Automatization of Scientific Articles Classification According to Universal Decimal Classifier

Aleksandr Romanov (0000-0002-9410-9431), Konstantin Lomotin (0000-0001-8741-502X) and Ekaterina Kozlova (0000-0003-2737-3694)

National Research University Higher School of Economics
Moscow, Russia
a.romanov@hse.ru, {ke.lomotin, hse.kozlovaes}@gmail.com

Abstract. This research examines the problems of automatic scientific articles classification according to Universal Decimal Classifier. To reveal the structure of the train data its visualization was obtained using the recursive feature elimination algorithm. Further; the study provides a comparison of TF-IDF and Weirdness – two statistic-based metrics of keyword significance. The most efficient classification methods are explained: cosine similarity method, naïve Bayesian classifier and artificial neural network. This research explores the most effective for text categorization structure of the multi-layer perceptron and derives appropriate conclusions.

Keywords: Keyword Selection, Text Classification, TF-IDF, Weirdness, Artificial Neural Network, Cosine Similarity.

1 Introduction

Nowadays, many intelligent systems use text databases. Thus, they incorporate text analysis algorithms to resolve various data issues. This research is dedicated to one of the most relevant data analysis problem - text classification performed mostly manually at present (publishing houses, libraries). The result of this study can be implemented in intelligent human interaction systems, and automatic document management. In this paper, we describe our experiments that aim to reveal the most optimal method for automatic text classification within the context of a specific task of scientific articles classification. In Section 2, we explain why the automatization of text classification is a crucial task. Section 3 is devoted to the stages of our research, experiments, and result analysis. Section 4 describes prospective areas of this study. Section 5 concludes the presentation of our experiments.

Universal Decimal Classifier, or UDC, is widely used in Russian science to describe the topic of the article. It consists of some general topics, which are divided into the specific ones. UDC has from two to six levels depending on the theme. So, a full UDC code accurately determines the topic of the article. Modern scientific articles cover almost all fields of human knowledge and include a massive amount of data. To handle with it the methods of machine learning are used.

2 The purpose and relevance of the research

The research is aimed at the development of the convenient scientific articles classifier according to UDC topics. This topic system is quite complicated and difficult for automatic classification. For this reason, the work results may find application in a wide range of tasks connected with natural language processing. For example, a web-search engine can use some considered principles to increase the efficiency of request handling, or context advertisements improvement [1].

At present, the majority of scientific articles are rubricated by the authors or publication moderators. The automatic classification system provides reduction of human maintenance and increases the articles search convenience. The application of text analysis system is realized mostly by large corporations like Google or Microsoft. The principles of machine learning can change the routine human work for automatic systems work in the nearest future.

UDC was selected as rubricating system for two reasons:

- UDC can be unambiguously translated to most other systems (SRSTI, HAC), but not vice versa;
- Most articles in the train data set contain UDC code, and only a few ones contain SRSTI or HAC codes.

3 Stages of the research

To create a natural language text classifier, it is necessary to pass through a number of stages. The following steps are needed [2]:

- Data collection for processing and classifier training;
- Keywords defining;
- Parameters selection and classifier training;
- Testing and results analysis.

3.1 Creation of scientific articles database

Data collection. As a source for scientific articles processing, web resource cyberleninka.ru was chosen [3]. All the articles are divided into 98 specific topics or “specializations” there. Every article has a UDC code which is assigned by author or moderator.

As the articles can be only downloaded in PDF format, a decision to download pages as a text was made. 39,000 web pages were downloaded in *.html format; then highly specialized parser was developed. The parser is adapted for cyberleninka.ru web page structure, and it does not need the full DOM-tree of document constituting. So, it was possible to get the articles texts quickly sorted according to the UDC topics (approximately for 30 minutes). The sorting results show that the specializations are distributed according to the UDC topics irregularly (Fig. 1), considering the fact that the nearly same amount of texts are downloaded for every specialization (Y-axis shows the amount of articles).

