Keywords Extraction Based on Word2Vec and TextRank

Yong Zhang
Computer School, Central China Normal University
Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning
Wuhan, China
+86 15802766977
ychang@mail.ccnu.edu.cn

Fen Chen
Computer School, Central China Normal University
Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning
Wuhan, China
+86 13476093407
1261653318@qq.com

Wufeng Zhang
Computer School, Central China Normal University
Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning
Wuhan, China
+86 13396062375
578901743@qq.com

Haoyang Zuo
Computer School, Central China Normal University
Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning
Wuhan, China
+86 18872249165
740645146@qq.com

Fangyuan Yu
Computer School, Central China Normal University
Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning
Wuhan, China
+86 13627207241
1915654292@qq.com

ABSTRACT

In order to improve the performance of keyword extraction by enhancing the semantic representations of documents, we propose a method of keyword extraction which exploits the document's internal semantic information and the semantic representations of words pre-trained by massive external documents. Firstly, we utilize the deep learning tool Word2Vec to characterize the external document information, and evaluate the similarity between the words by the cosine distance, thus we obtain the semantic information between words in the external documents. Then, the word-to-word similarity is used to replace the probability transfer matrix in the TextRank of word graph of the target document. At the same time, the information of the title and the abstract of the internal document are exploited to construct the words' semantic graph for keyword extraction. The experiments select the related academic paper data from AMiner as experimental data set. The experimental results show that our method outperforms the TextRank algorithm and the precision, recall and F-score of the five keywords are increased by 28.60%, 10.70% and 12.90% respectively compared to the single TextRank algorithm.

CCS Concepts

• **Theory of computation → Theory and algorithms for application domains → Machine learning theory → Unsupervised learning and clustering • Mathematics of computing → Discrete mathematics → Graph theory → Graph algorithms • Mathematics of computing → Discrete mathematics → Graph theory → Random graphs**

Keywords

keyword extraction; word2vec; TextRank; word map

ICBDE '20, April 1–3, 2020, London, United Kingdom
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7498-9/20/04…$15.00
DOI: https://doi.org/10.1145/3396452.3396460

1. INTRODUCTION

Keywords are a highly condensed core of an article, and you can quickly learn the general content of an article from a few keywords. Keyword extraction plays a crucial role in the fields of document retrieval, automatic abstracts of papers, text labeling, text topic classification and extraction. However, the manual extraction of keywords is time-consuming and laborious, and because of different personal cultural levels and thinking modes, the extracted keywords often have large deviations. Therefore, how to automatically extract keywords from articles has become a popular research direction.

The current common keyword extraction steps mainly include the following two aspects: Firstly, get candidate keywords. In this step, we first segment the obtained text corpus, then remove stop words and duplicate words, and finally retain related words as needed to obtain candidate keywords, generally adjectives, verbs, and nouns. Secondly, filter keywords from candidate keywords. At this step, we can generally judge whether it is a keyword by keyword tagging or keyword probability ranking. Meanwhile, it can also be identified in the form of a graph through the relationship between words in the text. The typical representative of the graph model is TextRank, which was first proposed by Mihalcea [1] for keyword and sentence extraction.

When the classic TextRank algorithm is applied to text keyword extraction, it mainly considers the internal information of a single document. However, by using local lexical information (co-occurrence window) to sort subsequent keywords, it ignores the influence between adjacent nodes. Based on the basic conception of TextRank, Xia [2] introduced three influences to calculate the probability transition matrix between words. Ning [3] trained the word vector from external information through Word2Vec and combined it with the TextRank algorithm to calculate the probability transition matrix.

In order to be able to combine external corpus information with the TextRank algorithm, we made some improvements on the basis of the algorithm proposed by Liu [4]. At the beginning, the external document set is trained by using Word2Vec in order to obtain the corresponding word vectors and then we calculate the similarity between the word vectors. The similarity between word vectors is used to replace the out-degree and in-degree of edges in TextRank to improve the TextRank algorithm. At the same time, the information of the title and the abstract of the internal document are exploited to construct the words' semantic graph for keyword extraction. Finally, iterative weights are used to rank candidate keywords and extract keywords.

