Analysis of Influence of Different Relations Types on the Quality of Thesaurus Application to Text Classification Problems

N. S. Lagutina*, K. V. Lagutina**, I. A. Shchitov***, and I. V. Paramonov****

*Demidov Yaroslavl State University, Yaroslavl, 150003 Russia
**e-mail: lagutinans@rambler.ru
***e-mail: lagutinakv@mail.ru
****e-mail: ivan.shchitov@e-werest.org
****e-mail: Ilya.Paramonov@fruct.org

Received October 16, 2017

Abstract—The main purpose of the article is to analyze how effectively different types of thesaurus relations can be used for solutions of text classification tasks. The basis of the study is an automatically generated thesaurus of a domain, that contains three types of relations: synonymous, hierarchical and associative. To generate the thesaurus the authors use a hybrid method based on several linguistic and statistical algorithms for extraction of semantic relations. The method allows to create a thesaurus with a sufficiently large number of terms and relations among them. The authors consider two problems: topical text classification and sentiment classification of large newspaper articles. To solve them, the authors developed two approaches that complement standard algorithms with a procedure that take into account thesaurus relations to determine semantic features of texts. The approach to topical classification includes the standard unsupervised BM25 algorithm and the procedure, that take into account synonymous and hierarchical relations of the thesaurus of the domain. The approach to sentiment classification consists of two steps. At the first step, a thesaurus is created, whose terms weight polarities are calculated depending on the term occurrences in the training set or on the weights of related thesaurus terms. At the second step, the thesaurus is used to compute the features of words from texts and to classify texts by the algorithm SVM or Naive Bayes. In experiments with text corpora BBCSport, Reuters, PubMed and the corpus of articles about American immigrants, the authors varied the types of thesaurus relations that are involved in the classification and the degree of their use. The results of the experiments make it possible to evaluate the efficiency of the application of thesaurus relations for classification of raw texts and to determine under what conditions certain relations affect more or less. In particular, the most useful thesaurus relations are synonymous and hierarchical, as they provide a better quality of classification.

Keywords: thesaurus, semantic relations, thesaurus relations, topical classification, sentiment classification

DOI: 10.3103/S0146411619070277

1. INTRODUCTION

Automatic processing of texts in natural languages is an essential part of modern information technology. The need for text analysis appears in solving many problems, such as information retrieval, search for answers to questions; document rubrication, classification, annotation; machine translation, and much more.

A huge amount of information is contained both in document collections on the Internet, in books, articles, specialized information databases, etc. For a qualitative and effective use of these resources, it is not enough to consider a text as a set of independent terms, it is necessary to take into account structure of sentences and relations between words and terms of the domain. It requires a model combining language knowledge and domain features. A thesaurus can be such a model, because it includes domain terms and relations between them.

Initially, thesauri were created to index documents manually. In the field of automation, one of the first applications of a thesaurus was related to machine translation in 1954 [1]. However, in automatic text processing there is no intermediary person between a text and a work result, who uses not only a thesaurus or a dictionary, but also her knowledge about the object of research. There is only an automatic process and a thesaurus that should include both a set of commonly used terms with relations between them, and knowledge that an expert uses to analyze a text. Thus, a thesaurus designed for automatic text processing
should contain significantly more information about the domain, in particular, more terms and more relations [2].

The expansion of the thesaurus base leads to increase and complication of relations between thesaurus terms. Therefore, in addition to the development and improvement of methods for extracting such relations, it is necessary to understand how different types of relations affect the efficiency of thesauri application in different tasks of automatic text processing. In many state-of-the-art studies, the relations between terms are actively used, but practically nowhere types of relations are differentiated, and, moreover, there is not analysis of the influence of different relations types on the quality of solution of a problem in question. That is why, the authors of this article raised the question of the influence degree of relations between the applied thesaurus terms as a part of their research in the field of sentiment text classification. This work describes experiments conducted in this area and their results.

2. THESAURUS AND TYPES OF RELATIONS

A thesaurus is a dictionary that includes terms or concepts organized in such a way as to establish explicit relations between these terms [3]. A term in the thesaurus is a word or phrase, usually in the form of a noun or a noun phrase, that is an exact designation of a certain concept of a domain. Also the thesaurus contains different types of relations between its terms. One of the main ones is the synonymy relation, which expresses the equality or similarity of word meanings, as well as morphemes, syntactic constructions, phrases. In classical thesauri each group of synonyms forms a separate dictionary unit—a synset. Among synonyms, a descriptor is chosen that is a term representing a particular concept of the domain. Other synonymous terms from the synset are called ascriptors [4].