Also, on this step text pre-processing included:

- Words with Latin symbols removal;
- Removal of punctuation symbols, which do not divide text into sentences;
- Changing all words register to lower case;
- All line break symbols deletion.

All these actions were taken in order to decrease the working time of morphological processor written in Python language, while the parser was written using C++. It was noticed that transferring the part of pre-processing to the parser increases the whole speed of system.

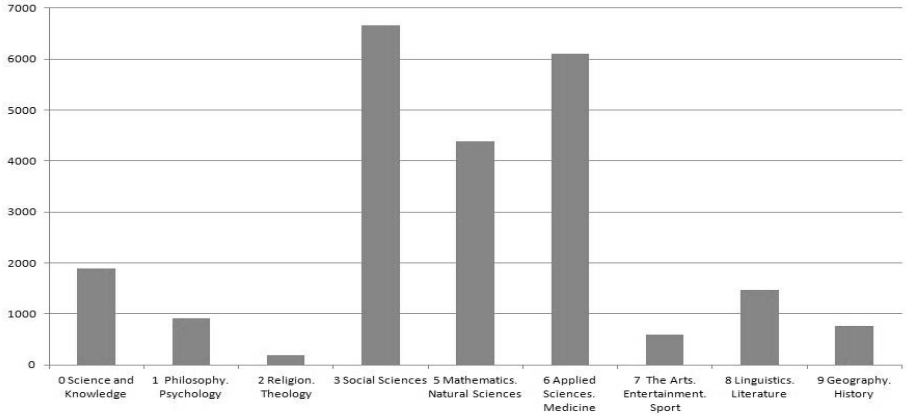


Fig. 1. Articles specialization distribution from cyberleninka.ru web-resource according to UDC topics.

In order to receive more complete information about collected data we used a data mining framework. It implements the visualization of scientific articles location in the feature space. This space has 270 dimensions (30 features for each of 9 classes showed the best result in the series of experiments) which depend on the selected keywords amount. To handle with it we introduced the feature reduction method to reduce the number of dimensions to 3. This method was implemented with recursive feature elimination algorithm.

The application of the feature selection method in this research allows estimating the intractability of text classification. Fig. 2 represents the result of the script that implements the data visualization. Each point means an article, and axes show relative coordinates of the point reduced from multidimensional space. The point color means UDC topic of the article according to the scale.

It is evident that not all of text clusters are linearly separable, some of them very intersected. Moreover, texts of the 5th and the 6th themes form a single cluster. In order to separate these classes, we approximately accept them for linearly separable. To separate some of the close classes, it is worth to apply deep learning algorithms or probability model. This approach is known as “boosting”. It has an implementation as a set of algorithms.

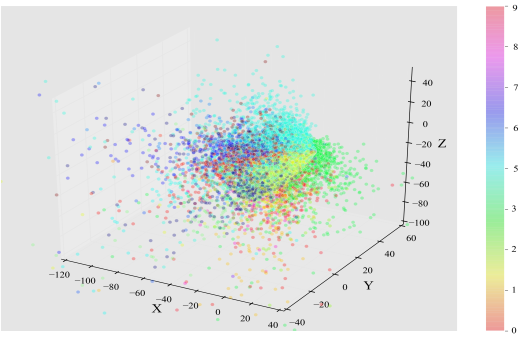


Fig. 2. Visualization of the collected data.

At the current research stage this visualization allows to estimate the amount of data required to train the classifiers. Unnecessarily large sample set increases the risks of such a situation when the model may be not enough complicated to select features of any class. At the same time the lack of data behavior of the model may be unpredictable if it processes the text that does not look like any of the training set. Another issue of the insufficient amount of data is the fact that the classifier selects only general features and forms the groups of the classes based on the available ones. For example it can manifest itself by division of all the articles into “humanitarian” and “engineering”.