2. RELATED WORK

2.1. Research Status

At present, there are mainly two methods of keyword extraction: supervised learning method and unsupervised learning method. The main difference between these two methods is whether there is a labeled corpus. The method of supervised learning requires a manually labeled corpus, which mainly converts the problem into a binary classification problem. It generally rely on methods such as decision tree (DT), support vector machine, and naive Bayes. Zhang [5] used SVM to train a keyword extraction model. The quality of the training corpus has a great effect on the accuracy of the model, and has a great impact on the results of keyword extraction. Now the text corpus that has been labeled with keywords is limited, and all training sets need to be manually labeled, which will inevitably bring certain subjective colors, resulting in inaccurate experimental data. At the same time, a large amount of manual labeling is costly. Therefore, the labeling of the corpus becomes a bottleneck for supervised learning methods.

The unsupervised learning method does not require a well-labeled corpus, and the keywords of the article can be determined by the probability ranking. Firstly, according to the statistical characteristics of the candidate keywords, some feature weighting indicators are set, and then according to the weights from high to low, the first 4-7 words with the highest weight are taken as text keywords. Keyword extraction algorithms for unsupervised learning are mainly divided into the following three categories: (1) Statistical methods based on word frequency, mainly based on the TF-IDF model. (2) Topic model based method, mainly based on LDA topic model. (3) Word graph-based method, mainly based on TextRank.

Statistical method based on word frequency----TF-IDF model. The TF-IDF model is mainly used to evaluate the importance of a word to a document set or a document in a corpus. The main idea is to pick out words that appear frequently in one document and less frequently in other documents, and use the words as keywords in the document. Because TF-IDF only considers the frequency of words, it is not comprehensive enough, and it cannot reflect the semantic information of words. Hu [6] and Jia [7] made different improvements to the TF-IDF algorithm, which greatly improved the accuracy of keyword extraction.

Method based on topic model----LDA [8] model. The LDA model is an unsupervised Bayesian model, which can give each document in the document set as a probability distribution. P (word | document) = P (word | topic) P (topic | document). The probability of a word appearing in a document can be obtained by multiplying the probability of a word appearing on the same topic and the probability of appearing on a topic in the same document. This method of keyword extraction based on hidden topic patterns has received increasing attention [9-10], and the extraction effect is strongly related to the topic distribution of the training document set.

Word graph based method----TextRank algorithm. Keyword extraction based on the word graph model is to convert the processing of text into a connection analysis of a network graph. TextRank algorithm is the most commonly used algorithm in the field of word graphs. This algorithm is total different from TF-IDF and LDA in that it does not require batch training corpus, and it mainly analyzes the relationship between words in the text. It is simple and effective for independent single text keyword extraction. However, because just a single document [11] is considered, the external document information cannot be used, and the network graph is constructed only through the window co-occurrence relationship, which results in that the TextRank algorithm has a lot of room for improvement in extracting keywords.

With the rise of deep learning and neural networks [12-14], Liu [4] proposed a keyword extraction algorithm based on the combination of Word2Vec and TextRank. Using political key thesaurus to modify the initial weight, determine the connection relationship of words based on the context, and build a probability transition matrix based on Word2Vec to improve the efficiency and accuracy of keyword extraction. Li [15] improved the automatic keyword extraction method based on TextRank by using domain knowledge. Bai [16] proposed a keyword extraction method based on BP neural network, which mainly includes analyzing and extracting term features (including 13 features such as word frequency and position), determining the number of nodes in the hidden layer, and judging whether it is a keyword, but the generalization ability of the network needs to be improved. Johannes [17] proposed an end-to-end neural keyword extraction algorithm based on the siameseLSTM network, eliminating the need for artificial feature engineering, and the model focused on extracting keywords from very short documents.

