining within social networks using konstanz information miner," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, no. 1, pp. 15–22, 2017.

[3] 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.

[4] S. L. Lo, E. Cambria, R. Chiong, and D. Cornforth, "Multilingual sentiment analysis: from formal to informal and scarce resource languages," *Artificial Intelligence Review*, vol. 48, no. 4, pp. 499–527, 2017.

[5] H. Peng, E. Cambria, and A. Hussain, "A review of sentiment analysis research in chinese language," *Cognitive Computation*, vol. 9, no. 4, pp. 423–435, 2017.

[6] S. Sharma, S. Bharti, and R. K. Goel, "Sentiment analysis of indian language," *International Research Journal of Engineering and Technology (IRJET)*, vol. 05, no. 05, pp. 4251–4253, 2018.

[7] I. Chetviorkin and N. Loukachevitch, "Evaluating sentiment analysis systems in russian," in *Proceedings of the 4th biennial international workshop on Balto-Slavic natural language processing*, 2013, pp. 12–17.

[8] I. A. Andrianov, V. D. Mayorov, and D. Y. Turdakov, "Modern approaches to aspect-based sentiment analysis," *Proceedings of the Institute for System Programming of the RAS*, vol. 27, no. 5, pp. 5–22, 2015.

[9] O. Y. Koltsova, S. Alexeeva, and S. Kolcov, "An opinion word lexicon and a training dataset for russian sentiment analysis of social media," in *Computational Linguistics and Intellectual Technologies: Materials of DIALOGUE 2016 (Moscow)*, 2016, pp. 277–287.

[10] S. Ermakov and L. Ermakova, "Sentiment classification based on phonetic characteristics," in *European Conference on Information Retrieval*. Springer, 2013, pp. 706–709.

[11] N. S. Sakenovich and A. S. Zharmagambetov, "On one approach of solving sentiment analysis task for kazakh and russian languages using deep learning," in *International Conference on Computational Collective Intelligence*. Springer, 2016, pp. 537–545.

[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 *19th Conference of Open Innovations Association (FRUCT)*. IEEE, 2016, pp. 138–144.

[13] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," in *Proceedings of EMNLP*. Association for Computational Linguistics, 2004, pp. 404–411.

[14] I. Paramonov, K. Lagutina, E. Mamedov, and N. Lagutina, "Thesaurus-based method of increasing text-via-keyphrase graph connectivity during keyphrase extraction for e-tourism applications," in *International Conference on Knowledge Engineering and the Semantic Web*. Springer, 2016, pp. 129–141.

[15] Z. Liu, W. Huang, Y. Zheng, and M. Sun, "Automatic keyphrase extraction via topic decomposition," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 366–376.

[16] N. S. Lagutina, K. V. Lagutina, I. A. Shchitov, and I. V. Paramonov, "Analysis of influence of different relations types on the quality of thesaurus application to text classification problems," *Modelirovanie i Analiz Informacionnyh Sistem*, vol. 24, no. 6, pp. 772–787, 2017, in Russian.

[17] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser, "Latent semantic analysis," in *Proceedings of the 16th international joint conference on Artificial intelligence*. Citeseer, 2004, pp. 1–14.

[18] S.-Y. Noh, S. Kim, and C. Jung, "A lightweight program similarity detection model using XML and Levenshtein distance." in *FECS*. Citeseer, 2006, pp. 3–9.

[19] N. Loukachevitch and B. Dobrov, "Ruthes linguistic ontology vs. russian wordnets," in *Proceedings of the Seventh Global Wordnet Conference*, 2014, pp. 154–162.

[20] E. Lefever, M. Van de Kauter, and V. Hoste, "Evaluation of automatic hypernym extraction from technical corpora in english and dutch," in *9th International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), 2014, pp. 490–497.

[21] I. Shchitov, K. Lagutina, N. Lagutina, and I. Paramonov, "Sentiment classification of long newspaper articles based on automatically generated thesaurus with various semantic relationships," in *Proceedings of 21st Conference of Open Innovations Association FRUCT*. IEEE, 2017, pp. 290–295.

[22] C. D. Manning, P. Raghavan, H. Schütze *et al.*, *Introduction to information retrieval*. Cambridge university press Cambridge, 2008, vol. 1, no. 1.

[23] P. Willet, "The porter stemming algorithm: then and now," *Program*, vol. 40, no. 3, pp. 219–223, 2006.

[24] A. Shelmanov, M. Kamenskaya, M. Ananyeva, and I. Smirnov, "Semantic-syntactic analysis for question answering and definition extraction," *Scientific and Technical Information Processing*, vol. 44, no. 6, pp. 412–423, 2017.