Features of UDC code. UDC code is a group of numbers, divided by dots. Symbol “.” means transition of a subtopic of the topic which number is written on the left side of the dot, to a subtopic which number is written on the right. Also, the code includes some operations like attitude of one topic to another, spread of the topic to another, accession, grouping etc.

The rules of working with UDC are uniquely described by the State Standard [4]. However, during the analysis of the texts for which the authors defined the UDC codes by themselves many mistakes were revealed, so codes of the majority of the articles cannot be parsed. On the current stage of research, the classification is made by only the first level of UDC topics. It is believed that the classification methods which are effective for the first level of UDC code will show good results on the following levels.

3.2 Keywords selection

Keywords, in the context of current research, are the words which provide the possibilities to refer the text to one topic or another. During the research, some experiments were made with two measures of word meaning: TF-IDF [5] and weirdness.

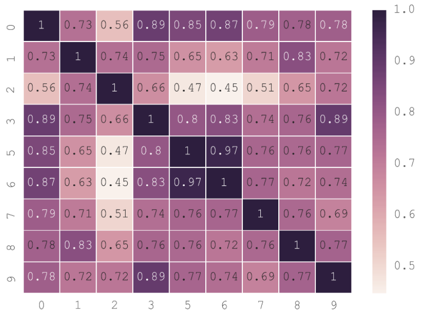
TF-IDF value shows the degree of significance in the keywords list of the current topic [6]. This measure shows modest results because it is not pre-oriented to processing “classes” of documents, corresponding to the topics.

Weirdness is number of word occurrence divided by the overall amount of words in the topic, multiplied by the same coefficient in other topics [7]. Further experiments are based on keywords, obtained using weirdness measure.

3.3 Choice of parameters and classifier training

For realization of this task three different methods were chosen: cosine similarity, artificial neural network and naive Bayes classifier.

Method based on cosine similarity. The main idea of this method is calculating the angles cosines between the text vector (which is made using keywords) and vectors of all topics. Then it is necessary to find the highest cosine. Theme vector which makes the highest cosine has the direction closer to text vector and, as a result, the text can be referred to this topic with a certain percentage of probability. Matrix at Fig. 3 shows cosines between theme vectors.



	0	1	2	3	5	6	7	8	9
0	1	0.73	0.56	0.89	0.85	0.87	0.79	0.78	0.78
1	0.73	1	0.74	0.75	0.65	0.63	0.71	0.83	0.72
2	0.56	0.74	1	0.66	0.47	0.45	0.51	0.65	0.72
3	0.89	0.75	0.66	1	0.8	0.83	0.74	0.76	0.89
5	0.85	0.65	0.47	0.8	1	0.97	0.76	0.76	0.77
6	0.87	0.63	0.45	0.83	0.97	1	0.77	0.72	0.74
7	0.79	0.71	0.51	0.74	0.76	0.77	1	0.76	0.69
8	0.78	0.83	0.65	0.76	0.76	0.72	0.76	1	0.77
9	0.78	0.72	0.72	0.89	0.77	0.74	0.69	0.77	1

Fig. 3. Cosines between theme vectors.

It should be pointed out that the angles between 5 and 6 topics are the least. This fact can be also noticed later in testing results of other classifiers. Also, it is shown that vectors of many topics are quite close to each other. There are some more methods based on other metrics: for instance, on Euclidean distance. In this method the Euclidean distance between vectors is the main measure. The disadvantage of this method is the sensitivity to the capacity of the sample meaning a multiply all the vector elements by the same constant value. The cosine of the angle will not change in contrast to the distance.

The advantages of the method based on cosine similarity are simplicity of realization and working speed. For the angle cosine calculation, formula (5) exists:

$$\cos\varphi = \frac{a \cdot b}{|a||b|}, \tag{1}$$

where φ – angle between a and b vectors.