According to the standard, the main types of relations between thesaurus synsets are the following ones:

- a genus—species;
- a part—a whole;
- cause—consequence;
- raw materials—a product;
- a process—an object;
- a process—a subject;
- a property—a property carrier;
- functional similarity;
- administrative hierarchy;
- antonymy.

In thesauri used for automatic text processing all these relations are usually divided into two types: hierarchical and associative, where the associative relation indicates the presence of a relation between concepts that differs from synonymous and hierarchical. Thus, the thesaurus is a domain model, consisting of terms and synonymous, hierarchical, and associative relations between them.

3. RELATED WORK

The influence of thesaurus relations on text analysis quality is rarely analyzed in the scientific literature. Existing research on this topic is mainly related to the task of indexing documents and the similar problem of thematic modeling, the other areas of text processing are practically not studied. Below the authors analyzed several works devoted to the analysis of the thesaurus structure and its significance for solving practical tasks.

The paper [5] analyzes the influence of the thesaurus structure on the quality of subject indexing. The authors developed a method that includes the random walk algorithm. This algorithm processes different semantic relations with different probabilities that leads to different influence of thesaurus relations to result. The authors experimented with four manually constructed thesauri: AGROVOC, HEP, NALT, and MeSH; and probabilities for hierarchical and associative relations. The obtained average accuracy of 19.01% is quite low. It was achieved by several possible combinations of method’s parameters: for three thesauri it is necessary to set a high probability for associations and a low probability for other relations; the opposite situation appears for the HEP thesaurus, hypernyms are more powerful than other relations.

The authors of [6] also use a thesaurus to index documents. They developed the semi-automatic tool DigiDoc MetaEdit that allows a user to match terms from the thesaurus with HTML documents. The tool
creates a set of possible keywords depending on the frequency of occurrence and relevance of the term and supplements this set with thesaurus synonyms, hyponyms, hypernyms, and associations. In this case the algorithm parameters are set by the user.

Experiments were conducted with 100 Spanish journal articles from the BiD portal and the TLIS thesaurus (Thesaurus on Library and Information Science). The authors compared efficiency of the automatic part of the instrument for cases when it uses different types of thesaurus relations. The best recall 73% was achieved using all types of relations. The use of horizontal relations only allows to reach 64%, hierarchical—58%. The algorithm without a thesaurus demonstrates the lowest recall—49%.

N. Lukashevich et al. [7] propose a method for solving a thematic modeling problem that complements sets of related topics with terms from a manually constructed thesaurus. The authors conducted several experiments with algorithms without a thesaurus, with synonyms, and with synonyms and hypernyms, using the kernel uniqueness metric to measure quality. It describes how different topics defined by groups of related thesaurus terms differ from each other. As a result, a combination of two relations provides the best value for this metric in the interval of 0.4–0.7, whereas in the case of an algorithm without a thesaurus, the value is between 0.3 and 0.5.

These studies demonstrate that different ways of using thesaurus relations significantly affect the result of the analysis of a natural language text. Unfortunately, the degree of such significance is understudied, especially in tasks of text classification by topic or sentiment. The methods for solving these problems usually apply one type of relations, or all relations are unified and treated identically as associations.

The research [8] solves the problem of text classification by topics using the OpenOffice Thesaurus of the Brazilian Portuguese language for matching texts and terms from the ontology. For each word from the text the algorithm searches the closest related terms from the thesaurus and selects ones that are also contained in the ontology. The classifier proposed by the authors works basing on these data. In result, precision and recall grow by 7–10% (up to 60–70%), comparing with the Support Vector Machine method without a thesaurus. At the next stage of the research, the thesaurus and ontology are supplemented by terms and relations manually with the help of experts, and the authors also change the similarity measure of a term and a text. It allows to achieve one of the best results in the field: 96% of precision and recall.

