An Improved TextRank Keywords Extraction Algorithm*

Suhan Pan
College of Information Engineering, Yangzhou University
Yangzhou, Jiangsu, China
yzdxpsh@163.com

Zhiqiang Li†
College of Information Engineering, Yangzhou University
Yangzhou, Jiangsu, China
yzqqLzq@163.com

Juan Dai
College of Information Engineering, Yangzhou University
Yangzhou, Jiangsu, China
yzdxdaij@163.com

ABSTRACT

Keywords extraction is widely used in the field of natural language processing. How to quickly and accurately extract keywords has become the key issue in text processing. At present, there are many methods for keywords extraction, but the accuracy and versatility of the method still have much room for improvement. Thus, an improved TextRank keywords extraction algorithm is proposed in this paper. The algorithm uses the TF-IDF algorithm and the average information entropy algorithm to calculate the importance of words, and then calculates the comprehensive weight of words based on the calculation results in the text. The initial weight of the TextRank algorithm node and the node probability transfer matrix are improved by using the comprehensive weight of words, and the weights of all nodes are iteratively calculated until convergence. The weights of the nodes are sorted to obtain the weight information of the words, then the top N words are selected as the keywords.

Finally, the keywords extraction function is realized by outputting the keywords. The experimental results show that compared with the traditional TF-IDF method and TextRank method, the improved TextRank keyword extraction method proposed in this paper is more general and its accuracy of extracting keywords is higher.

*Article Title Footnote needs to be captured as Title Note
†Author Footnote to be captured as Author Note

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ACM TURC-AIS'19, 17-19 May 2019, Chengdu, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7158-2/19/05…$15.00
https://doi.org/10.1145/3321408.3326659

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; *Language resources*;

KEYWORDS

Keywords extraction, TF-IDF algorithm, TextRank algorithm, Average information entropy, Natural language processing

ACM Reference Format:
Suhan Pan, Zhiqiang Li, and Juan Dai. 2019. An Improved TextRank Keywords Extraction Algorithm. In *TURC-AIS 2019: ACM TURC Conference on Artificial Intelligence and Security, May 17-19, 2019, Chengdu, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3321408.3326659

1 INTRODUCTION

With the rapid development of the Internet, the data volume of network resources is increasingly large. In the face of massive text data, how to effectively and accurately retrieve the article content has been a research hotspot. When analyzing an article, it is common to start with the keywords. The keywords of an article can not only summarize the theme of the article, but also reflect the main content and emotional tendency of the whole article. Therefore, accurate and fast extraction of keywords is crucial for text clustering, text summarization and information retrieval.

In recent years, relevant scholars at home and abroad have carried out a lot of research work in the field of keywords extraction technology, and also have put forward a lot of keywords extraction algorithms. Among them, the main algorithms are keywords extraction based on implied subject model (LDA) [2], keywords extraction based on TF-IDF word frequency statistics [9] and keywords extraction based on word graph model (TextRank) [10]. The above three algorithms have been widely used for their simplicity and efficiency.

Among them, the keywords extraction algorithm based on the implied topic model calculates the importance of words according to the similarity of the topic distribution of documents and words. Since this method usually needs to train

the corpus content to get relevant information, the quality of keywords extracted by this method is greatly affected by the topic distribution of the training corpus. The keywords extraction algorithm based on TF-IDF word Frequency statistics (Term Frequency-Inverse Document Frequency) is a common statistical extraction method, which mainly judges the importance of words to articles by calculating the word Frequency. In the process of keywords extraction, this method relies too much on word frequency features and ignores semantics, context and other features. For this reason, other characteristic factors are often introduced to reduce the dependence on word frequency. TextRank algorithm is a graph-based sorting algorithm, which uses the relationship between local words (co-occurrence window) to sort the subsequent keywords and extract the keywords directly from the text itself. TextRank algorithm has the advantages of simple implementation, unsupervised, weak language correlation, and is suitable for single text and multi-text processing. But this method does not investigate whether the importance of different words will affect the problem of the adjacent nodes weight transfer, and ignore the whole information document corpus. The weight of words information has no actual meaning, and cannot distinguish between the strength of the connection. The connection between words is only determined by the use of a sliding window within a single sentence, without considering the context as a whole.