The disadvantage of this method is the situation when angle between topics is small; there is a high chance to refer the text to the incorrect, but close topic by destination. It is caused because of the fact that the cosine measure is approximate, so the probability of a big mistake exists. And if the angle between these vectors is smaller than the error, these themes can be considered to be correct by this method. However, UDC has tree structure, and the fact of mistake can cause movement to the incorrect branch. That's why this method shows good results when the topics are well distinguishable.

Artificial neural network. The artificial neural network (ANN) is a mathematical model based on the principles of humans brains functioning. ANN are capable of training and they are used for solving such tasks as classification, clustering, prediction, extrapolation, and function approximation etc. Contribution of artificial neural networks in the text categorization is generally recognized [8, 9].

There are many different kinds of ANN and each of them has its own advantages in different conditions. On the current stage of the research Rosenblatt perceptron with McCulloch neurons [10] was chosen. This structure is quite popular because of its flexibility and universality. It is proved experimentally that two layers of neurons are enough to solve any task, if we talk about perceptron [11]. Such parameters as amount of neurons and epochs can be hardly calculated, so they are defined experimentally. According to series of experiments, it was revealed that two layers with eight neurons in each one are the best configuration for the available data [12].

Tests after 1000 epochs and after 3000 epochs proved that the system did not overfitted. There were also experiments with the structure which realized principles of deep learning, but they did not show satisfactory results. It can be connected with the fact that deep learning is based on more complicated theory and it demands to choose the parameters more carefully.

Naive Bayesian classifier. This method is based on Bayesian theorem. According to it, a probability of belonging to a class can be calculated using the bunch of features (events).

Naive Bayesian classifier is a simple probabilistic classifier, based on the usage of Bayesian theorem with strict (naive) assumptions of independence [13]. This model has a lot of strong sides that give it precedence in text processing [14, 15]: relatively small data set for training, simplicity and small number of essential parameters.

Training by naive Bayesian is based on the independence of features in general. According to Bayesian formula:

$$P(y|x) = P(y|(x^1, \dots, x^d)) = \frac{\mathrm{P(y)P(x^1,\dots,x^d|y)}}{\mathrm{P(x^1,\dots,x^d)}} \tag{2}$$

In the assumption of features independence we can derive the following formula:

$$P\big(y\big|(x^1, \dots, x^d)\big) = \frac{\mathrm{P(y)}\ \prod_{i=1}^{d} \mathrm{P(x^i|y)}}{\mathrm{P(x^1,\dots,x^d)}} \tag{3}$$

As $P(x^1, \dots, x^d)$ is independent from *y*, so the Bayesian classifier formula can be written as following:

$$\hat{y}(x) = \arg\max_{y} \prod_{i=1}^{d} P(x^{d}|y) \quad (4)$$

Then on the base of formula (8) the definite amount of words are used as event X^{i}, where i – word number in the keywords list. The model tries to calculate the probability of the fact that with the certain class, the value of feature is equal to the one in the training set. Each class is used for the prediction; multiplying probabilities are found for close values, and then the maximum ones are chosen.

3.4 Testing classifiers and analysis of the results

UDC allows assigning of several code numbers to an article. Owing to this feature, 2 topics are considered to be chosen as a possible result of a cosine similarity and ANN. Naive Bayesian classifier produces only one result because of limitations its realization. It does not make an insurmountable problem, and it will be eliminated in the future research.

Preliminary estimation shows that percentage of errors that authors make in the preparation of the UDC codes can reach 10%–15%. Currently, the intelligent system of parsing is in development, and it can significantly reduce this amount.

For clarity of the analysis, the results are shown in charts of "heatmap" type. The vertical axis includes the numbers of tested topics while the horizontal one shows the results of testing. The brighter area at the intersection of string *a* and column *b* is, the more texts of theme *b* are defined as the texts of theme *a.*

When there are no mistakes at all, this diagram looks like as set of bright peaks on the matrix main diagonal without peaks in other points. But the results of two possible themes make secondary peaks appear.

Cosine similarity. The result of testing classifier based on cosine similarity is shown in Fig. 4.