Based on the basic research ideas of document [4], this paper leads the information of external documents into TextRank. First, Word2Vec is used to vectorize the external information. Then the cosine distance is used to calculate the similarity between the words to change the probability transition matrix in the TextRank word graph algorithm. Finally, the title and abstract information are combined to construct a new semantically related node relationship for keyword extraction.

2.2. TextRank and Word2Vec

The TextRank algorithm is derived from the well-known page ranking algorithm PageRank [18]. The model of the TextRank algorithm can be expressed as a weighted graph model $G=(V,E)$, Where "V" represents the set of all nodes in the graph, that is, the set of nodes formed by all words, and "E" is the set of all edges in the graph, that is, the weight "W_{ji}" between each word "V_i" and its neighboring node "V_j". For any given node "V_i", $In\ (V_i)$ is the in-degree of the node set that points to the node "V_i", $Out\ (V_i)$ is the set of nodes pointed to by the node "V_i". The calculation formula for the criticality of the word node "V_i" is as follows:

$$S(V_i)=(1-d)+d*\sum_{V_j\in In(V_i)}\frac{W_{ji}}{\sum_{V_k\in Out(V_j)}W_{jk}}S(V_j) \tag{1}$$

Equation (1) is a recursive formula for calculating the criticality of words using the TextRank algorithm, where *d* is the damping coefficient and the value is between 0-1. Generally, 0.85 is taken. The candidate keyword map is shown in Figure 1:

Word2Vec is an efficient tool for characterizing words as real-valued vectors introduced by Google in 2013. It can efficiently train datasets with more than one million levels. There are mainly

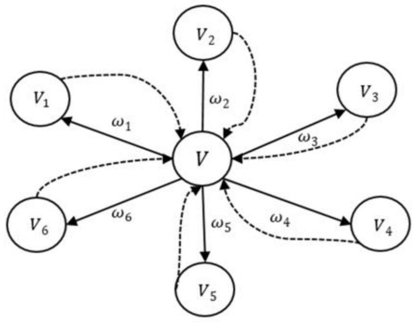


Figure 1. TextRank candidate keyword graph.

two models: CBOW and Skip-gram. CBOW model predicts "ω_a" given the context "ω_{a-2}", "ω_{a-1}", "ω_{a+1}", "ω_{a+2}" of the current word "ω_a". Skip-gram model predicts the context "ω_{a-2}", "ω_{a-1}", "ω_{a+1}" "," ω_{a+2} " on the premise of that the current word "ω_a" is known.

3. EXTRACTION MODEL

The main steps of the improved Word2Vec and TextRank algorithms are as follows: First, the collected corpus resources are preprocessed for text. Then use the deep learning tool Word2Vec to train the pre-processed word set into word vectors, and calculate the cosine distance between the word vectors to build a new probability transition matrix. Next, set the size of the sliding window in TextRank, determine the connection edges between the nodes by combining the title and the abstract, and calculate the TextRank value of each word according to the new probability transfer matrix trained by Word2Vec. Finally, the values of the words are arranged from high to low, and the first few words are selected as the keyword set of the text. The algorithm flow is shown in Figure 2.

3.1. Text Preprocessing

First, the training set consisting of N documents including the title and abstract is segmented by the NLTK word segmentation tool, and then the stop word list is used to filter the stop words in the segmentation result, and finally the corresponding word set T_0 is obtained. At the same time, vocabulary reduction and deduplication are performed on the word set T_0, and important words such as nouns, verbs, verbs, adjectives are retained to obtain the word set T_1.

At the same time, all the keywords of the training corpus are extracted, and a keyword phrase table T_2 is established after deduplication processing, and it is merged into the word set T_1. The final word set is $T_3 = T_1 \cup T_2$. The word set T_3 will eventually be used for word vector training of Word2Vec.