The paper [9] is devoted to investigation of the problem of Internet user reviews classification into positive and negative. The authors propose a classifier based on the use of an automatically generated thesaurus. The thesaurus is built for the full text corpus, on which the classifier is subsequently be trained and tested. Words are marked as positive and negative depending on sentiments of texts, where these words occur more often. Terms appearing in one sentence are considered to be related by an associative relation, and the weight of this relation also depends on the frequency of occurrence of the terms in the same text. Next, weights of term-review pairs are calculated depending on numbers and weights of thesaurus relations between this term and other words from the review. These features form vectors for reviews and finally are used as an input for the Maximum Entropy classifier. As a result, the following results were obtained for 800 positive and 800 negative reviews from Amazon: precision without a thesaurus is 62–72%, precision with a thesaurus is 72–87% that is one of the best for this task. The proposed method is positioned as domain-independent, i.e., it can potentially be applied to reviews of any topic.

Thus, despite the fact that most researchers do not differentiate semantic relations of various types between thesaurus terms for text classification, this idea is successfully applied to several problems of natural language text analysis, so we can try to expand it to the field of classification.

4. TOPICAL TEXT CLASSIFICATION USING A THESAURUS

Topical text classification is the task of dividing a text corpus into several classes (possibly overlapping), each of which denotes the main topic of the text. For the purpose of this study, the authors use a topical classification algorithm that complements the existing BM25 algorithm [10] with a procedure developed by the authors that generates a specialized thesaurus and apply it during classification.

4.1. Thesaurus Generation

A specialized thesaurus is generated fully automatically from a texts corpus of a given domain to be classified. It allows to create a domain model that describes the main topics of the field and relations between them. A detailed description of the developed hybrid algorithm and the resulting thesaurus is given in the authors’ recent article [11]. This algorithm combines several existing statistical and linguistic methods for
extracting thesaurus relations. Due to this combination it allows to construct a significant number of different semantic relations between terms. A brief description of the steps of the algorithm is as follows:

1. Extraction of terms using the TextRank algorithm [12].
2. Extraction of semantic relations between the terms (associative, synonymous, and hyponymic-hypernymic) by the hybrid method.
3. Filtering terms without relations.

As a result of the algorithm, a thesaurus is created with a large set of terms and different relations between them. Also, the main advantage of the algorithm is that it works fully automatically and does not require expert participation at any stage, so the thesaurus is generated quite quickly. Comparison with the existing manually constructed thesaurus showed that the automatic thesaurus shows good precision and recall of terms and relations, so it can be successfully used to process texts [11].

4.2. Text Classification

The constructed thesaurus is used by the authors for topical classification of natural language texts. Automatic classification is performed using the standard unsupervised BM25 algorithm that additionally takes into account relations between thesaurus terms. The input to the algorithm is a corpus of texts that are not labeled by class, a list of classes, and an automatically generated thesaurus.

The algorithm consists of the following steps:

1. Selecting all nouns and adjectives from the corpus and calculating frequencies of their occurrence.
2. Creating a list of terms-neighbors in a thesaurus for each class from the input class list.
3. Ranking text-class pairs by the BM25 algorithm depending on the frequency of occurrence of class terms and their neighbors.

Firstly, the algorithm selects individual terms from the texts, constructs an inverted index for each text, and computes the frequency of occurrence of each word.

Secondly, it finds first-order neighbors for class terms, i.e., terms that have a direct relation with them—synonymous, hyponymic, or hypernymic. Associations are not processed, because they often denote a rather weak semantic relation and can even link words from different topics, for example, “blood clot” and “heart valve.” For synonymous and hierarchical relations, the situation is opposite: in most cases they reflect relations between terms from the same topic, so they suit better for thematic comparison of texts and classes.

Finally, BM25 is applied to the text-class pairs to rank texts depending on terms of classes and their terms-neighbors from the thesaurus. This algorithm is popular for solving various problems of text analysis, including classification [13]. Also, it does not require adjusting additional parameters and training on existing data corpora, so it is easily supplemented by processing information from a thesaurus, therefore, it suits well for the purposes of this study.

5. SENTIMENT TEXT CLASSIFICATION USING A THESAURUS

The problem of sentiment text classification implies the division of a text corpus into two or more classes, depending on the sentiment of a text as a whole: positive or negative, or positive, negative, and neutral. In this paper the authors investigate the problem of classifying large news articles into two classes using an automatically generated thesaurus.

The approach proposed by the authors consists of the following two steps: creation of a sentiment thesaurus and classification of texts. At the first step the thesaurus is created fully automatically, at the second one the articles are classified using the Support Vector Machines or the Naive Bayesian Classifier. Feature vectors are composed basing on weights of thesaurus terms. Both steps get as an input a corpus of raw news articles preliminarily divided into training and test sets. The training set was initially marked by philologists.

