

## RESEARCH PAPER

# Application of Natural Language Processing Algorithms to the Task of Automatic Classification of Russian Scientific Texts

Aleksandr Romanov<sup>1</sup>, Konstantin Lomotin<sup>2</sup> and Ekaterina Kozlova<sup>2</sup><sup>1</sup> Moscow Institute of Electronics and Mathematics. National Research University Higher School of Economics, Moscow, RU<sup>2</sup> Faculty of Computer Science, National Research University Higher School of Economics, Moscow, RUCorresponding author: Konstantin Lomotin ([ke.lomotin@gmail.com](mailto:ke.lomotin@gmail.com))

This work is devoted to the study of applicability of modern methods of machine learning to the task of automatic classification of scientific articles and abstracts. For this purpose, the study of such models of machine learning as artificial neural networks, random forest, logistic regression, and support vector machine was carried out with taking into account such a feature of scientific texts as a large number of terms specific for various categories. Separately, the stages of data collection and extraction of text characteristics are considered. The results of research are used in development of a decision support system for assignment of scientific texts to the code of the department or abstract journal of All-Russian Institute of Scientific and Technical Information of Russian Academy of Sciences.

**Keywords:** text classification; machine learning; artificial neural network; logistic regression; random forest; support vector machine; natural language processing

## Introduction

The problem of automatic classification of texts is becoming increasingly required due to the growing amount of textual information stored on the Internet. In order to be able to meet this challenge, algorithms of machine learning (such as supervised learning algorithms) are applied. For their setting, they require a set of marked data already having a class label.

This study is aimed at developing a model capable to determine the probability of text belonging to a category of a certain rubricator, i.e. to work in a Decision Support System (DSS) mode. The work is carried out as part of development of a text analysis system for All-Russian Institute of Scientific and Technical Information of Russian Academy of Sciences (VINITI RAS) (Viniti.ru, 2019). One of the objectives of the Institute is the collection and storage of scientific publications from around the world. Documents go through thematic departments, where specialists assign them codes of topics in various systems of classification. In this case, the number of codes of abstract journals and State Rubricator of Scientific and Technical Information (SRSTI) reaches several hundred. The use of DSS is intended to reduce the number of possible topics for the text providing the specialist with an estimate of the probabilities for each rubric.

The work reviews existing software products, designed to solve this problem; the training data, provided by VINITI RAS for process of tuning algorithms, is described; two different approaches to interpretation of the problem are compared, as well as methods for performing experiments and quality metrics calculation are revealed; analysis of obtained classification results by codes of thematic departments, codes of abstract journals, and SRSTI is performed.

## Review of existing solutions

### *Software products*

There are a number of tools and software products that allow classification of texts. Consider the key ones for applicability for solving classification problems at VINITI RAS.

ABBYY Smart Classifier (Automatic Content Classification with ABBYY Solutions, 2019) is a text analysis tool developed by ABBYY. The functionality of software includes classification of an arbitrary rubricator, semantic analysis of the text, as well as many auxiliary functions. The stated objective of this product is to simplify electronic workflow and to automate business processes. From the point of view of software implementation, Smart Classifier SDK requires a lot of computing resources (a 64-bit 4-core processor with a clock speed of 2 GHz or higher), as well as a large amount of memory (2 GB of RAM for each processor core), which does not meet VINITI performance requirements (Intel Core i5 3GHz, 4Gb RAM). Moreover, to solve the problems facing VINITI RAS, the functionality of this software product is redundant.

IBM Watson Natural Language Classifier (Watson Natural Language Classifier, 2019) is a cloud platform that allows using processing power of IBM Watson supercomputer to solve the task of text classification. The system does not allow training of user data models. Nevertheless, available classification modules have the ability to determine a sufficient set of topics to project them into departmental codes or abstract journals. It is unlikely that such an approach will make sense for SRSTI codes because of their number (more than 8 thousand for the full tree of the rubricator). In addition, the application requires a connection to the Internet. This solution does not meet the requirements for information security, and it is not suitable for deploying the system on secure servers.

LPU tool (Learning from Positive and Unlabeled data tool) (LPU download page, 2019), developed at the University of Illinois at Chicago, allows using EM algorithm and the method of reference vectors for binary classification of text on an arbitrary rubricator. To train the system, it is necessary to prepare a training set consisting of training vectors. The advantages of the product include independence of language of the text and method of characteristic extraction. Nevertheless, the functionality of this tool is too narrow for performing the task of multiclass classification at VINITI RAS. The classifier is free, but its source code is non-open. For this reason, this software solution cannot be extended by such prospective models as convolutional and recurrent neural networks. In addition, the application supports binary classification only.

Thus, the existing solutions do not meet the requirements for solving classification problems at VINITI RAS which means the need to develop a new software.

### ***Preprocessing***

Natural language text in the original form is difficult to process automatically. In order to bring the text into a form suitable for further analysis, various kinds of preprocessing are used (see the description given in the article by Uysal (Uysal and Gunal, 2014)). The paper considers the influence of four independent aspects of text preprocessing (tokenization, stop words removal, lowercasing and stemming) on the classification result. The training set was formed by two different corpora of texts, each of which was presented in English and Turkish. After that, a comparison of the results of classification of texts, subjected to all possible combinations of these types of preprocessing, was made; in this work, the support vector machine (SVM) was used.

A.K. Uysal found that lowercasing significantly increases the quality of classification according to F1-score, regardless of the language and subject matter of the document. For texts in English, the most appropriate combination of types of preprocessing was tokenization, stemming and lowercasing without stop words removing.

Results of the above work are of great importance for this study, since they are basic elements for the development of a pre-processor for scientific texts. Nevertheless, in (Uysal and Gunal, 2014) only one algorithm of machine learning is considered as a classifier, whereas for other algorithms, the optimal combination of preprocessing types can be different.

In work (Goncalves and Quaresma, 2018) "bag-of-words" model, applied to two sets of English texts, ensured that a classification quality of 0.7 to 0.9 according to F1-score was achieved. The results of this study suggest that for different types of texts, different types of preprocessing can imply different classification quality. Nevertheless, the article does not cover how the quality depends on the type of text and preprocessing.

### ***Extraction of features from textual information***

Algorithms of supervised machine learning can work with numerical vectors, but not with natural language texts. At present, many researchers are exploring different approaches to extraction of characteristics from textual information. Lately, semantic model word2vec, based on neural network technologies, has been used. A number of recent studies have demonstrated the advantage of word2vec in comparison with previously used statistical approaches (for example, when it used in tandem with LSTM networks (Semberecki and Maciejewski, 2017)), although in another recent study (Wang et al. 2017), the authors failed to demonstrate

experimentally significant advantage of the semantic approach (as compared to statistical one) in experiments on the classification of texts with different number of class labels. Despite this, technology word2vec is considered to be a promising area for research, being actively developed over the past few years.