In order to further improve the effect and quality of keywords extraction, many scholars have optimized the above algorithms. Lang Dongdong [8] et al. proposed a key phrase extraction method based on LDA and TextRank. Gu Yijun [3] et al. combined LDA with TextRank, so that the importance of the candidate node words was non-uniformly transferred according to the document set topic distribution. However, the results are greatly influenced by the subject distribution of training corpus. Zhang Jin et al. [7], Xie Jin et al. [6] considered improving TF-IDF weight based on word position and word span, or using semantic coherence and combining word frequency and position characteristics to carry out weighting [15]. At the same time, some scholars introduced the method of information entropy [5].However, these methods all have some computational complexity problems or limitations in the type of articles and corpus size. Some scholars combined the comprehensive information in the article with the method of citing news category factors [4, 13, 16], and added other characteristic factors to weight, which can correct the word frequency dependence problem to a certain extent. However, the method does not take into account the influence of the part of speech and the different coverage of keywords. Biswas S K [1] et al., Yan Ying [17] et al. proposed a keywords extraction method based on graph, which took into account the context, location, centrality, part of speech and other characteristics of the words respectively, modified the initial weight of the words, etc., and achieved a good extraction effect.

For all the questions raised above, this paper comprehensively considers the importance of words to a single document and document set for keywords extraction based on the inspiration of literature [1, 17]. We use the TF-IDF method and average information entropy to comprehensively calculate the importance of words for single document and document set, and then calculate the comprehensive weight of words to improve the initial weight and probability transfer matrix of TextRank vocabulary nodes. Experiments show that the improved method improves the accuracy of keywords extraction.

2 RELEVANT TECHNOLOGIES

This paper mainly introduces the relevant algorithms needed in the process of keywords extraction, including the TF-IDF algorithm and the average information entropy algorithm. These two algorithms are mainly used to calculate the importance of words for a single document and a set of documents. The following is an introduction to the principles of these algorithms:

2.1 TF-IDF Algorithm

The basic idea of TF-IDF algorithm: using Term Frequency (TF) and Inverse Document Frequency (IDF) to get the weight value of words by multiplying them [12]. According to the TF-IDF algorithm, the calculation formula of $W_{\text{TF-IDF}}(i)$ is as follows:

$$W_{\text{TF-IDF}}(i) = TF_i * IDF_i \tag{1}$$

$$IDF_i = \log\left(N/DF_i\right) \tag{2}$$

In the calculation formula (1) (2), TF_i represents the number of times the word i appears in the document content division by the total number of words in the document content, that is, the frequency of the word i appears in the document; N represents the total number of documents in the corpus; DF_i represents the number of documents containing the word i; IDF_i can be calculated by formula (2), it represents the category discrimination ability of word i.

According to the above formulas, when there is high frequency words in the document, and in the number of documents containing the word frequency is low, then the words according to the TF - IDF algorithm to get the weight value of $W_{\text{TF-IDF}}(i)$ is the higher, at this time, the words can express the content of the article to some extent. On the other hand, it shows words are not the important words, not the main content of the article.

2.2 Average Information Entropy Algorithm

The basic idea of average information entropy is: according to the frequency of word frequency in different documents, the importance of all words to a single document and a document set is calculated by combining with the overall corpus. The average information entropy can be used to measure the equilibrium degree of word distribution in the whole document. According to the average entropy algorithm, the formula for calculating the weight of the word $W_{Entropy}(i)$ is as follows:

$$W_{Entropy}(i) = 1 - \frac{1}{\log N} \sum_{k=1}^{N} \left[\frac{f_{wk}}{n_w} \log \left(\frac{n_w}{f_{wk}} \right) \right] \quad (3)$$

In the calculation formula (3), f_{wk} represents the frequency of the word w in document k; n_w stands for the frequency of the word w in the entire document set; N represents the total number of documents.

If the word i occurs with equal frequency in the various documents, the value of $W_{Entropy}(i)$ is close to the minimum value of 0, indicating that it is not sufficiently expressive for the document topic. On the other hand, if the word i varies widely in the various documents, the value of $W_{Entropy}(i)$ is close to the maximum value of 1, indicating that it is very expressive of the document topic.