5.1. Thesaurus Generation

All words that can have positive or negative semantics are chosen as terms of a sentiment thesaurus: nouns, adjectives, verbs, and adverbs. Relations between them are extracted according to the algorithm described in the previous section. The result of this algorithm is a specialized thesaurus with a large number of synonymous, associative, and hyponym-hypernym relations. A large number of relations allow to calculate more accurate numerical values of a term sentiment, since the sentiment in this case would depend on several thesaurus neighbors of the term.
At the last stage of generation the terms are assigned with sentiment weights denoting sentiments: from −1 to 0 for negative terms and from 0 to 1 for positive ones. Firstly, the algorithm calculates weights of terms that are found in the training text set. Since the texts in this set are already marked by sentiments, they can be spread to terms.

The authors suggested that positive terms are more likely to be found in positive texts and negative ones—in negative texts. Therefore, they used the following formula for a term weight: \( w = (p - n)/(p + n) \), where \( p \) denotes, how many times the term appeared in positive texts, \( n \)—in negative ones. This formula allows to assign sentiments from the range between −1 and 1 as described above.

Next, weights of terms missing in the training set, but existing in the test one are calculated. The authors suggested that terms that have a common thesaurus relation also have a close sentiment. Therefore, their sentiments can be calculated from weights of their thesaurus neighbors. To implement this idea, each type of relations was assigned its own coefficient for sentiment conversion, ranged from 0.1 to 1.0.

The algorithm for weight calculation based on thesaurus relations is the following. If a term has synonyms marked by sentiments, their average weight is taken and multiplied by the relation coefficient, the result is written as the term weight. If the term does not have such synonyms, but has hypernyms, the algorithm does the same with them. And if the term has only associations, the average value multiplied by the coefficient is calculated for them. Since there are many relations in the thesaurus, each term would have at least one marked neighbor.

Thus, the generated thesaurus contains a set of terms from the domain, semantic relations between them, and sentiment weights of all terms.

### 5.2. Text Classification

Term sentiments from the generated thesaurus are applied to calculate feature vectors used in the classification algorithm. Each text is assigned a numerical vector, the size of which is equal to the number of thesaurus terms. Each element of the vector corresponds to a specific thesaurus term and is calculated as the composition \( w \cdot F \), where \( w \) is the weight of the term and \( F \) is a standard statistical feature of the term-text pair, depending on the frequency of occurrence of the term in the corpus or text and/or its thesaurus weight. The authors used four different statistical features: TF*IDF, the Gini index, info gain, mutual information, and chi-square statistic \( \chi^2 \) [14].

At the last stage, feature vectors are used as an input of one of standard binary classifiers. Many research show that the best classifiers for calculating text sentiments are machine learning algorithms [15]. Among these algorithms, Support Vector Machines and the Naive Bayesian Classifier are considered to be ones of the most effective methods [16], so they were chosen by the authors for experiments.

### 6. TEXT CORPORA AND THE EXPERIMENTING TECHNIQUE

The classification algorithms proposed by the authors were tested on several English texts corpora from different domains:

- The PubMed corpus (https://www.nlm.nih.gov/databases/download/pubmed contains 1000 medical articles from 63 classes, the total number of words is 154 850.
- The Reuters corpus (http://www.daviddlewis.com/resources/testcollections/reuters21578/) contains 1534 economical articles from 15 classes, the total number of words is 294 813.
- The BBCSport corpus (http://mlg.ucd.ie/datasets/bbc.html) contains 737 sports news articles from 5 classes, the total number of words is 253 667.
- The corpus of articles about American immigrants from The New York Times, The New York Post, and The Los Angeles Times contains 56 articles, of which 34 are positive and 22 are negative, the total number of words is 37 669. The corpus was created and marked by sentiments by philologists.

The first three text corpora were used for topical classification, the last one—for sentiment classification. Also, the fourth corpus was divided into training and test sets, each contained 17 positive and 11 negative articles.