### **Classification**

#### **Method of centroids and k-nearest neighbors (k-NN) algorithm**

The resulting numerical vectors should be used for training and testing the classifier. By now, a number of studies have demonstrated the effectiveness of application of different classification models to texts in natural languages. One of important trends in recent studies of text classification is the experiment with algorithms, based on the use of centroids, which are considered to be promising for multiclass classification. In work (Lex et al. 2010), a method of centroids was studied on texts with an average length of 92.5 nouns which is close to the average length of texts in VINITI RAS set and, therefore, gives the results significant for this study. The classifier, based on k-NN algorithm, also uses similar algorithms (Lex et al. 2010; Sammouda, 2017). However, the researchers were unable to demonstrate the significant advantages of both of these approaches in the area of text classification.

As in the case of other metric classifiers, there is also the problem of choosing a metric in the feature space when using the K-nearest-neighbor method. In the feature space of Word2Vec, the Word Mover's Distance (WMD) metric was proposed (Kusner et al. 2015). Based on the research results, it can be concluded that this metric algorithm is able to show high results in the case of using a complex distance metric, but this raises a performance problem. According to the results, obtained by (Wu et al. 2018), the speed of the metric classifier, based on k-NN, significantly (from 2.5 to 150 times) depends on the choice of metrics and embedding. Thus, when solving the problem of classification in the environment of a well-established corporate infrastructure, such a classifier requires great care when using. The performance and quality of the k-NN algorithm require further research in the context of the specifics of the work of VINITI RAS.

#### **Support vector machine**

SVM is often used to solve the classification problem. In the studies, devoted to this algorithm (Bin Xu and Yufeng Zhang, 2011; Liu et al. 2017), attention is paid to its main advantage – the possibility of constructing a nonlinear separating surface with the help of kernels. In (Bin Xu and Yufeng Zhang, 2011) a kernel, with taking into account semantic characteristics of the text, is defined. The authors achieved the quality of classification up to 0.96 by F-score for individual topics, which is an extremely high indicator. The usage of texts in Chinese, however, makes this approach complicated for the current study.

The results of studies on the application of SVM to the classification of texts in Russian (Sokolova and Bobicev, 2009; Yussupova et al. 2012) suggest that when using simple linear or polynomial kernels, the quality of the SVM classifier is likely to be lower than the quality of k-NN or decision-based classifiers. During the experiments conducted in the course of this work, it was found that training the classifiers with more complex cores requires significantly more computational resources.

#### **Artificial neural networks**

In works, devoted to neural networks usage for text classification (Chen et al. 2017; Du, 2017), emphasis is placed on the effectiveness of recurrent neural networks. In particular, the architecture of LSTM (Du, 2017) is considered as a promising choice for designing a neural network for solving this problem. Thus, for the texts in English and Chinese, processed using Word2Vec technology, or the TF-IDF algorithm, the most effective were convolutional recurrent neural networks. According to N.V. Vorobyov and E.V. Puchkov (Vorobyov and Puchkov, 2017), for the Russian language, convolutional network architecture turned out to be more suitable one, in case the features are extracted by word2vec tool. The results, obtained by researchers in the field of neural networks, make it possible to significantly reduce the range of neural network structures applicable to the problem being solved. On the other hand, the authors of the studies examined used a certain neural network structure in their studies, not selecting the optimal structure. Potentially, such experiments could increase text classification accuracy.

In (Joulin et al. 2016) paper classifier models are considered separately from feature extraction methods. Relatively high classification quality (67.6 % of correct answers) was achieved using the classifier based on the LSTM recurrent neural network compared to the support vector method and the convolutional neural network.

The authors of the study (Wang et al. 2016) describe a combined model that uses the two most promising architectures of neural networks: convolutional and recurrent networks – in the task of determining

tonality. In the described work, the text is presented in the form of a matrix of sentences, where each column is a vector of features of a single word obtained using Word2Vec. The matrix is padded with zeros to achieve a given dimension (since the number of words can vary from sentence to sentence), then convolutional layers in alternation with layers of pooling form a high-level representation vector that is processed by the LSTM layer and fully connected layers. The authors managed to achieve a quality classification in 0.9 by accuracy metric (the proportion of correct classifier answers). This result exceeds the quality of a separate LSTM classifier by 6% and a separate classifier based on a convolutional neural network – by 4%. It was decided to abandon the use of this combined model, since its training time doesn't fit the requirements of VINITI RAS.

The LSTM network is considered in comparison with the support vector method and the convolutional neural network in (Tang et al. 2015). Despite the fact that the purpose of the study was to develop a method for modeling sentences using GRNN (Gated Recurrent Neural Network), the authors also reviewed and compared the most promising classifiers based on the model of proposals that they developed. In all the considered data sets, the LSTM classifier showed the highest proportion of correct answers when testing.

Thus, the recurrent neural network with fully connected output layers was chosen as a classifier model. In many works, this architecture is considered as the best choice for the text classification task which does not require large computational resources for learning, unlike classifiers based on deep convolutional neural networks.

### Naive Bayesian Classifier

In work (RAJU et al. 2017), naive Bayesian classifier is considered to be a quick and easy-to-implement algorithm which, nevertheless, shows low results with a high correlation between the input data features. The reason for this problem is the assumption of independence of words among themselves underlying naive Bayesian approach. This assumption turns out to be incorrect when a strong correlation appears between the occurrences of words in the set. The work states that in most cases, naive Bayesian classifier allows fast categorization of texts with sufficient quality. Nevertheless, this statement is refuted by other scientists. In (Wang et al. 2017), naive Bayesian classifier demonstrated the worst classification quality. In another study (Bourgonje et al. 2018), the relatively high quality of this model was observed only with a relatively large length of texts.

In work (Yussupova et al. 2012) in the context of a similar task, the NB classifier is considered as a model that is inferior in quality to the SVM classifier and k-NN. The weak point of this probabilistic model is a restriction on the values of the features – they must be non-negative, whereas in Word2Vec feature space, word vectors can have arbitrary meanings.

### Boosting algorithms

Recently, the use of boosting algorithms to improve the results of simple classifiers also enjoys interest in academic environment. A number of studies have confirmed the higher efficiency of weighted (Sun et al. 2007) and gradient boosting (Dimov et al. 2017) in relation to other models. Boosting algorithms are successfully applied in various fields of science, such as Biology (Feng et al. 2005), Robotics (Luo et al. 2017), Agricultural Sciences (Dimov et al. 2017), etc. In many cases, ensemble algorithms have shown the results which were higher than SVM, multilayer neural network, and k-NN algorithm. Nevertheless, recent studies have shown that this does not always lead to an improvement in quality when applied to natural language texts (Abuhaiba and Dawoud, 2017), apart from requiring significant computational resources.

## Discussion

### ***Influence of the length of text on the quality of classification***

It was noted that the average length of texts affects the result of classification. In (Bourgonje et al. 2018) the authors deal with publications in social network Twitter and articles from Wikipedia with an average length of 18 and 65 words respectively. As a result of experiments, it became obvious that different classifiers give best result for different average length of texts. In experiments with short texts, logistic regression showed better results for almost all quality metrics, whereas on long texts, naive Bayesian classifier always showed the highest results.