3 KEYWORDS EXTRACTION ALGORITHM

3.1 Algorithm Description

The traditional TextRank algorithm is a graph-based unsupervised method for generating keywords for text. The PageRank algorithm is a link analysis algorithm. Google uses PageRank algorithm to calculate the ranking of web pages, which can be used to measure the importance of web pages. The main idea of PageRank algorithm is to calculate the number and quality of a web page links, so as to estimate the importance of this page. The idea is based on the assumption that more important pages will receive more links from other pages. Inspired by PageRank algorithm, the main idea of TextRank algorithm is to divide the document into several text units. These text units are used as nodes, and the similarity between the nodes is used as edges, and then a text graph is formed. Finally, the nodes are sorted by matrix iterative convergence, and the keywords corresponding to the text are obtained.

The graph of the TextRank algorithm is constructed as follows [14]. It is assumed that $V = \{V_1, V_2, \ldots, V_n\}$ is a set of n elements $V_i (1 \le i \le n)$. By using each V_i in V as the node and the similarity between the nodes as the edge, an undirected TextRank text graph $G = (V, E, W)$ can be constructed, where $E \subseteq V \times V$ is nonempty finite set of each edge between the nodes. It is denoted as $E = \{(V_i, V_j) | V_i \in V \wedge V_j \in V \wedge w_{ij} \in W \wedge w_{ij} \neq 0\}$; $W = \{w_{ij} | 1 \le i \le n \wedge 1 \le j \le n\}$ is the weight set of edges, and w_{ij} is the weight value of edges between node V_i and V_j.

According to the constructed text graph $G = (V, E, W)$, an $n \times n$ similarity matrix $SD_{n \times n}$ between nodes can be obtained:

$$SD_{n \times n} = \begin{bmatrix} w_{11} & \ldots & w_{1n} \\ \ldots & \ldots & \ldots \\ w_{n1} & \ldots & w_{nn} \end{bmatrix}$$

Obviously, the matrix $SD_{n \times n}$ is a symmetric matrix, that is, the contribution of node V_i to V_j is the same as that of node V_j to V_i, and the values of elements on the diagonal of $SD_{n \times n}$ are all 1.

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{W_{ji}}{\sum_{V_k \in Out(V_j)} W_{jk}} WS(V_j) \quad (4)$$

Where $WS(V_i)$ is the weight value (PR value) of the node V_i; d is the damping coefficient, which is generally set at 0.85. It means that it can represent the probability of the current node jumping to any other node, and at the same time can enable the weight to be transferred to the convergence stably. In (V_i) is the collection of all nodes pointing to node V_i; Out (V_i) is the set of all nodes pointed by node V_i. The sum of the right side in the formula (4) indicates the contribution of each adjacent node to the node. The summed numerator w_{ij} represents the degree of similarity between the two nodes V_i and V_j and the denominator is a weighted sum. $WS(V_j)$ represents the weight value of node V_j after the last iteration.

The traditional TextRank algorithm text graph model is shown as follows:

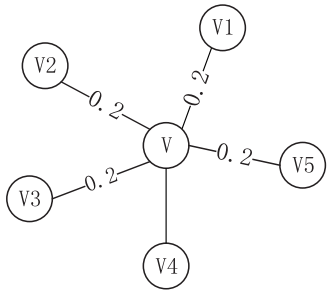

Figure 1: Traditional TextRank algorithm text graph model.

In traditional TextRank algorithm, the initial weight of each node is 1, that is, the initial weight of the node is the same, and the weight of the node is transferred evenly. That is, $B_0 = (1/|V|, 1/|V|, 1/|V|, \ldots, 1/|V|)^T$, then generally converges after several iterations of calculation: $B_i = SD_{n \times n} \cdot B_{i-1}$.

However, according to the research, the method of weighted word transfer probability based on word importance can effectively improve the effect of keywords extraction [11, 18]. For the initialization problem of word nodes, the improved method proposed in this paper is based on the importance of words.

Therefore, the TF-IDF algorithm and the average information entropy algorithm are selected to calculate the importance of words in this paper, and the comprehensive weight of words is used to improve the initial weight and probability transfer matrix of TextRank vocabulary nodes.