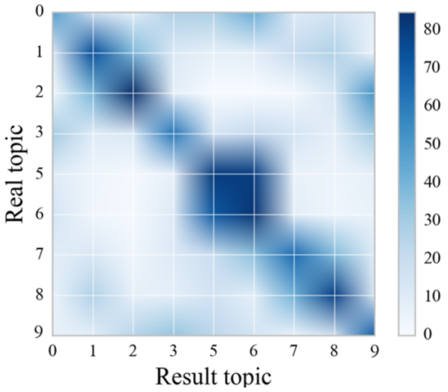


Fig. 4. Heatmap of cosine similarity method results.

It is obvious that this classifier has difficulties in distinguishing topics 5 (Mathematics. Natural Sciences) and 6 (Applied Sciences. Medicine). It is directly related to small angle between these topics. Tests of cosine similarity method showed the average amount of right answers of about 72%. Mistakes are distributed among the topics irregularly, so it makes sense to take a look at Fig. 5, where X-axis marks a topic and Y-axis marks a fraction of correct answers in the test.

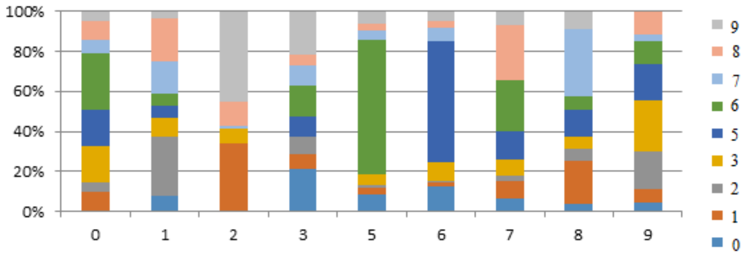


Fig. 5. The distribution of mistakes among the topics.

Now it is apparent that there are other pairs of topics that classifier muddled up.

Artificial neural network. ANN with two layers by 8 neurons showed the average amount of right answers of about 51%.

As it was said before, selection of such parameters as amount of neurons and layers is usually done experimentally. Amount of layers means only layers that take part in calculations, such as hidden and output layers. A series of experiments was carried out to find the best structure for ANN. We made experiments with such structures as:

- 3 layers by 9 neurons
- 9 layers by 9 neurons
- 1 layer by 27 neurons
- 3 layers by 27 neurons
- 1 layer by 243 neurons

Structure “9-9-9” represents the implementation a sequence of nonlinear transformations. As shown in Fig. 6, this structure demonstrated the best classification quality.

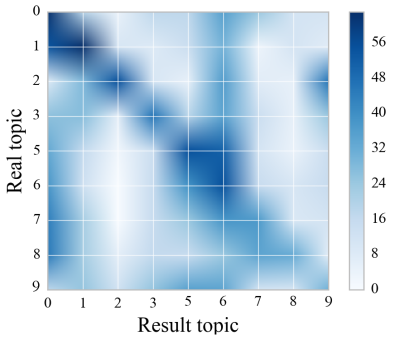


Fig. 6. “9-9-9” ANN test result.

Naive Bayesian classifier. The classifier based on Bayesian theorem showed the average amount of right answers of about 64%. The distribution of mistakes is shown in Fig. 7.

The classifier based on probability model has good chances to keep high results after conversion to UDC subtopics.

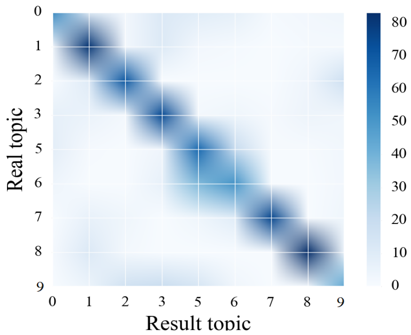


Fig. 7. Naive Bayesian classifier testing result.

Results analysis. The results cannot be called ideal from the applied point of view, but their analysis can be useful for understanding how to improve the classification. The percentage of the mistakes occurred because of the choice of incorrect source was distributed quite evenly.