### ***Influence of number of classes on the quality of classification***

It was experimentally proved that number of classes influences the results of classification (Wang et al. 2017). In this study, attention was focused on the differences in classification results for classes of large and small size with number of labels of class 59 and 8, respectively. The strong point of this study is the achieved balance of distribution of texts by classes, which allows more accurately assessing the applicability

of the results to real data. Moreover, the results were compared using different classification models and approaches to representation of text in the form of a vector. The best result was demonstrated by logistic regression; SVM proved to be a little worse, and naive Bayesian classifier turned out to be the worst algorithm. It is also shown that in case of a small number of classes, the difference in the results in experiments with different approaches for representing the text in the form of a vector is much lower than in case of classes of a large size.

Methods of text preprocessing, considered in (Goncalves and Quaresma, 2018; Semberecki and Maciejewski, 2017), showed their effectiveness in case when it is necessary to train the model for classification of the English text. Nevertheless, replacement of stemming to lemmatization when working with the Russian language can significantly improve the quality of classification, since it is much easier to conduct POS-tagging for lemmatization of Russian words than for lemmatization of English words.

In works (Lex et al. 2010; Wang et al. 2017), devoted to extraction of features from text data, the researchers found that the use of statistical methods, based on word frequency metrics in the text corpus, makes it possible to achieve a higher quality of the classifier work in comparison with the use of advanced word2vec technology. However, this tool has a large number of parameters, affecting the vector representation of the texts it generates, whereas statistical methods are not easily tunable. In this work, experiments in selection of the most suitable word2vec parameters are conducted.

Based on works (Abuhaiba and Dawoud, 2017; Bourgonje et al. 2018; Liu et al. 2017; Semberecki and Maciejewski, 2017), it is possible to identify the models of machine learning that are most suitable for classification of textual data. Such models are: logistic regression, random forest, SVM, and artificial neural network (both feedforward and LSTM). In this paper, the results of experiments on the creation of a classifier, working with texts in Russian, subjected to lemmatization and transferred to the vector space of features by using the word2vec tool, are given. Conclusions, drawn from the results of the research, allow choosing word2vec parameters and classification algorithm configuration to create a system capable of high quality processing of the flow of scientific texts presented in Russian.

Thus, the results of studies, dedicated to application of various models of machine learning to the task of text classification, can be useful at the stage of algorithm selection. Nevertheless, the authors of the works considered only the theoretical side of the problem whereas in the real flow of input texts (emails, news articles, etc.) different results of algorithm implementation can be expected. This study considers the use of described algorithms for solving an applied problem.

## Training data analysis

For training the VINITI RAS classifiers, a marked set of texts in Russian, including the title of publication, a short annotation by an expert in the relevant field of science, and a list of key words, was provided. The set size was 143000 texts, 95000 of which were used directly for model training, and the remaining 48000 texts were used for testing. An example of the record is given in **Table 1**.

The texts contain authors' annotations and expert abstracts to full-text scientific articles. The average text length is approximately 120 words; it varies slightly for different categories.

Classification was carried out in three rubricators: codes of thematic departments, codes of abstract journals and the second level of the SRSTI hierarchy. The rubricators vary considerably in number of topics. The simplest one for automatic classification is "department codes" rubricator which includes 15 topics. Significantly more complex is the classification of 237 codes of abstract journals. The most narrowly specialized rubricator was SRSTI that was tasked to determine the first five symbols of the code; one of them is the separating point which corresponds to the second level of rubricator. The documents, constituting the training set, represented 449 codes of SRSTI.

The diagram in **Figure 1** shows how many classes, marked on the Y-axis for each of the three rubricators, include not less than the specified number of texts marked on the X-axis.

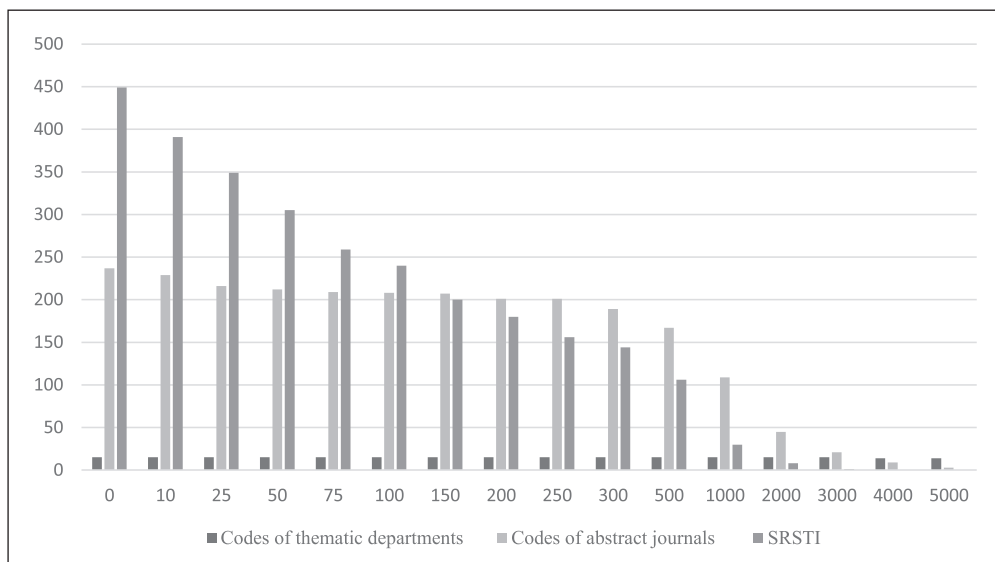
Statistics for department codes is almost unchanged, as in all 15 topics, there are at least 3000 texts, and in 14 of them, there are no less than 5000 texts. This is a good indicator of uniform distribution of texts in the topics that allows the classifier to learn to recognize each class well.

The number of codes of abstract journals practically does not change up to the mark of 250 texts (201 topics); at the mark of 1000 documents, the number of codes falls almost twice (46% of the total number of topics of abstract journals). 29 topics have less than 100 texts which can negatively affect the quality of recognition of these categories by the classifier. In addition, 21 of 237 topics have more than 3000 texts, which are at least 63,000 texts and about two-thirds of the entire set. As a result, it is possible to expect overfitting of the classifier on the topics with a large number of classes and a high proportion of correct answers.



**Table 1:** Sample text from the dataset.

Field	Value
Text ID	B1111259224
Title	Изучение трематодофауны инвазивных видов моллюсков на территории Беларуси
Body	В результате проведенных исследований и анализа литературных данных в водоемах Беларуси у инвазивных видов моллюсков выявлено как минимум 14 представителей класса Trematoda: у <i>D. polymorpha</i> - <i>Phyllodistomum folium</i> ...
Keywords	Республика Беларусь \ видовое разнообразие \ диагностика \ пресноводные моллюски \ таксономия \ трематоды \ фауна \ хозяева
Codes of thematic departments	e3\e4
Codes of abstract journals	04AND9\07Д
SRSTI	341.33.23.17.11.09\391.19.25.31

**Figure 1:** Dependence of number of topics of three rubricators on minimum number of texts.

Statistics on SRSTI codes demonstrates a stable reduction in the number of topics with a given minimum number of texts. In total, there are 449 topics in this rubricator, while less than half have 150 or more texts, and only 30 topics have more than 1000 texts (only 7% of SRSTI topics). At least 50 texts have about 106 topics (24% of the total number; it is not less than 53,000 texts and 56% of the set). At the same time, exactly 100 topics (22% of their total number) have less than 25 texts, and it is not representative enough for training the classifier to recognize such topics. From this, it can be concluded that the set is extremely unbalanced, and as a result, the quality of the trained classifier may be low. Nevertheless, the accuracy may be quite high, since a significant number of texts belong to just a few of the topics the recognition of which the classifier can perform well.