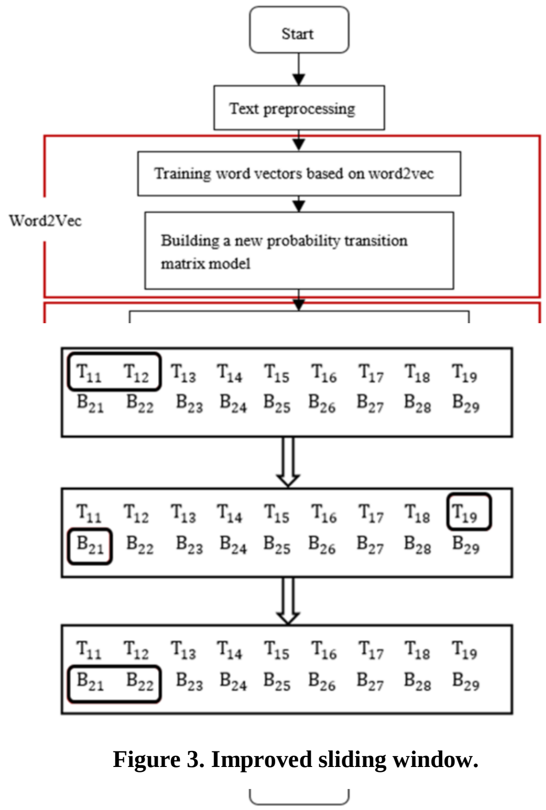
3.2. Training Word Vectors and Building Probability Transition Matrices

In the traditional TextRank algorithm, the probability transition matrix is a sparse matrix. If there is a connection between nodes, the weight is assigned a value of 1, and if not, the value is assigned 0. The probability transition matrix is as follows:

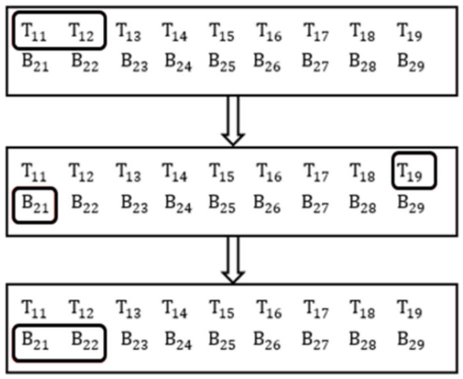
$$W=\begin{bmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1n} \\ \vdots & & \ddots & \vdots \\ \omega_{n1} & \omega_{n2} & \cdots & \omega_{nn} \end{bmatrix} \quad (2)$$

Use the Skip-gram model in the deep learning tool Word2Vec to train the word set T_3 to obtain the node word vector, and use the similarity between the nodes as the edge weights. For example, the word vector of the word "V_i" is $\overrightarrow{V_i}$, and the word vector of the word "V_j" is $\overrightarrow{V_j}$, Then the transition probability "ω_{ij}" is calculated using formula (3):

$$\omega_{ij}=\frac{\overrightarrow{V_i} * \overrightarrow{V_j}}{\left|\overrightarrow{V_i}\right| * \left|\overrightarrow{V_j}\right|} \quad (3)$$



Start
Text preprocessing
Training word vectors based on word2vec
Building a new probability transition matrix model
Word2Vec



T_{11} T_{12} T_{13} T_{14} T_{15} T_{16} T_{17} T_{18} T_{19}
B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29}

Figure 3. Improved sliding window.

Figure 2. Keyword extraction model diagram.

3.3. Determining Connecting Edges between Nodes by Combining Title and Context

Generally speaking, the keywords of the paper are mostly related to the title and abstract of the article, and many keywords come from the title or abstract of the paper. In order to make the abstract clear and coherent, there will be a certain connection between each sentence, and the content between the contexts will be rigorous. Therefore, when determining the connecting edge between nodes, we must consider the semantic relationship between the title and the context.

When the TextRank algorithm determines the edges between nodes, the sliding window only slides between sentences, causing a fault in context semantics. And the test set doesn't include the paper title, which to some extent excludes the influence of the paper title on the extracted keywords. The improved algorithm joins the title and abstract of the paper, and pre-processes the text to extract keywords. At this time the text doesn't have to be divided into many sentences, the sliding window is set to 2, and the processing flow is shown in Figure 3.