In the automatic extraction of the text keywords, the weight values calculated according to the improved algorithm are sorted, and the N words with the highest importance are extracted as the keywords of the text.

For any word i in the text, its comprehensive weight calculation formula is defined as follows: the given graph G and similarity matrix $SD_{n\times n}$, the weight of each node can be calculated iteratively. The calculation formula is as follows:

$$W_{\text{Weight}}(i) = \frac{1}{2}W_{\text{TF-IDF}}(i) + \frac{1}{2}W_{Entropy}(i) \tag{5}$$

In the formula (5), $W_{\text{TF-IDF}}(i)$ is the weight value of words calculated by TF-IDF algorithm; $W_{Entropy}(i)$ is the weight value of words calculated by average information entropy algorithm; $W_{Weight}(i)$ is the average entropy weight of the word. Through multiple comparisons of experimental results, when the weights of the TF-IDF algorithm and the average information entropy algorithm are each 0.5, the extracted keyword results have the highest accuracy. Therefore, in the algorithm of this paper, the weight is set to 0.5.

Table 1: Algorithm weight comparison analysis table.

Weight of TF-IDF	Weight of Entropy	Precisoin%	Recall%	F1-Measure%
0.3	0.7	50.97	29.08	37.03
0.4	0.6	54.46	29.87	38.58
0.5	0.5	55.52	31.33	40.06
0.6	0.4	53.88	30.22	38.72
0.7	0.3	51.23	30.03	37.86

According to TextRank algorithm, the calculation formula of the transition probability between nodes is:

$$W(V_j, V_i) = \frac{W_{Weight}(V_i)}{\sum_{V_k \in Out(V_j)} W_{Weight}(V_k)} \tag{6}$$

According to formula (6), the weight iteration formula of nodes is as follows:

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} W(V_j, V_i) WS(V_j) \tag{7}$$

In the formula (6), (7). For a given point V_i, In (V_i) is the set of points pointing to that point, and Out(V_i) is the set of points are pointed to by point V_i. In formula (6), $W(V_j, V_i)$ represents the transition probability of the edge of node V_j to node V_i, and the score of point V_i is calculated as formula (7). The d in the formula (7) is the damping coefficient, ranging from 0 to 1. The probability of representing a point from a particular point in the graph to any other point is generally 0.85. In the calculation, the weight value is calculated according to formula (5) as the initial value, and recursively calculate until convergence. That is, the error rate of any point in the graph is less than the given limit value, and the convergence can be reached. Generally, the limit value is 0.0001. The score of each point in the graph can be obtained by calculating, that is the score of the word.

3.2 Keywords Extraction

The extraction process of the improved TextRank keywords extraction algorithm is shown in the figure as follows.

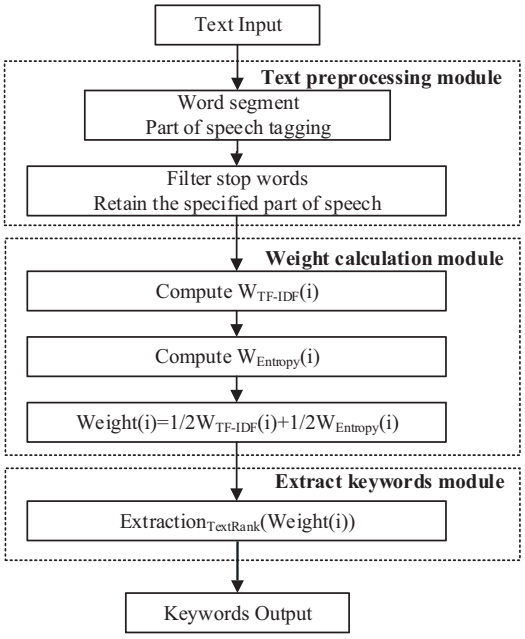

Figure 2: Flow chart of TextRank keywords extraction based on the improvement.

For the given input text, the keywords extraction steps are as follows:

Step 1: Text preprocessing. Divide the text into words, mark the part of speech, keep only nouns, proper nouns, verbs, adjectives and adverbs, and delete the stop words in the text.

Step 2: Weight calculation. $W_{\text{TF-IDF}}$, $W_{Entropy}$ for each word in the text is calculated and then the composite weight W_{Weight} is calculated.