## Methods of conducting experiments

### Preprocessing of data

The purpose of preprocessing the text in this work is to obtain the “bag of words” (Lex et al. 2010) from the input text. This approach is frequently applied with methods of feature extraction based on the frequency of occurrence, measure of importance, or context of word in the text.

Primary text processing includes several stages:

1. Parsing of VINITI RAS internal document markup, removal of formulas.
2. Combining title, text of the annotation and keywords into one text.

3. Lowercasing.
4. Removing common stop words (prepositions, conjunctions, interjections, suffixes, etc.).
5. Lemmatization of words.

To store formulas, indexes, and other markup elements in the texts, VINITI RAS information system uses its own markup language. The accepted markup standard allows using TEX language. Extracting plain text is the first stage of preprocessing.

The method, used to extract characteristics, does not take into account the case of text. For this reason, the text is changed to lowercase to simplify further processing.

Stop words in Russian texts do not carry such a semantic load, as in English texts (Uysal and Gunal, 2014). In this study, when preparing data for training, stop words were deleted.

In the Russian language, practically all notional words can change according to a set of parameters, such as tense, gender, case, etc. Despite the fact that word forms cause a large semantic load, the stage of lemmatization (Lapach and Radchenko, 2012) was included in the preprocessing (transforming the word into its normal form). This decision is justified by a phenomenon that the applied method of extracting characteristics is capable to identify semantic features of words only on the basis of context, and one word in its different forms is considered to be different words. This leads to the fact that the same word can have several vector representations.

An alternative to lemmatization in the formation of a “bag of words” is stemming (Korenus et al. 2004). Stemming algorithms show a higher speed of operation which is a useful feature when implemented in the developed software. As a rule, lemmatization allows achieving the same or higher quality classification as compared with stemming algorithms (Toman et al. 2006).

Thus, the result of text preprocessing is a sequence of notional words in their normal forms in lowercase.

### ***Feature extraction***

Classification models, used in this study, are not capable to work directly with textual information; for their correct use, it is required to create (on the basis of texts) vectors that encode information in natural language into numerical values (Géron, 2018). The most promising approaches to forming such vectors can be combined into two groups: statistical and semantic ones. Statistical approaches, such as measure of word frequency, measure of “weirdness” (Klyshinsky and Kochetkova, 2015) – and TF-IDF itself proved to be extremely expensive for memory and computing resources, and for this reason, they will not be considered in this study. The most promising and modern approaches are semantic ones. In this paper, contextual semantic model word2vec, based on neural network technologies, is used. It should be said that this model does not require large computational resources and special data markup (gensim: models.word2vec – Word2vec embeddings, 2019).

Word2vec uses large amounts of text information to identify semantic links and create a vector representation of words in a space of a given dimension. The dimension of word2vec features is specified during training and determines the number of semantic features that it can select in the word. Each word is a point in the space of this dimension, and its synonyms are nearby. Moreover, in the presence of an example with a certain semantic connection (for example, antonymy), one can use a model to find a word with a similar connection for a given connection (search for an antonym). With sufficient dimensionality of space, the model creates a detailed representation of word relations and their semantics. Selected semantic features cannot be interpreted by a person, as a rule.

Word2vec forms a vector of semantic features based on the word and its context in the sentence. The text can be converted into  $n \times m$  sized matrix, where  $n$  – number of words in the text,  $m$  – dimension of word2vec vectors. The number of words in the texts varies, and the dimensions of matrices may differ. Different length of texts leads to the problem of unification of dimensions of input data. In this study, methods of pooling (Scherer et al. 2010), which make it possible to obtain a vector of dimension  $m$ , are used. These methods are intuitively understandable and consist of summing, averaging, or selecting the maximum element within one feature – i.e., column of the matrix.

It was observed that dimension  $m$  of word2vec vectors for a given task also affects the quality of classification. To establish the most suitable dimension, experiments with vectors, containing 50, 100, and 500 features, were carried out (Tables 2–4). The models considered in this paper include logistic regression (LR), random forest (RF), feedforward artificial neural network with 1 (ANN1) and 2 (ANN2) hidden layers, support vector machine (SVM) and Recurrent ANN with LSTM layer (LSTM). In this research not all models were studied when choosing methods for extracting features due to the limitations of computational resources.

**Table 2:** Results of selection of method for extraction of features for vectors with 50 elements.

Classification model	Averaging	Average			Maximum			Sum					
		Accuracy	Precision	Recall	F-score	Accuracy	Precision	Recall	F-score	Accuracy	Precision	Recall	F-score
LR	Micro	0.94	0.75	0.54	<b>0.63</b>	0.92	0.60	0.40	0.48	0.94	0.77	0.53	0.62
	Macro	0.94	0.67	0.51	0.55	0.92	0.54	0.38	0.43	0.94	0.72	0.51	<b>0.58</b>
RF	Micro	0.94	0.75	0.53	<b>0.62</b>	0.92	0.60	0.39	0.48	0.93	0.72	0.50	0.59
	Macro	0.94	0.75	0.43	<b>0.51</b>	0.92	0.60	0.27	0.29	0.93	0.73	0.39	0.45
ANN1	Micro	0.94	0.79	0.53	0.63	0.92	0.63	0.44	0.52	0.94	0.80	0.57	<b>0.67</b>
	Macro	0.94	0.79	0.44	0.52	0.92	0.57	0.35	0.41	0.94	0.78	0.51	<b>0.60</b>
ANN2	Micro	0.94	0.78	0.54	0.64	0.92	0.62	0.43	0.51	0.94	0.80	0.56	<b>0.66</b>
	Macro	0.94	0.77	0.46	0.54	0.92	0.59	0.33	0.38	0.94	0.78	0.51	<b>0.60</b>