Both classification algorithms were implemented in the Python programming language using the NLTK library (http://www.nltk.org). This library contains implementations of standard text processing and machine learning algorithms, including the Support Vector Machine method (SVM) and the Naive Bayesian Classifier.
The procedure for estimation of classification results also was written in Python. The authors used standard quality metrics for evaluation: precision, recall, F-measure, and accuracy [17]. It should be noted that micro-averaged metrics were chosen for topical classification. They consider recall and precision for all classes at the same time, since the texts have many classes and computing the individual metrics for each class will complicate the analysis of the result. And otherwise, sentiment classification implies a division into only two classes, therefore precision, recall, and F-measure were calculated separately for positive and negative texts.

### Table 1. Topical classification of the BBCSport corpus

| Relations               | Filtering threshold | P   | R   | F   | A   |
|-------------------------|---------------------|-----|-----|-----|-----|
| No                      | 4                   | 0.835 | 0.096 | 0.173 | 0.815 |
| Synonyms                | 4                   | 0.828 | 0.098 | 0.175 | 0.815 |
| Hyponyms                | 4                   | 0.790 | 0.107 | 0.189 | 0.816 |
| Hypernyms               | 0                   | 0.403 | 0.479 | 0.438 | 0.754 |
| Synonyms and hyponyms   | 4                   | 0.784 | 0.109 | 0.191 | 0.816 |
| Synonyms and hypernyms  | 0                   | 0.399 | 0.482 | 0.437 | 0.751 |

### Table 2. Topical classification of the Reuters corpus

| Relations | Filtering threshold | P   | R   | F   | A   |
|-----------|---------------------|-----|-----|-----|-----|
| No        | 4                   | 0.738 | 0.246 | 0.369 | 0.933 |
| Synonyms  | 2                   | 0.519 | 0.559 | 0.538 | 0.924 |
| Synonyms  | 4                   | 0.749 | 0.276 | 0.404 | 0.935 |
| All       | 0                   | 0.329 | 0.746 | 0.457 | 0.859 |

### Table 3. Topical classification of the PubMed corpus

| Relations | Filtering threshold | P   | R   | F   | A   |
|-----------|---------------------|-----|-----|-----|-----|
| No        | 4                   | 0.234 | 0.065 | 0.101 | 0.943 |
| Synonyms  | 4                   | 0.216 | 0.072 | 0.109 | 0.941 |
| Hypernyms | 2                   | 0.169 | 0.169 | 0.169 | 0.917 |
| All       | 0                   | 0.125 | 0.201 | 0.154 | 0.890 |

The procedure for estimation of classification results also was written in Python. The authors used standard quality metrics for evaluation: precision, recall, F-measure, and accuracy [17]. It should be noted that micro-averaged metrics were chosen for topical classification. They consider recall and precision for all classes at the same time, since the texts have many classes and computing the individual metrics for each class will complicate the analysis of the result. And otherwise, sentiment classification implies a division into only two classes, therefore precision, recall, and F-measure were calculated separately for positive and negative texts.

### 7. RESULTS OF EXPERIMENTS

The authors conducted several experiments with the developed algorithms and at the same time varied algorithms' parameters: they used different relations types individually and in combination. For the topical classification algorithm the filtering threshold was also changeable: when the BM25 result was less than a certain threshold (0, 2, or 4), a class-text pair was rejected. For the sentiment classification algorithm various relations coefficients were considered from 0.1 to 1.0 with a step of 0.1. In this case coefficients for different relation types were used to analyze influence degree of each relation type separately.

Tables 1—3 present the best results of topical classification for each text corpus. The symbols $P$, $R$, $F$, and $A$ denote precision, recall, F-measure, and accuracy respectively. “No” in the first column means that this experiment was conducted with an algorithm without relations, “all”—that all relations were used: hyponyms, hypernyms, and synonyms.

From the results of the BBCSport news corpus classification it is clear that the best precision of 0.835, but the lowest recall of 0.096 are provided by the algorithm that does not use a thesaurus. The algorithm that uses synonyms and hypernyms and at the same time does not use filtering has the best recall of 0.482 and the worst precision of 0.399. It is worth mentioning that its F-measure of 0.437 differs from the best by only 0.1, which is the result of an algorithm with hypernyms. The algorithms with synonyms and hyponyms have the best accuracy of 0.816, i.e., they give the largest number of correct answers, but their F-measure of all of them is low: 0.175—0.191. Thus, the best features for the BBCSport corpus are achieved using hierarchical relations.
For the Reuters corpus synonyms turned out to be the most significant relations. There are algorithms using them that provide the best F-measure of 0.538 and the accuracy of 0.935.