Cosine similarity

This method of classification has the highest average amount of right answers. But while on the lower levels the UDC topics become closer to its meanings, the percentage of successful classification significantly decreases. In other words, this method does not scale well.

The efficiency of method directly depends on the quality of keywords that match topics and on the angle between the vectors of samples. So, it is possible to match the higher results, but the more perspective way is to evolve other classifiers.

Artificial neural network

This method is chosen as the most promising, despite the low average amount of right classifications. The efficiency of ANN depends on a large number of options: different structures, quantity of epochs in training, variety of selection etc. If it has enough synapses and diversity of the training data. ANN can classify the texts of any difficulty according to the topics. However, this model is quite difficult to set up and it faces such problems as reeducation, allocation meaningless features etc.

Multiplicity of structures and paradigms permits to proceed from multilayer perceptron to other structures, such as a recurrent neural network (RNN) and a radial basis function network. It is possible to adapt a convolutional neural network (CNN) to work with the assigned purpose.

Naive Bayesian classifier
The efficiency of this method has an intense relation to a capacity of keywords, as well as cosine similarity. But in comparison with cosine similarity, it does not need exemplary vectors of topics. This factor makes it possible to consider Bayesian classifier to be a perspective method on a line with ANN.

4 PERSPECTIVES OF FUTURE RESEARCH

The analysis of classification results shows the most suitable ways that are worthwhile to deepen the future research.

4.1 Boosting algorithm

Boosting is a system of several consecutive classifiers, where each one improves the mistakes of the previous one. In the considered objective, boosting can solve the problem of irregular distribution of mistakes according to the topics and ensure supplement of the model with a certain kind of flexibility. Consequently, it will be possible to develop the classifiers that specialize on topics that are difficult to differentiate, and to simplify classifiers that work with other topics.

4.2 Latent semantic analysis (LSA)

In this research statistical processing of all texts of scientific articles is performed. However, such a technique named “bag of words” has some accuracy limits while the meaning of overwhelming majority of words depends on the context where they are used. Therefore, texts semantic processing can become a considerable step to enhancement of classification quality.
The intermediate between statistical and semantic processing is a processing when model works with the list of key collocations instead of the list of keywords or a semantic map. It is obvious that the phrase “neural network” comprises wore sense and has different meaning compared with the words “neural” and “network” separately. So, it is possible to considerably increase the percentage of successfully classify texts not deepening to the semantic analysis theory.

4.3 Recession in the UDC structure

As it was said, UDC has a tree structure. In order to go to the next levels, it plans to create the classifiers system as it is represented in Fig. 8.

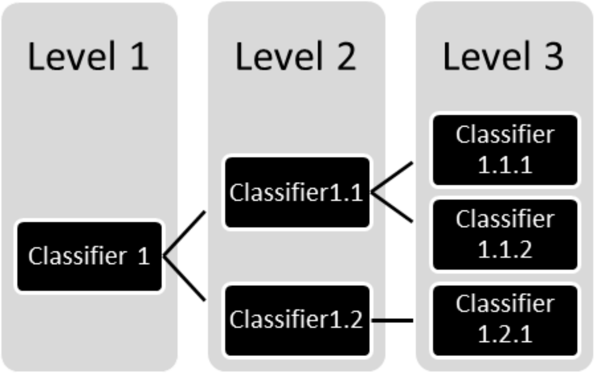


Fig. 8. The structure of the classifiers system.

It is possible to select 2 of the most promising ways of movement within UDC structure [16].

1. Decision on the next classifier selection for the lower UDC level shall be taken on the basis of the result of the previous one. The advantage of this modus is its productivity and no need in extra calculations.

   The disadvantage is the classification error multiplies during the movement from level to level. In this way, when having 90% correct answers on the each level separately, approximately 73% of correct answers on the third level of the system are received. Also, if the result of the classification is wrong, the start of all the next classifiers will make no sense at all. This approach requires an extremely high accuracy of the classification.