**Table 3:** Results of selection of method for extraction of features for vectors with 100 elements.

Classification model	Averaging	Average			Maximum			Sum					
		Accuracy	Precision	Recall	F-score	Accuracy	Precision	Recall	F-score	Accuracy	Precision	Recall	F-score
LR	Micro	0.94	0.53	0.57	0.55	0.93	0.50	0.50	0.50	0.94	0.62	0.62	<b>0.62</b>
	Macro	0.94	0.53	0.57	0.55	0.93	0.50	0.50	0.50	0.94	0.62	0.62	<b>0.62</b>
RF	Micro	0.94	0.47	0.54	0.49	0.93	0.44	0.47	0.44	0.94	0.56	0.59	<b>0.57</b>
	Macro	0.94	0.53	0.49	0.51	0.92	0.47	0.47	0.47	0.94	0.56	0.56	<b>0.56</b>
ANN1	Micro	0.94	0.47	0.39	0.41	0.92	0.42	0.35	0.36	0.94	0.52	0.46	<b>0.47</b>
	Macro	0.95	0.57	0.55	0.56	0.93	0.53	0.53	0.53	0.95	0.64	0.64	<b>0.64</b>
ANN 2	Micro	0.95	0.52	0.42	0.42	0.93	0.49	0.42	0.43	0.95	0.60	0.56	<b>0.57</b>
	Macro	0.95	0.57	0.55	0.56	0.93	0.53	0.53	0.53	0.95	0.64	0.64	<b>0.64</b>



**Table 4:** Results of selection of method for extraction of features for vectors with 500 elements.

Classification model	Averaging	Average			Maximum			Sum				
		Accuracy	Precision	Recall	F-score	Accuracy	Recall	F-score	Accuracy	Precision	Recall	F-score
LR	Micro	0.95	0.64	0.64	<b>0.64</b>	0.94	0.58	0.58	0.58	0.95	0.64	<b>0.64</b>
	Macro	0.95	0.58	0.61	0.59	0.94	0.52	0.55	0.52	0.95	0.59	<b>0.60</b>
RF	Micro	0.94	0.57	0.57	<b>0.57</b>	0.93	0.49	0.49	0.49	0.94	0.56	0.56
	Macro	0.94	0.53	0.47	<b>0.48</b>	0.93	0.44	0.38	0.38	0.94	0.53	<b>0.48</b>
ANN1	Micro	0.95	0.62	0.62	0.62	0.94	0.57	0.57	0.57	0.95	0.64	<b>0.64</b>
	Macro	0.95	0.59	0.52	0.53	0.94	0.57	0.45	0.46	0.95	0.61	<b>0.57</b>
ANN 2	Micro	0.95	0.62	0.62	0.62	0.94	0.58	0.58	0.58	0.95	0.64	<b>0.64</b>
	Macro	0.95	0.59	0.52	0.53	0.94	0.55	0.47	0.47	0.95	0.61	<b>0.57</b>

In almost all cases, textual matrix pooling over the sum of features values showed best results for the F-score. The exception was made by two models with 50 features in the vector and by one – with 500 features. But the quality of recognition, performed by these models, is relatively low in comparison with those in which the best recognition is achieved when summing the values of characteristics. Pooling by average value of features is further used in training of classifiers. At the same time, improving the quality of classification with increasing the number of features in vectors is not observed, and computational complexity of the problem increases. For this reason, vectors that contain 50 characteristics are used.

The relationship between features of the objects under study can cause a significant reduction in the quality of classification (Toloşi and Lengauer, 2011).

### Classifier settings

#### *Logistic regression*

For logistic regression, a series of experiments to select an algorithm for optimizing the loss functions and regularization parameter was performed. The model always showed better results at high values of regularization coefficient ( $C = 10$ ) preventing the model from overfitting.

#### *Random forest*

In this study, minimum number of objects in the sheet was selected, as well as minimum number of objects in the partition, maximum number of features used, and maximum depth of the tree. In this case, a forest of 10 decision trees was used. The optimal model is the one with a depth of 60, using 15 or 30 features when learning for different rubricators; this model uses the entropy criterion for partitioning as well.

#### *Artificial neural network*

Among the extensive class of neural network models, two classes were used for conducting experiments: dense forward propagation layers, and recurrent structures with LSTM layers. The highest quality of classification was shown by a multilayer neural network with one and two layers in a hidden layer, as well as by a recurrent neural network with LSTM input layer and a layer of classical neurons with a logistic activation function at the output.

LSTM classifiers have recently become the most popular model for processing textual information, but most often they are used for embedded words. This paper tests the assumption that the use of a recurrent architecture at the text scale will help the model take into account the specifics of the sample set.

As a result of the choice of the model, based on the search strategy, the optimal number of neurons in the hidden layers of neural network turned out to be 50–55.

#### *Support vector machine*

The strong point of SVM is the possibility of a non-linear transformation of the feature space by means of a transition from scalar products to arbitrary kernels. In this work, the application of a linear kernel and the one, based on a radial-basis function, were studied. Models, using the radial-basis function, always show better results, but the optimization algorithm for this model has greater computational complexity.

## Analysis of results

The results of the experiments suggest that classifiers, based on SVM and a recurrent neural network, most efficiently solve the problem of classifying short scientific texts. Let's consider the results in more detail for different size recommendation classifier.

**Tables 5–7** show results of experiments on training of classifiers for three classification systems: codes of VINITI RAS thematic departments, codes of abstract journals, and SRSTI.

The tables summarize the results of work of the classifiers for all the rubricators on several quality metrics. The most informative in this case is F-score since the sets are unbalanced by the number of texts per topic, and the share of correct responses practically does not take into account the topics with a small number of texts. Precision and recall are also not indicative because errors and correct responses are of equal importance. Based on the above, the main indicator of quality being focused on is F-score. Since metrics are calculated for each class of all the rubricators separately, and complete tables are not possible, micro- and macro-averaging is used to obtain the general picture (Clark et al. 2003). Micro-averaging often gives higher quality metrics, but is only comparative, and as a result, it is impossible to assert whether any of these approaches is correct.

**Table 5:** Results of testing classifiers for codes of thematic departments.

Classifier	Averaging	Accuracy			Precision			Recall			F-score		
		1 response	2 responses	3 responses	1 response	2 responses	3 responses	1 response	2 responses	3 responses	1 response	2 responses	3 responses
LR	Micro	0.94	0.93	0.90	0.77	0.59	0.49	0.53	0.71	0.80	0.62	<b>0.64</b>	0.60
	Macro	0.94	0.93	0.90	0.72	0.55	0.45	0.51	0.70	0.79	0.58	<b>0.60</b>	0.56
RF	Micro	0.93	0.92	0.88	0.72	0.55	0.43	0.50	0.67	0.78	0.59	<b>0.60</b>	0.55
	Macro	0.93	0.92	0.88	0.73	0.55	0.42	0.39	0.57	0.70	0.50	<b>0.53</b>	0.51
ANN1	Micro	0.94	0.93	0.90	0.80	0.60	0.47	0.57	0.76	0.85	<b>0.67</b>	<b>0.67</b>	0.60
	Macro	0.94	0.93	0.90	0.78	0.58	0.45	0.51	0.71	0.82	0.60	<b>0.63</b>	0.57
ANN 2	Micro	0.94	0.93	0.90	0.80	0.61	0.48	0.56	0.75	0.85	0.66	<b>0.68</b>	0.61
	Macro	0.94	0.93	0.90	0.78	0.59	0.46	0.51	0.71	0.82	0.60	<b>0.64</b>	0.58
SVM	Micro	0.95	0.94	0.91	0.82	0.64	0.50	0.59	0.77	0.87	0.69	<b>0.70</b>	0.64
	Macro	0.95	0.94	0.91	0.80	0.61	0.48	0.55	0.74	0.85	0.65	<b>0.67</b>	0.61
LSTM	Micro	0.95	0.91	0.86	0.80	0.52	0.39	0.60	0.78	0.87	<b>0.68</b>	0.63	0.54
	Macro	0.95	0.91	0.86	0.77	0.49	0.37	0.54	0.75	0.85	<b>0.63</b>	0.59	0.50