Classification results of the PubMed corpus were low for all combinations of thesaurus relations. The algorithm provided too few class-text pairs, so the precision and recall do not exceed 0.25. Accuracy is high due to the fact that it takes into account how many texts were correctly not correlated with inappropriate classes and due to the large number of classes (63) this number is quite large.

The quality of sentiment classification is higher than topical in absolute numbers. Its results are presented in Tables 4–10. $P$, $R$, and $F$ denote precision, recall, and F-measure, the lower indices $neg$ and $pos$—negative and positive text classes. Words $Hyp$, $Syn$, and $Assoc$ implies the use of hypernyms, synonyms, and associations in the experiment.

Each table describes results of classification of articles about American immigrants for pairs of a classifier and a term weight feature. Tables do not include pairs with TF*IDF and the Bayesian classifier and $\chi^2$ pair, since their results became lower than the others and did not depend on the method of thesaurus use. Note that in all cases results of algorithms with a thesaurus turned out to be higher than results of the algorithm without it, and in most cases they were significantly higher.

In the combination of SVM and the Gini index (Table 4) the use of hypernyms give the best result by all metrics, and the coefficient of hypernyms practically do not matter: it can be 0.1, 0.3, 0.7, or 0.9. For the combination of the Bayesian classifier and the Gini index (Table 5) the same can be said about synonyms, hypernyms, and associations in the experiment.

In the combination of SVM and the Gini index (Table 4) the use of hypernyms give the best result by all metrics, and the coefficient of hypernyms practically do not matter: it can be 0.1, 0.3, 0.7, or 0.9. For the combination of the Bayesian classifier and the Gini index (Table 5) the same can be said about synonyms, hypernyms, and associations in the experiment.

In the combination of SVM and the Gini index (Table 4) the use of hypernyms give the best result by all metrics, and the coefficient of hypernyms practically do not matter: it can be 0.1, 0.3, 0.7, or 0.9. For the combination of the Bayesian classifier and the Gini index (Table 5) the same can be said about synonyms, hypernyms, and associations in the experiment.

In the combination of SVM and the Gini index (Table 4) the use of hypernyms give the best result by all metrics, and the coefficient of hypernyms practically do not matter: it can be 0.1, 0.3, 0.7, or 0.9. For the combination of the Bayesian classifier and the Gini index (Table 5) the same can be said about synonyms, hypernyms, and associations in the experiment.

In the combination of SVM and the Gini index (Table 4) the use of hypernyms give the best result by all metrics, and the coefficient of hypernyms practically do not matter: it can be 0.1, 0.3, 0.7, or 0.9. For the combination of the Bayesian classifier and the Gini index (Table 5) the same can be said about synonyms, hypernyms, and associations in the experiment.

In the combination of SVM and the Gini index (Table 4) the use of hypernyms give the best result by all metrics, and the coefficient of hypernyms practically do not matter: it can be 0.1, 0.3, 0.7, or 0.9. For the combination of the Bayesian classifier and the Gini index (Table 5) the same can be said about synonyms, hypernyms, and associations in the experiment.
Summing up the results of all experiments, the authors found the following tendency. Use of thesaurus relations significantly increases the quality of the result for all metrics compared to algorithms without a thesaurus. The most significant relations in the thesaurus are synonymous and hierarchical, because in the solutions of both classification problems they provided the best results: F-measure of 0.5–0.8 and accuracy of 0.75–0.9. Associative relations did not make a significant contribution to the result quality, as it can be seen from almost all experiments with different parameters.

Another tendency consists in the dependence of the method of using the thesaurus on the text domain. Experiments showed that hierarchical relations give better results for BBCSport news corpus, and synonymous ones—for Reuters economic text corpus. For large newspaper articles about immigrants two types of relations became important at once: synonyms and hypernyms, and none of them can be distinguished as the most significant.

### 8. CONCLUSIONS

The study of classification results for texts from different domains allows not only to draw a general conclusion about the larger influence of synonymous and hierarchical relations in comparison with associative ones, but also highlight more detailed tendencies. From the authors’ point of view, the most important conclusion is that, in the classification of short news texts, one type of relations has a significant impact, while in classifying large articles, the best results are obtained using all relations between thesaurus terms. This can be due to the fact that real journalistic texts in natural language use a much richer vocabulary and sentence structure, as opposed to short news. Therefore, the analysis of such texts requires a thesaurus with a large number of terms and relations between them.
Another important conclusion is that the influence of different types of relations between terms depends on a particular domain. This observation needs additional thorough research. Another direction for further research is to specify associative relations and analyze their influence to processing of large texts, journalistic, scientific, artistic, etc.