Step 3: Keywords extraction. An improved TextRank model is constructed based on the weighted node initial value and node probability transfer matrix based on the comprehensive weight of words, and the first N words with relatively large weight are selected as keywords for final calculation and output.

4 EXPERIMENTAL RESULTS AND ANALYSIS

The experimental datum in this paper are news data randomly selected from various major portal websites, including word number of news content, topic and other information. Each news article is saved as a document, and a total of 500 documents constitute the corpus. For the data set, keywords were extracted in the form of multi-person manual cross annotation, and 5, 7, 10 keywords were extracted for each document.

For the same data set, different quantities of keywords were extracted. In the experiment, the traditional TF-IDF algorithm, TextRank algorithm (the size of co-occurrence window is 7) and the improved TextRank algorithm proposed in this paper were used for cross comparison. In this paper, Precision (P), Recall rate (R) and F1-Measure are selected as the performance indexes to evaluate keywords extraction. The formula for calculating the indexes is as follows.

$$Precision = \frac{1}{N}\sum_{i=1}^{N}\frac{|P_i \cap T_i|}{|P_i|} \tag{8}$$

$$Recall = \frac{1}{N}\sum_{i=1}^{N}\frac{|P_i \cap T_i|}{|T_i|} \tag{9}$$

$$F1 - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

Where P_i is the extracted keywords from the algorithm and T_i is the actual keywords extracted by manual annotation.

Precision (P) is the ratio between the size of the intersection of the keywords extracted by the algorithm and the keywords actually extracted manually and the size of the keywords extracted by the algorithm. Precision gives a measurement of how relevant the extracted key words are to the actual desired key words.

Recall (R) is the ratio between the intersection size of the keywords extracted by the algorithm and the actual manually extracted keywords and the actual manually extracted keywords. Recall gives a measurement of how much of the relevant data was accurately extracted.

F1-Measure is a comprehensive measure of the combination of P and R. This value can more directly reflect the extraction effect of the keywords extraction algorithm.

4.1 Experimental Results

The experimental results are shown in table 2. According to the analysis of experimental results in table 2, it can be found that the method proposed in this paper is superior to the traditional TF-IDF and TextRank algorithms in keyword extraction.

4.2 Experimental Analysis

According to the experimental results, the cross-comparison figures of experimental results are shown as follows.

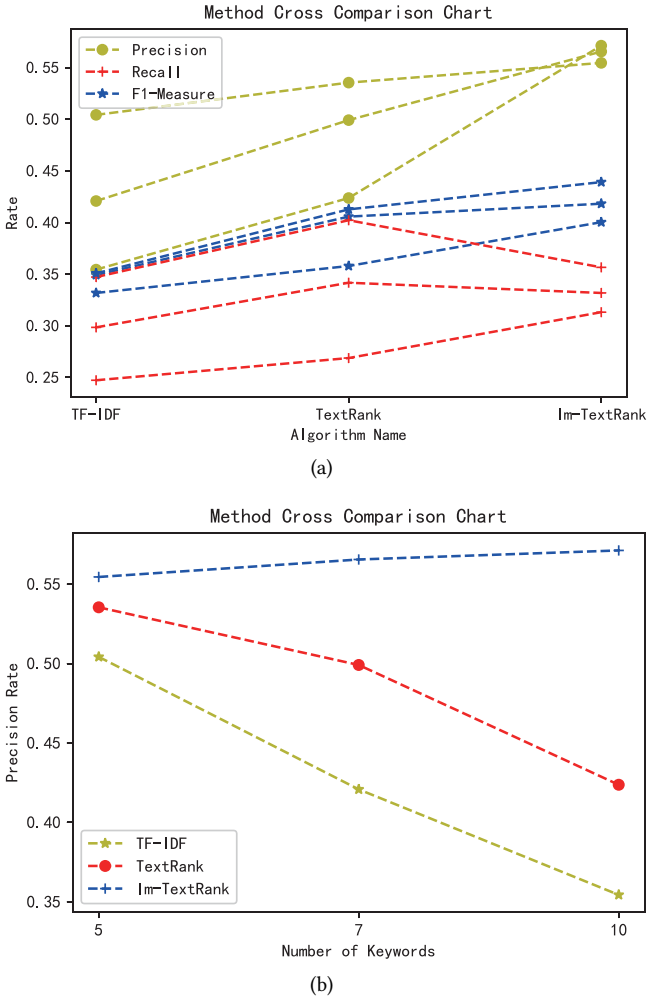

Figure 3: Methods cross comparison chart.