**Table 6:** Results of testing classifiers for codes of abstract journals.

Classifier	Averaging	Accuracy			Precision			Recall			F-score		
		1	2	3	1	2	3	1	2	3	1	2	3
LR	Micro	0.99	0.99	0.98	0.49	0.36	0.29	0.33	0.49	0.59	0.39	0.39	0.39
	Macro	0.99	0.99	0.98	0.46	0.36	0.29	0.35	0.51	0.60	0.37	0.40	0.37
RF	Micro	0.99	0.99	0.98	0.45	0.35	0.29	0.23	0.39	0.49	0.23	0.36	<b>0.37</b>
	Macro	0.99	0.99	0.98	0.36	0.31	0.26	0.20	0.33	0.42	<b>0.31</b>	0.30	0.30
ANN1	Micro	0.99	0.99	0.98	0.47	0.37	0.30	0.24	0.40	0.51	0.32	<b>0.39</b>	0.38
	Macro	0.99	0.99	0.98	0.41	0.34	0.28	0.20	0.34	0.43	0.23	0.31	<b>0.32</b>
ANN 2	Micro	0.99	0.99	0.98	0.46	0.36	0.30	0.25	0.41	0.52	0.32	<b>0.39</b>	0.38
	Macro	0.99	0.99	0.98	0.40	0.33	0.28	0.22	0.35	0.45	0.25	<b>0.32</b>	<b>0.32</b>
SVM	Micro	0.99	0.99	0.99	0.61	0.48	0.38	0.36	0.54	0.65	0.45	<b>0.51</b>	0.48
	Macro	0.99	0.99	0.99	0.54	0.44	0.36	0.33	0.50	0.60	0.40	<b>0.46</b>	0.44
LSTM	Micro	0.99	0.98	0.98	0.49	0.36	0.28	0.33	0.50	0.59	0.39	<b>0.42</b>	0.38
	Macro	0.99	0.98	0.98	0.47	0.36	0.28	0.34	0.49	0.59	0.37	<b>0.40</b>	0.37

**Table 7:** Results of testing classifiers for SRSTI codes.

Classifier	Averaging	Accuracy						Precision						Recall						F-score							
		1		2		3		1		2		3		1		2		3		1		2		3			
		response	responses	response	responses	response	responses	response	responses	response	responses	response	responses	response	responses	response	responses	response	responses	response	responses	response	responses	response	responses		
LR	Micro	0.99	0.99	0.99	0.99	0.41	0.31	0.25	0.28	0.43	0.53	0.33	0.36	0.34	0.99	0.99	0.99	0.99	0.41	0.31	0.25	0.28	0.43	0.53	0.33	0.36	0.34
	Macro	0.99	0.99	0.99	0.99	0.29	0.23	0.19	0.29	0.42	0.50	0.24	0.25	0.24	0.99	0.99	0.99	0.99	0.29	0.23	0.19	0.29	0.42	0.50	0.24	0.25	0.24
RF	Micro	0.99	0.99	0.99	0.99	0.43	0.32	0.26	0.19	0.34	0.44	0.27	0.33	0.32	0.99	0.99	0.99	0.99	0.43	0.32	0.26	0.19	0.34	0.44	0.27	0.33	0.32
	Macro	0.99	0.99	0.99	0.99	0.15	0.16	0.14	0.06	0.12	0.17	0.06	0.11	0.13	0.99	0.99	0.99	0.99	0.15	0.16	0.14	0.06	0.12	0.17	0.06	0.11	0.13
ANN1	Micro	0.99	0.99	0.99	0.99	0.46	0.35	0.28	0.25	0.40	0.49	0.32	0.38	0.36	0.99	0.99	0.99	0.99	0.46	0.35	0.28	0.25	0.40	0.49	0.32	0.38	0.36
	Macro	0.99	0.99	0.99	0.99	0.16	0.14	0.13	0.08	0.14	0.18	0.09	0.12	0.13	0.99	0.99	0.99	0.99	0.16	0.14	0.13	0.08	0.14	0.18	0.09	0.12	0.13
ANN 2	Micro	0.99	0.99	0.99	0.99	0.46	0.36	0.29	0.27	0.42	0.52	0.34	0.38	0.37	0.99	0.99	0.99	0.99	0.46	0.36	0.29	0.27	0.42	0.52	0.34	0.38	0.37
	Macro	0.99	0.99	0.99	0.99	0.19	0.18	0.16	0.09	0.16	0.22	0.11	0.15	0.16	0.99	0.99	0.99	0.99	0.19	0.18	0.16	0.09	0.16	0.22	0.11	0.15	0.16
SVM	Micro	0.99	0.99	0.99	0.99	0.62	0.46	0.36	0.37	0.55	0.65	0.46	0.51	0.47	0.99	0.99	0.99	0.99	0.62	0.46	0.36	0.37	0.55	0.65	0.46	0.51	0.47
	Macro	0.99	0.99	0.99	0.99	0.41	0.35	0.28	0.20	0.33	0.42	0.25	0.32	0.32	0.99	0.99	0.99	0.99	0.41	0.35	0.28	0.20	0.33	0.42	0.25	0.32	0.32
LSTM	Micro	0.99	0.99	0.99	0.99	0.45	0.33	0.25	0.31	0.46	0.54	0.37	0.38	0.35	0.99	0.99	0.99	0.99	0.45	0.33	0.25	0.31	0.46	0.54	0.37	0.38	0.35
	Macro	0.99	0.99	0.99	0.99	0.15	0.11	0.08	0.11	0.16	0.20	0.11	0.13	0.11	0.99	0.99	0.99	0.99	0.15	0.11	0.08	0.11	0.16	0.20	0.11	0.13	0.11



**Table 5** provides metrics for the quality of classification by thematic department codes. When comparing the results by different numbers of responses, it became obvious that the best results were achieved with 2 responses of the classifier, and F-score quality assessment reached 0.7. With 3 responses, the result was the worst, indicating that for this rubricator 3 responses were redundant. The share of correct responses in this case was at a very high level – 94–95% of answers of the classifier in 1 response were correct; in 2 responses, about 92–95% of answers were correct.

Best F-score results were shown by SVM-based classifier and recurrent neural network with LSTM layer. The results of neural networks of forward propagation are at a relatively high level too, and there is no significant difference between networks with 1 or with 2 hidden layers. The logistic regression proved itself somewhat worse, and the lowest results were shown by random forest. The poor result of the latter may be stipulated by weak resistance to emissions in its constituent models (decision trees) in combination with a significant amount of noise and emissions in text data.