In any case, studies have shown that relations between terms are an essential part of the thesaurus as a model of the domain, and that a careful choice of the way of their use allows to solve problems of automatic text processing better.

ACKNOWLEDGMENTS

The authors would like to thank Natalia Kasatkina, associate professor, head of the Department of Foreign Languages of Humanities in P.G. Demidov Yaroslavl State University, for preparation of the corpus of articles about American immigrants.

FUNDING

This work was supported by the grant of the President of Russian Federation for state support of young Russian scientists (project MK-5456.2016.9).

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Masterman, M., Semantic message detection for machine translation, using an interlingua, *Proc. 1961 International Conf. on Machine Translation*, 1961, pp. 438–475.

2. Loukachevitch, N. and Dobrov, B., The Sociopolitical Thesaurus as a resource for automatic document processing in Russian, *Terminology*, 2015, vol. 21, no. 2, pp. 237–262.

3. Aitchison, J. and Clarke, S.D., The thesaurus: A historical viewpoint, with a look to the future, *Cataloging Classif. Q.*, 2004, vol. 37, nos. 3–4, pp. 5–21.

4. Loukashevich, N., *Tezaurusy v zadachah informatsionnogo poiska* (Thesauri in Information Retrieval Problems), Moscow: Mosk. Gos. Univ., 2011.

5. Willis, C. and Losee, R., A random walk on an ontology: Using thesaurus structure for automatic subject indexing, *J. Am. Soc. Inf. Sci. Technol.*, 2012, vol. 64, no. 7, pp. 1330–1344.

6. Vállez, M., Pedraza-Jiménez, R., Codina, L., Blanco, S., and Rovira, C., A semi-automatic indexing system based on embedded information in HTML documents, *Libr. Hi Tech*, 2015, vol. 33, no. 2, pp. 195–210.

7. Loukachevitch, N., Nokel, M., and Ivanov, K., Combining Thesaurus Knowledge and Probabilistic Topic Models, 2017. https://arxiv.org/abs/1707.09816.

8. Sanchez-Pi, N., Marti, L., and Garcia, A.C.B., Improving ontology-based text classification: An occupational health and security application, *J. Appl. Logic*, 2016, vol. 17, pp. 48–58.

9. Bollegala, D., Weir, D., and Carroll, J., Cross-domain sentiment classification using a sentiment sensitive thesaurus, *IEEE Trans. Knowl. Data Eng.*, 2013, vol. 25, no. 8, pp. 1719–1731.

10. Sparck Jones, K., Walker, S., and Robertson, S.E., A probabilistic model of information retrieval: Development and comparative experiments: Part 2, *Inf. Process. Manage.*, 2000, vol. 36, no. 6, pp. 809–840.

11. Lagutina, N.S., Lagutina, K.V., Mamedov, E.I., and Paramonov, I.V., Methodological aspects of semantic relation extraction for automatic thesaurus generation, *Model. Anal. Inf. Sist.*, 2016, vol. 23, no. 6, pp. 826–840.

12. Mihalcea, R. and Tarau, P., TextRank: Bringing order into texts, *Proceedings of Empirical Methods in Natural Language Processing—EMNLP*, Barcelona, 2004, pp. 404–411.

13. Trieschnigg, D., Pezik, P., Lee, V., De Jong, F., Kraaij, W., and Rebholz-Schuhmann, D., MeSH Up: Effective MeSH text classification for improved document retrieval, *Bioinformatics*, 2009, vol. 25, no. 11, pp. 1412–1418.

14. Aggarwal, C. and Zhai, C., A survey of text classification algorithms, in *Mining Text Data*, New York: Springer-Verlag, 2012, pp. 163–222.

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

16. Ravi, K. and Ravi, V., A survey on opinion mining and sentiment analysis: Tasks, approaches and applications, *Knowl.-Based Syst.*, 2015, vol. 89, pp. 14–46.

17. Junker, M., Hoeh, R., and Dengel, A., On the evaluation of document analysis components by recall, precision, and accuracy, *Proceedings of the Fifth International Conference on Document Analysis and Recognition, IEEE*, 1999, pp. 713–716.