The first part is the title and the second part is the abstract. The sliding window is not bounded by sentences and makes it sliding smoother. It solves the problem that the traditional TextRank doesn't consider the semantic relationship between sentences and it simultaneously reflects the contextual connection.

3.4. Calculate TextRank Value

The text passes through the sliding window to find the connection between each nodes, and obtains the TextRank value of each nodes by the new probability transition matrix. By improving the algorithm, keywords are ranked according to the score of the TextRank value from high to low. This method extract 5, 7, and 10 keywords to observe the accuracy.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1. Data Set

This experiment uses computer science field papers as experimental data set, which are the knowledge graph related papers in AMiner, including data mining, machine learning, social networks, deep learning, and other computer field papers. There are 18,465,798 paper titles and abstracts from MAG (Microsoft Academic Graph) and 51,590,387 paper titles and abstracts from AMiner, and we extract the keywords marked by the author to form a vocabulary table.

During the test, we use 100 papers on the computer network and 100 papers randomly selected from conference papers such as ACL as test data set. The papers are roughly evenly distributed in the fields of data mining, machine learning, social networking, and deep learning. Both the training set and the testing set contain data on three aspects: title, keywords, and abstract.

4.2. Experimental results

4.2.1. *Evaluation Benchmark*

The keyword extraction method proposed in this paper is named Vector-TextRank (V-TextRank). In order to study whether the V-TextRank keyword extraction method can improve the keyword extraction effect, the term frequency method TF-IDF and the word graph method TextRank are used as comparison benchmarks. At the same time, we also selected the same kind of comparison, that is, consider the factor of the title and not consider the factor of the title. The keyword extraction method without considering the title factor is called the UNT-TextRank method.

4.2.2. *Evaluation Index*

This experiment uses three evaluation indicators: precision rate P, recall rate R, and F-score. These three evaluation formulas are calculated as follows:

$$P=\sum_{i=1}^{N}\frac{/A_i \cap B_i/}{/A_i/}\Big/N \tag{4}$$

$$R=\sum_{i=1}^{N}\frac{/A_i \cap B_i/}{/B_i/}\Big/N \tag{5}$$

$$F=2*P*R/(P+R) \tag{6}$$

Where "N" is the length of the test document, and "A_i" represents the keyword set extracted by the various extraction algorithms in the *i*-th document, "B_i" represents the set of keywords marked by the author in the *i*-th document.

4.2.3. *Result Analysis*

For the four keyword extraction methods, we compared the average of the precision rate P, recall rate R and F-score of the first 5 keywords, the first 7 keywords and the first 10 keywords. The specific comparison results are shown from Table 1 to Table 3.

Table 1. Average precision rate P

Keyword extraction	**5**	**7**	**10**
TF-IDF	0.2060	0.1729	0.1520
TextRank	0.2668	0.2378	0.2192
UNT-TextRank	0.2762	0.2516	0.2238
V-TextRank	0.3043	0.2840	0.2503
Relative TF-IDF improvement	47.71%	64.26%	64.67%
Relative TextRank improvement	28.60%	19.40%	14.20%
Relative UNT-TextRank improvement	10.20%	12.90%	11.80%

Table 2. Average Recall rate R

Keyword extraction	**5**	**7**	**10**
TF-IDF	0.2234	0.2606	0.3262
TextRank	0.2832	0.3421	0.4524
UNT-TextRank	0.2832	0.3566	0.4520
V-TextRank	0.3136	0.4057	0.5083
Relative TF-IDF improvement	40.37%	55.68%	55.82%
Relative TextRank improvement	10.70%	18.60%	12.40%
Relative UNT-TextRank improvement	10.70%	13.80%	12.50%

Table 3. Average F-score

Keyword extraction	**5**	**7**	**10**
TF-IDF	0.2116	0.2050	0.2048
TextRank	0.2686	0.2766	0.2921
UNT-TextRank	0.2740	0.2911	0.2960
V-TextRank	0.3033	0.3298	0.3319
Relative TF-IDF improvement	43.33%	60.88%	62.06%
Relative TextRank improvement	12.90%	19.20%	13.60%
Relative UNT-TextRank improvement	10.70%	13.30%	12.10%

As can be seen from Table 1 to Table 3, the precision of the four methods shows a downward trend when the number of extracted keywords increases, and the extraction performance is better when 5 keywords are extracted. Recall and F-score value increase with the increase of the number of keyword.