By observing the comparison diagrams, it can be found that with the increase of the number of keywords extracted,

Table 2: Cross comparison and analysis table of experimental results.

Algorithms	Num of keywords	Precisoin%	Recall%	F1-Measure%
TF-IDF	5	50.4	24.7	33.1
TextRank		53.5	26.8	35.7
Im-extank		55.4	31.3	40.0
TF-IDF	7	42.0	29.8	34.9
TextRank		49.9	34.1	40.5
Im-extank		56.5	33.1	41.8
TF-IDF	10	35.4	34.7	35.0
TextRank		42.3	40.2	41.2
Im-extank		57.1	35.6	43.9

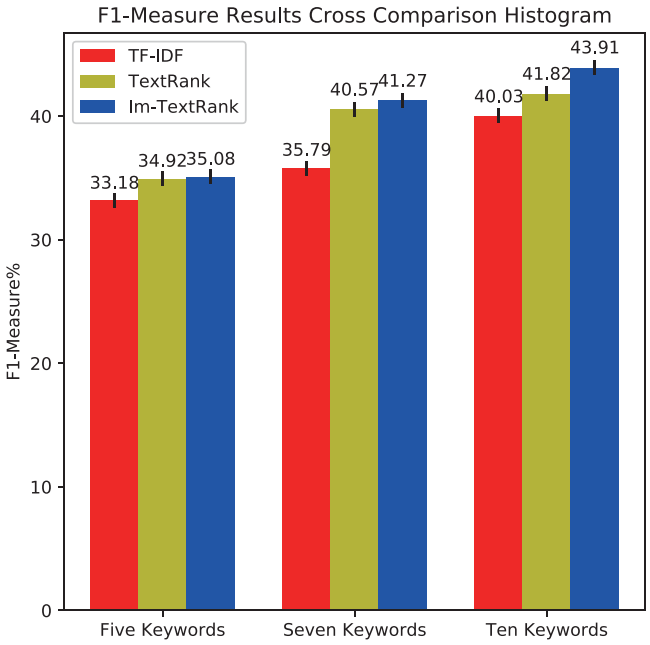

Figure 4: F1-Measure results cross comparison histogram.

the accuracy of traditional TF-IDF algorithm and TextRank algorithm presents a declining trend. As shown in figure 3, the improved TextRank algorithm proposed in this paper has a relatively stable accuracy rate as the number of keyword extraction changes. Moreover, the proposed algorithm precision rate is also significantly better than the other two algorithms. According to the calculation of the comprehensive weight of words, keywords are relatively concentrated and prominent, and the weight is relatively large.

In addition, in the algorithm experiment comparison, it can be found that the size of the co-occurrence window of the traditional TextRank algorithm also has an impact on the accuracy. Because the size of co-occurrence window determines the density of the weight transfer probability matrix, it affects the result of keyword extraction. When the co-occurrence window is 7, the keyword extraction accuracy is higher than that when the co-occurrence window is 5. Therefore, when comparing the results in this paper, the co-occurrence window is 7.

In general, it can be clearly seen from the histogram of figure 4 that the improved method proposed in this paper is higher than the traditional method in terms of F1 value (F1-Measure). Since F1-Measure is a comprehensive measurement index combining precision (P) and recall (R), it mainly refers to F1-Measure for comparison of experimental results. This basically proves that the method proposed in this paper is better than the traditional TF-IDF and TextRank algorithms in terms of keywords extraction.

5 CONCLUSION AND PROSPECT

Keywords in an article can not only summarize the theme of the article, but also reflect the main content and emotional tendency expressed in the whole article. Therefore, the results of keyword extraction by the algorithm need to reflect the theme content of the article relatively accurately. Therefore, accurate and fast extraction of keywords is crucial for text clustering, text summarization and information retrieval.