**Table 6** provides similar metrics for codes of abstract journals. SVM-based classifier gave the highest values for all quality metrics. Almost for all classifiers approximately 99% of responses turned out to be correct, but the quality of the classification in the evaluation by F-score fell dramatically. This was caused by overfitting of the classifier for the most prevalent topics and by the availability of topics with an extremely low number of texts. At the same time, it was again optimal to give 2 responses to the text, but 3 responses gave a better classification quality compared to 1 response. Logistic regression and neural network with LSTM layer in this rubricator showed approximately the same results. In this case, this model had the smallest spread in micro- and macro-averaging. By micro-averaging, it had a significant advantage over all models, except the classifier based on SVM. There was also no significant difference in the results of neural networks of different structures. Random forest showed results only slightly worse than most other models; there was no significant backlog in F-score in this rubricator.

SRSTI classification results are shown in **Table 7**. SRSTI has the largest number of topics and the most unbalanced set which makes learning difficult. Nevertheless, the results for F-score with microaveraging are almost as good as the results for a similar quality metrics for the codes of abstract journals. With macro-averaging, the gap is already very significant. The F-score values reach 0.51 and 0.32 with micro- and macro-averaging, respectively. 2 responses again showed the maximum value of F-score; the quality of classification with 3 responses was not much better than with 1 response.

SVM-based classifier showed the best results in this rubricator. The quality of other models is much lower. Neural networks of forward propagation with 1 and 2 hidden layers and with LSTM layer showed the second-best result with micro-averaging; logistic regression – with macro-averaging. The random forest at both approaches to averaging showed the lowest results.

## Conclusion

Of all the tested models of machine learning, SVM-based classifier proved to be significantly better than others in the framework of the task, showing the highest results for all quality metrics. At the same time, the optimal size of the recommendation was 2 responses for thematic department codes, one response for abstract journals, and 2 responses for SRSTI. Nevertheless, the approach with a precise number of responses is not the most promising and will soon be replaced by an analysis of probabilities of responses and by the choice of one or more responses based on these indicators (with the expected increase in quality by F-score).

Accuracy is the most important metric of quality assessment for analyzing the capability of the classifier to process the actual flow of received texts; in all cases for 2 responses, it was at the level of 0.94 and higher. Despite the poor recognition of topics with a small number of texts in the training set, classifiers adapt well to the available data and can already be used to increase the efficiency of the work of specialists involved in manual text rubrication.

In the course of creating a basis for the development of DSS for VINITI RAS was found that the set of abstracts of scientific publications contains texts that are unevenly distributed across the topics which may reduce the quality of the classification. Alternative interpretations corresponding to the problem of regression in probability space, were proposed. Nevertheless, it was suggested that quality metrics, based on correctness of the prediction by a model of one, two, or three topics with the greatest probabilities and also averaged over micro and macro approaches, would be more indicative for assessing the quality of DSS.

A complete cycle of text analysis consists of the stages of text preprocessing, feature extraction, and classification. Preprocessing of texts included removing markup elements, lowercasing, removing stop words

and lemmatization. To substantiate the choice of the method of extracting features, a review of the methods was carried out and a number of experiments on training models with fixed hyperparameters were conducted. It was found that the vectors of dimension 50, obtained from the matrix of the text by the method of averaging column pooling, are better suited for classification. The text matrix was obtained using word2vec technology. When conducting experiments to select a classification model for DSS, the hyperparameters of the algorithms were selected by examining the combinations from a predetermined range. As a result of testing the classifiers, it was established that SVM with a kernel, based on the radial-basis function, had showed better results in the task of classifying scientific texts. For the rubricator, consisting of codes of VINITI RAS thematic departments, the method of reference vectors showed a quality of 0.65–0.70 in F-score, taking into account 1 and 2 responses with the greatest probabilities. The quality of work of the recurrent neural network for one response was approximately the same (0.65–0.68). When classified by codes of issues of abstract journals, F-score was 0.45–0.5 for 2 and 3 responses. For SRSTI codes, the quality in testing reached 0.3–0.5 in F-score stipulated by insufficient number of texts for some of the topics in the training set.

Thus, in DSS problem for the rubrication of scientific texts, the most suitable algorithm is SVM. The optimal value of the recommendation, based on the test results, is 2 responses.

## Acknowledgements

This research was supported by All-Russian Institute of Scientific and Technical Information of Russian Academy of Sciences (VINITI RAS). The authors are grateful to Alexandr Shapkin and Oleg Fedorec (VINITI RAS), for providing training datasets of actual documents.

## Competing Interests

The authors have no competing interests to declare.

## References

- Abuhaiba, ISI and Dawoud, HM.** 2017. Combining Different Approaches to Improve Arabic Text Documents Classification. *International Journal of Intelligent Systems and Applications*, 9(4): 39–52. DOI: <https://doi.org/10.5815/ijisa.2017.04.05>
- Automatic Content Classification with ABBYY Solutions. 2019. Available at <https://www.abbyy.com/solutions/document-classification/> [Last accessed 28.08.2018].
- Bourgonje, P, Moreno-Schneider, J, Srivastava, A and Rehm, G.** 2018. Automatic Classification of Abusive Language and Personal Attacks in Various Forms of Online Communication. 180–191. DOI: [https://doi.org/10.1007/978-3-319-73706-5\\_15](https://doi.org/10.1007/978-3-319-73706-5_15)
- Chen, G, Ye, D, Xing, Z, Chen, J and Cambria, E.** 2017. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. *B: 2017 International Joint Conference on Neural Networks (IJCNN)*, 2377–2383. IEEE. DOI: <https://doi.org/10.1109/IJCNN.2017.7966144>
- Clark, J, Koprinska, I and Poon, J.** 2003. A neural network based approach to automated e-mail classification. *B: Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, 702–705. IEEE Comput. Soc. DOI: <https://doi.org/10.1109/WI.2003.1241300>
- Dimov, D, Low, F, Ibrakhimov, M, Stulina, G and Conrad, C.** 2017. SAR and optical time series for crop classification. *B: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 811–814. IEEE. DOI: <https://doi.org/10.1109/IGARSS.2017.8127076>
- Du, J.** 2017. Automatic text classification algorithm based on Gauss improved convolutional neural network. *Journal of Computational Science*, 21: 195–200. DOI: <https://doi.org/10.1016/j.jocs.2017.06.010>
- Feng, K-Y, Cai, Y-D and Chou, K-C.** 2005. Boosting classifier for predicting protein domain structural class. *Biochemical and Biophysical Research Communications*, 334(1): 213–217. DOI: <https://doi.org/10.1016/j.bbrc.2005.06.075>
- gensim: models.word2vec – Word2vec embeddings. 2019. Available at <https://radimrehurek.com/gensim/models/word2vec> [Last accessed 28.08.2018].
- Géron, A.** 2018. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*. 1-e and. O'Reilly.
- Goncalves, T and Quaresma, P.** 2018. Evaluating preprocessing techniques in a Text Classification problem.
- Joulin, A, Grave, E, Bojanowski, P and Mikolov, T.** 2016. Bag of Tricks for Efficient Text Classification. Available at <http://arxiv.org/abs/1607.01759>. DOI: <https://doi.org/10.18653/v1/E17-2068>