The TF-IDF algorithm is relatively stable when extracting keywords, and its accuracy is not high. Based on this, the V-TextRank algorithm improves the precision rate by at least 47.71%, the recall rate by at least 40.37%, and the F-score by at least 43.33%.

The traditional TextRank algorithm is more effective in extracting keywords than the TF-TDF algorithm based on word frequency. V-TextRank algorithm has a certain improvement on the traditional TextRank algorithm. The V-TextRank algorithm improves the precision rate by at least 14.20%, the recall rate by at least 10.70%, and the F-score by at least 12.90%.

Comparing the similar UNT-TextRank algorithm without the title factor and the V-TextRank algorithm with the title factor, we found that after considering the title factor, both the precision rate, the recall rate, and the F-score have improved to varying degrees. Compared with the UNT-TextRank algorithm, the V-TextRank algorithm improves the precision rate by at least 10.20%, the recall rate by at least 10.70%, and the F-score by at least 10.70%.

The comparison of the above experimental results shows that the keyword extraction method based on Word2Vec and TextRank in this research has made certain improvements in keyword extraction.

4.2.4. Analysis of Extraction Error Results

The extraction results of the two articles in Table 4 are completely inconsistent with the keywords marked by the author, but there are some differences between the results.

Table 4. Example of completely unmatched keyword extraction results

Document number	**30**	**167**
Author Annotated Results	Open-plan office, Work performance, Irrelevant sound effect, Background speech, Office noise, Room acoustics	Deep learning, Classification architecture for echo video images, Convolutional neural network, Echocardiography, KAZE, SIFT, SURF
TF-IDF	Level, rating, acoustical, Condition, Offices	Flow, accuracy, networks, rate
TextRank	Rating level, number recall task, speech impairs, pressure, percentile	accuracy rate whereas, information, acceleration measurement, time direction
UNT-TextRank	rating level, number recall task, speech impairs, percentile, pressure	accuracy rate whereas, information, acceleration measurement, time direction
V-TextRank	percentile level, speech annoyance, number recall task, perception, impairs	accuracy rate whereas, framework, cnn, motion flow technique, deep learning architecture

The TD-IDF algorithm has poor extraction results because it only considers the frequency of words. Both the TextRank and UNT-TextRank algorithms perform clustering based on the co-occurrence relationship between words. There is not much difference between the two in the extraction results, but the weight of individual keywords will be different. It can be seen from the extraction results that some commonly used and unclear words such as "rate", "level", etc. have too high weights that cause the ranking structure to be inaccurate. Although the V-TextRank algorithm has some deviations from the keywords marked by the author, but the extraction results are semantically close to the keywords marked by the author.

Comprehensive analysis shows that after using the extraction algorithm combining TextRank and Word2Vec, there is still a chance that the occurrence of synonyms can lead to inaccurate extraction results, but it can be improved in subsequent research.

5. CONCLUSION

This paper proposes a keyword extraction method based on Word2Vec and TextRank algorithms, where the Word2Vec is used to obtain the semantic representation of documents and then the cosine vector distance in Word2Vec is used to construct the new probability transition matrix for TextRank. This method effectively solves some extraction errors caused by TextRank algorithm only using the target document. At the same time, it solves the semantic fault caused by the incoherence between sentences in the sliding window, and the title weight is increased to improve the quality of keyword extraction. The experimental results show that the combination of Word2Vec and TextRank algorithms can effectively optimize the performance of keyword extraction, and our algorithm outperforms the baseline methods in the terms of precision, recall and F-score.