This paper proposes an improved algorithm to solve the problem that traditional TextRank algorithm ignores the importance of the words themselves and the overall information of the document. In this method, word frequency and average information entropy are selected as the characteristics to calculate the importance of words, and the initial weight and probability transfer matrix of the lexical nodes of TextRank algorithm are improved according to the comprehensive weight of words obtained from the calculation. The improved algorithm improves the accuracy of keyword

extraction, and the operation is simple, without training and manual intervention. It has a strong universality, can meet the demand for keyword extraction for general articles.

The algorithm proposed in this paper can be further improved by combining more word features and semantic environment of word context. This will be the main direction of the following research.

ACKNOWLEDGMENTS

I would like to express my gratitude to all who have helped me during the writing of the thesis. I gratefully acknowledge the help of my tutor Professor Zhiqiang Li. I do appreciate his patience, encouragement, and professional instructions during my thesis writing. Also, I would like to thank Juan Dai, who helped me modify format and proofread the paper.

Last but not least, my gratitude also extends to the conference organizer who gave me a chance to show my research results.

REFERENCES

[1] Saroj Kr. Biswas, Monali Bordoloi, and Jacob Shreya. 2018. A graph based keyword extraction model using collective node weight. *Expert Systems with Applications* 97 (2018), 51–59.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, 01 (2003), 993–1022.

[3] Yijun Gu and Tian Xia. 2014. Study on Keyword Extraction with LDA and TextRank Combination. *New Technology of Library and Information Service* 30, zl (2014), 41–47.

[4] Li Hang, Chaolan Tang, Yang Xian, and Wanting Shen. 2017. Research on keyword extraction of political news based on Word2Vec and TextRank. *Information Research* 248, 06 (2017), 26–31.

[5] Li Hang, Chaolan Tang, Yang Xian, and Wanting Shen. 2017. TextRank Keyword Extraction Based on Multi Feature Fusion. *Journal of Intelligence* 36, 08 (2017), 183–187.

[6] Xie Jin. 2012. A Method of Automatic Keyword Extraction Based on Word Span. *Modern Property Management* 11, 04 (2012), 108–111.

[7] Zhang Jin. 2014. A Method of Intelligence Key Words Extraction Based on Improved TF-IDF. *Journal of Intelligence* 33, 04 (2014), 153–155.

[8] Dongdong Lang, Chenchen Liu, Xupeng Feng, Lijun Liu, and Qingsong Huang. 2018. design and implemention of a key phrases extraction scheme in the text based on lda and textrank. *Computer Applications and Software* 03 (2018), 54–60.

[9] Juanzi Li, Kuo Zhang, et al. 2007. Keyword extraction based on tf/idf for Chinese news document. *Wuhan University Journal of Natural Sciences* 12, 05 (2007), 917–921.

[10] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

[11] Y. U. Shan-Shan, S. U. Jin-Dian, and L. I. Peng-Fei. 2016. Improved TextRank-based Method for Automatic Summarization. *Computer Science* 43, 6 (2016), 240–247.

[12] Congying Shi. 2009. Study of TFIDF algorithm. *Journal of Computer Applications* 29, 6 (2009), 167–170.

[13] Y. Wang, C. Mao, Z. Yu, J. Guo, and L. Luo. 2016. Approach for topical sentence of news events extraction based on graph. 40, 04 (2016), 438–443.

[14] Tian Xia. 2013. Study on Keyword Extraction Using Word Position Weighted TextRank. *New Technology of Library and Information Service* 9 (2013), 30–34.

[15] H. U. Xue-Gang, L. I. Xing-Hua, Fei Xie, and W. U. Xin-Dong. 2010. Keyword Extraction Based on Lexical Chains for Chinese News Web Pages. *Pattern Recognition and Artificial Intelligence* 123, 01 (2010), 45–51.

[16] Yang Yan. 2011. Exploration and improvement in keyword extraction for news based on TFIDF. 13 (2011), 3551–3556.

[17] Yan Ying, Qingping Tan, Qinzheng Xie, Zeng Ping, and Panpan Li. 2017. A Graph-based Approach of Automatic Keyphrase Extraction. *Procedia Computer Science* 107 (2017), 248–255.

[18] Jinzhang Zhou. 2019. Keyword extraction method based on word vector and TextRank. *Application Research of Computers* 36, 5 (2019).