2. Decision shall be taken on the basis of the result of all classifiers. The advantage of this approach is classification error does not increase recession. All considered classifiers
   a. give a result as a number from 0 to 1 for any topic. It is the evaluation of the attachment of the text to any theme. If we
   b. firstly get a result of each classifier in the system and then set it up for each topic and its subtopics, we will receive a set of ways from the first UDC level to the last one with evaluation of how this set (UDC code) matches the article.

The disadvantage of this approach is a low performance. To implement this method, the result of all classifiers is needed. It is rather time consuming, and it requires significant machine resources costs.

5 CONCLUSION

In the research, a brief review of most suitable methods of classification is given. It includes cosine similarity, artificial neural network, and naive Bayesian classifier. It is revealed that the applied classification method, based on the cosine similarity, can ensure 72% of correct answers. The optimal application area for two measures of the word meaning (TF-IDF and Weirdness) is found. The recession in the UDC structure, Boosting algorithm, and Latent semantic analysis in the context of more complex classifier development are considered. Also, the analysis of results is made and the best ways for future research are proposed.

References

1. Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: The Adaptive Web. Lecture Notes in Computer Science, vol. 4321, pp. 325–341, Springer, Heidelberg (2007).
2. Romanov, A.Yu., Lomotin, K.E., Kozlova, E.S., Kolesnichenko, A.L.: Research of Neural Networks Application Efficiency in Automatic Scientific Articles Classification According to UDC. In: IEEE 2016 International Siberian Conference on Control and Communications (SIBCON), NRU HSE, Moscow (2016).
3. Cyberleninka, Open Access Journal Articles for Open Science in Russian, http://cyberleninka.ru/, last accessed 2017/03/28.
4. GOST 7.90-2007 System of standards on information, librarianship and publishing. Universal decimal classification. Structure, rules for use and indexing, Standartinform, Moscow (2008).
5. Ramos, J.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning, vol. 242, pp. 133–142 (2003).
6. Aizawa, A.: An information-theoretic perspective of TF-IDF measures. In: Information Processing & Management, vol. 39, 45–65 (2003).
7. Klyshinsky, E.S., Kochetkova, N.A.: Technical terms selection using weirdness measure. In: Modern informatical technologies in automatical system, vol. 17, pp. 365–370 (2014).
8. Nam, J., Kim, J., Mencía, E.L., Gurevych, I., Fürnkranz, J.: Large-scale multi-label text classification-revisiting neural networks. In: arXiv preprint arXiv:1312.5419 (2013).
9. Wang, F., Wang, Z., Li, Z., Wen, J.R.: Concept-based short text classification and ranking. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 1069–1078, ACM (2014).
10. Anderson, J.A. McCulloch–Pitts Neurons, Encyclopedia of Cognitive Science. John Wiley & Sons, New York (2006).
11. Rosenblatt, F.: Perceptron simulation experiments. In: Proceedings of the IRE, vol. 48, 301–309 (1960).
12. Clark, J., Koprinska, I., Poon, J.: A neural network based approach to automated e-mail classification. In: Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI'03), pp. 702–705, IEEE, Halifax (2003).
13. Rojas, R., Neural networks – A systematic introduction chapter 3. Springer, Berlin (1996).
14. Narayanan, V., Arora, I., Bhatia, A.: Fast and accurate sentiment classification using an enhanced Naive Bayes model. In: International Conference on Intelligent Data Engineering and Automated Learning, pp. 194–201, Springer, Berlin, Heidelberg (2013).
15. Xuan, J., Jiang, H., Ren, Z., Yan, J., Luo, Z.: Automatic bug triage using semi-supervised text classification. In: arXiv preprint arXiv:1704.04769 (2017).
16. Tóth, E. Innovative solutions in automatic classification: a brief summary, vol. 52, 48–53 (2002).