There are still some shortcomings when extracting keywords in this method. For example, some of the keywords marked by the author do not appear in the abstract or title, which has a certain negative impact on the extraction of keywords. Meanwhile one of the side effects of using external documents is that it increases the possibility of polysemy and leads to some deviations in the extraction results. Therefore, how to extract new words as keywords based on semantic information and reduce the interference of polysemy is also an important work in future research.

6. ACKOWLEDGEMENTS

This work was partially supported by National Natural Science Foundation of China (61977032) and Project of State Language Commission of China (ZDI135-99). We would thank the anonymous reviewers for their hard working. Fen Chen is the corresponding author.

7. REFERENCES

[1] Mihalcea, Rada., and Tarau, Paul. 2004. Textrank: bringing order into texts. *Emnlp*, 404-411. DOI= http://dx.doi.org/

[2] Tian, Xia. 2013. Study on keyword extraction using word position weighted textrank. *New Technology of Library and Information Service*. 237 (09): 30-34.

[3] Jianfei, Ning., and Jiangzhen, L. 2016. Using word2vec with text rank to extract keywords. *New Technology of Library and Information Service*.271 (06): 20-26.

[4] Qifei, Liu., and Weiyu, Sheng. 2018. Research of keyword extraction of political news based on word2vec and textrank. *Information Research*. 248 (06): 26-31. (In Chinese)

[5] Zhang, Kuo., Xu, Hui., Tang, Jie., and Juangzi, Li. 2006. Keyword Extraction Using Support Vector Machine. *International Conference on Advances in Web-age Information Management*. Springer-Verlag.85-96. DOI= https://doi.org/10.1007/11775300_8

[6] Liang, Hu., Lei, Xia., and Wei, Li. 2017. Keyword Extraction System Based on Improved TF-IDF Algorithm. *Journal of Xiamen University of Technology*. 25 (05): 73-78. (In Chinese)

[7] Qiang, Jia., et al. 2017. Research on Improved TF-IDF Text Feature Word Extraction Algorithm. *Journal of Liaoning University of Petroleum & Chemical Technology*. 37 (4): 23-29. (In Chinese)

[8] Blei, David M., et al. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*. 3993-1022.

[9] Alokaili, Areej., Aletras, Nikolaos., Stevenson, Mark. 2019. Re-ranking words to improve interpretability of automatically generated topics. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*. 43-54.DOI=10.18653 / v1 / W19-0404

[10] Yijun, Guang., and Tian, Xia. 2014. Study on keyword extraction with lda and textrank combination. *New Technology of Library and Information Service*. 41-47.

[11] Negi, Sumit. 2014. Document Keyphrase Extraction Using Label Information. In *Proceedings of {COLING} 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 1468-1476.

[12] Jun,Chen., et al. 2018. Keyphrase Generation with Correlation Constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4057–4066.DOI=10.18653/v1/D18-1439

[13] Nicosia, Massimo., and Moschitti, Alessandro. 2018. Semantic Linking in Convolutional Neural Networks for Answer Sentence Selection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.1070-1076.DOI= 10.18653/v1/D18-1133

[14] Timothy, Niven. and Hung-Yu, Kao. 2019. Probing Neural Network Comprehension of Natural Language Arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4658-4664. DOI= 10.18653/v1/P19-1459

[15] Guangyi, Li., and Houfeng, Wang. 2014. Improved automatic keyword extraction based on textrank using domain knowledge. *Communications in Computer & Information Science*. 496:403-413.

[16] Xiao-Lei, Bai., Guang-Jun, Huang., and Jian-Hui, Duan. 2014. A keyword extraction method based on bp neural network. *Journal of Hefei University of Technology (Natural Science)*. 37(07):807-811.

[17] Johannes, Villmow., Marco, Wrzalik., Dirk, Krechel. 2018. Automatic Keyphrase Extraction Using Recurrent Neural Networks. M. *Machine Learning and Data Mining in Pattern Recognition*.

[18] Page Lawrence., et al. 1999.The PageRank Citation Ranking: Bringing Order to the Web. R. *Stanford InfoLab*.