- Klyshinsky, ES** and **Kochetkova, NA**. 2015. Method of extracting technical terms using the measure of strangeness. *Novye Informacionnye Tekhnologii v Avtomatizirovannyh Sistemah*, 17: 365–370.
- Korenus, T, Laurikkala, J, Järvelin, K** and **Juhola, M**. 2004. Stemming and lemmatization in the clustering of finnish text documents. *B: Proceedings of the Thirteenth ACM conference on Information and knowledge management – CIKM '04*, 625. New York, USA: ACM Press. DOI: <https://doi.org/10.1145/1031171.1031285>
- Kusner, M, Sun, Y, Kolkin, N** and **Weinberger, K**. 2015. From Word Embeddings To Document Distances. Bach, F and Blei, D (ed.), *B: Proceedings of the 32nd International Conference on Machine Learning*, 957–966. Lille, France: PMLR. Available at <http://proceedings.mlr.press/v37/kusnerb15.html>.
- Lapach, SN** and **Radchenko, SG**. 2012. The main problems of constructing regression models. *Matematicheskie Mashiny I Sistemy*, 1(4): 125–133.
- Lex, E, Seifert, C, Granitzer, M** and **Juffinger, A**. 2010. Efficient Cross-Domain Classification of Weblogs. *International Journal of Intelligent Computing Research*, 1(3): 55–62. DOI: <https://doi.org/10.20533/ijicr.2042.4655.2010.0007>
- Liu, Y, Bi, J-W** and **Fan, Z-P**. 2017. A method for multi-class sentiment classification based on an improved one-vs-one (OVO) strategy and the support vector machine (SVM) algorithm. *Information Sciences*, 394–395: 38–52. DOI: <https://doi.org/10.1016/j.ins.2017.02.016>
- LPU download page**. 2019. Available at <https://www.cs.uic.edu/~liub/LPU/LPU-download.html> [Last accessed 28.08.2018].
- Luo, Y, Ye, W, Zhao, X, Pan, X** and **Cao, Y**. 2017. Classification of Data from Electronic Nose Using Gradient Tree Boosting Algorithm. *Sensors*, 17(10): 2376. DOI: <https://doi.org/10.3390/s17102376>
- Raju, MK, Subrahmanian, ST** and **Sivakumar, T**. 2017. A Comparative Survey on Different Text Categorization Techniques. *Journal of Computer Science and Engineering*, 5(4): 1612–1618.
- Sammouda, R**. 2017. A Comparative Study of Effective Supervised Learning Methods on Arabic Text Classification, 17(12): 130–133.
- Scherer, D, Müller, A** and **Behnke, S**. 2010. Evaluation of pooling operations in convolutional architectures for object recognition. *B: Proceeding ICANN'10 Proceedings of the 20th international conference on Artificial neural networks: Part III*, 92–101. DOI: [https://doi.org/10.1007/978-3-642-15825-4\\_10](https://doi.org/10.1007/978-3-642-15825-4_10)
- Semberecki, P** and **Maciejewski, H**. 2017. Deep Learning methods for Subject Text Classification of Articles. cc. 357–360. DOI: <https://doi.org/10.15439/2017F414>
- Sokolova, M** and **Bobicev, V**. 2009. Classification of Emotion Words in Russian and Romanian Languages, 416–420.
- Sun, Y, Kamel, MS, Wong, AKC** and **Wang, Y**. 2007. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12): 3358–3378. DOI: <https://doi.org/10.1016/j.patcog.2007.04.009>
- Tang, D, Qin, B** and **Liu, T**. 2015. Document modeling with gated recurrent neural network for sentiment classification. *B: Proceedings of the 2015 conference on empirical methods in natural language processing*. cc. 1422–1432. DOI: <https://doi.org/10.18653/v1/D15-1167>
- Toloşi, L** and **Lengauer, T**. 2011. Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics*, 27(14): 1986–1994. DOI: <https://doi.org/10.1093/bioinformatics/btr300>
- Toman, M, Tesar, R** and **Jezeck, K**. 2006. Influence of word normalization on text classification. *B: Proceedings of InSciT*. cc. 354–358. Available at <http://www.kiv.zcu.cz/research/groups/text/publications/inscit20060710.pdf>.
- Uysal, AK** and **Gunal, S**. 2014. The impact of preprocessing on text classification. *Information Processing & Management*, 50(1): 104–112. DOI: <https://doi.org/10.1016/j.ipm.2013.08.006>
- Viniti.ru**. 2019. Available at <http://www.viniti.ru/> [Last accessed 01.01.2019].
- Vorobyov, NV** and **Puchkov, EV**. 2017. Text classification using convolutional neural network. *Yuniy Issledovatel Dona*, 9(6): 2–7.
- Wang, X, Jiang, W** and **Luo, Z**. 2016. Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts. *B: Proceedings of {COLING} 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee. cc. 2428–2437. Available at <https://www.aclweb.org/anthology/C16-1229>
- Wang, Y, Zhou, Z, Jin, S, Liu, D** and **Lu, M**. 2017. Comparisons and Selections of Features and Classifiers for Short Text Classification. *IOP Conference Series: Materials Science and Engineering*, 261: 012018. DOI: <https://doi.org/10.1088/1757-899X/261/1/012018>

- Watson Natural Language Classifier. 2019. Available at <https://www.ibm.com/watson/services/natural-language-classifier/> [Last accessed 28.08.2018].
- Wu, L, Yen, IE-H, Xu, K, Xu, F, Balakrishnan, A, Chen, P-Y, Ravikumar, P and Witbrock, MJ.** 2018. Word Mover's Embedding: From Word2Vec to Document Embedding. *CoRR*, abs/1811.01713. Available at <http://arxiv.org/abs/1811.01713>. DOI: <https://doi.org/10.18653/v1/D18-1482>
- Xu, B and Zhang, Y.** 2011. A new SVM Chinese text of classification algorithm based on the semantic kernel. *B: 2011 International Conference on Multimedia Technology*, 2857–2860. IEEE. DOI: <https://doi.org/10.1109/ICMT.2011.6003097>
- Yussupova, NI, Bogdanova, D and Boyko, MN.** 2012. Applying of Sentiment Analysis for Texts in Russian Based on Machine Learning Approach.


**How to cite this article:** Romanov, A, Lomotin, K and Kozlova, E. 2019. Application of Natural Language Processing Algorithms to the Task of Automatic Classification of Russian Scientific Texts. *Data Science Journal*, 18: 37, pp.1–17. DOI: <https://doi.org/10.5334/dsj-2019-037>

**Submitted:** 14 January 2019

**Accepted:** 23 July 2019

**Published:** 12 August 2019

**Copyright:** © 2019 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

 *Data Science Journal* is a peer-reviewed open access journal published by Ubiquity Press.

**OPEN ACCESS** 