TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 10/13 pp886–893
DOI: 10.26599/TST.2020.9010051
Volume 26, Number 6, December 2021

News Keyword Extraction Algorithm Based on Semantic Clustering and Word Graph Model

Ao Xiong, Derong Liu, Hongkang Tian, Zhengyuan Liu, Peng Yu*, and Michel Kadoch

Abstract: The internet is an abundant source of news every day. Thus, efficient algorithms to extract keywords from the text are important to obtain information quickly. However, the precision and recall of mature keyword extraction algorithms need improvement. TextRank, which is derived from the PageRank algorithm, uses word graphs to spread the weight of words. The keyword weight propagation in TextRank focuses only on word frequency. To improve the performance of the algorithm, we propose Semantic Clustering TextRank (SCTR), a semantic clustering news keyword extraction algorithm based on TextRank. Firstly, the word vectors generated by the Bidirectional Encoder Representation from Transformers (BERT) model are used to perform k-means clustering to represent semantic clustering. Then, the clustering results are used to construct a TextRank weight transfer probability matrix. Finally, iterative calculation of word graphs and extraction of keywords are performed. The test target of this experiment is a Chinese news library. The results of the experiment conducted on this text set show that the SCTR algorithm has greater precision, recall, and F1 value than the traditional TextRank and Term Frequency-Inverse Document Frequency (TF-IDF) algorithms.

Key words: keyword extraction; TextRank; semantics; word vector

1 Introduction

At present, with the rapid development of the internet, the speed of information dissemination is becoming increasingly fast. An abundance of data-based news are spreading quickly every day, and people can easily obtain the latest information. However, such advancement also has disadvantages, including the spread of irrelevant news on the internet. The rapid extraction of the core views and main content of news has become an important demand. Therefore, analysis and processing of news texts and labeling or extraction of keywords have become research hot spots in the field of natural language processing. An efficient and accurate keyword extraction algorithm not only reduces the cost of screening and filtering effective information but also accelerates the dissemination and circulation of information in the network.

A preferred keyword extraction algorithm is to calculate the feature weight of words on the basis of Term Frequency-Inverse Document Frequency (TF-IDF)[1, 2]. In specific, high-frequency words are identified using word frequency or Term Frequency (TF). The importance of words with no representativeness is reduced in high-frequency words to the text, and then the accuracy of keyword extraction is improved. TextRank[3, 4], which is based on network graph, is a classic unsupervised keyword extraction method. This method decomposes the content of a single document into a network graph model by word segmentation and

• Ao Xiong, Derong Liu, Hongkang Tian, Zhengyuan Liu, and Peng Yu are with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: xiongao@bupt.edu.cn; liuderong@bupt.edu.cn; 386065557@qq.com; 18004408557@163.com; yupeng@bupt.edu.cn.
• Michel Kadoch is with École de Technologie Supérieure, Université du Québec, Montreal, H3C 1K3, Canada. E-mail: Michel.Kadoch@etsmtl.ca.
∗ To whom correspondence should be addressed.
Manuscript received: 2020-09-09; accepted: 2020-10-09

extracts keywords by considering the structural features and word frequency features of the document. However, the two algorithms only focus on word frequency. In iterative calculation, words with high frequency are given higher weight, thereby omitting keywords with low frequency. Therefore, this paper proposes Semantic Clustering TextRank (SCTR), a semantic clustering news keyword extraction algorithm that is based on the TextRank algorithm. It adopts a new word vector model that generates a word vector for each word to represent the semantic meaning of a word. Then, it is clustered, and the result of clustering is used as the semantic feature of the word. A new transition probability matrix is constructed to extract keywords.

This paper is organized as follows. Related work and analysis are given in Section 2. A detailed analysis of SCTR is presented in Section 3. Experiment and analysis are shown in Section 4, and conclusion and future work are given in Section 5.

2 Related Work

The mainstream methods in the research field of keyword extraction can be roughly divided into two types: supervised and unsupervised.

In supervised algorithms, the idea of binary classification is usually used to solve this problem, that is, to determine whether or not the candidate words in the document are keywords. These algorithms usually require huge amounts of manpower and time to label the keyword corpus in advance, and they also take time to train the keyword discrimination model. Fox example, Chen et al.[5] designed an enhanced smart learning system to recommend resources based on learning style model. In addition, the precision of labeling affects the effectiveness of the model in extracting keywords. Thus, this research focuses on unsupervised algorithms.

Unsupervised algorithms mainly sort the keyword weights through some specified indicators and select keywords on the basis of the sorting results. Among them are representative TF-IDF based on statistical features[1, 2], TextRank based on word graph model[3, 4], and Latent Dirichlet Allocation (LDA) based on topic model[6]. To optimize the effect of algorithm extraction, Luo et al.[7] derived the calculation formula for the number of words of the same frequency in the text through Zipf's law and then determined the proportion of each frequency word in the text by using the calculation formula for the number of words of the same frequency. Finally, the traditional TF-IDF algorithm was improved by using the statistical law of word frequency. Geng et al.[8] combined the word frequency statistical method with the word co-occurrence map to find keywords with low frequency. Gu and Xia[9] improved the extraction of keywords through the effective fusion of LDA and TextRank. Jiang et al.[10] used semantic distance for the density clustering of words to obtain topic-related classes and selected the central word as the topic-related class of keywords. Li et al.[11] fused the various features of words to calculate the comprehensive weight of words, thereby improving the probability transfer model of TextRank, and extracted keywords using an iterative method. Tian et al.[12] proposed an improved Bag-of-Words (iBoW) for person identification.

All of the above methods ignore the most important semantic features of words and the semantic association between words. Therefore, accurate extraction of the semantic features of words has become a research hotspot in keyword extraction and natural language processing.

With the development of deep learning and the emergence of various word vector models in recent years, the semantic features that were originally difficult to represent completely in machine learning algorithms because of high data dimension have become simple, such as LSTM, which was used by Shen et al.[13] to study a deep learning method for Chinese singer identification. Word vector models using deep learning can map words to a relatively low-dimensional vector space. In this vector space, the distance between two points can be regarded as the similarity of the two words, thus avoiding the problem of extracting the semantic characteristics of the word. Word2vec is a representative word vector tool proposed by Google[14]; it uses a shallow neural network to train the corpus to generate word vectors for each word. Researchers also integrated the word vector model into the keyword extraction algorithm. For example, Li et al.[15] used Word2Vec word vectors to achieve word clustering and obtain article keywords. Zhou and Cui[16] and Wen et al.[17] used word vectors to calculate the semantic differences of words to construct the transition probability matrix of TextRank and then iteratively calculate the word graph and extract keywords.

Therefore, using TextRank as a basis, we introduce the semantic features of words for clustering, construct a new transition probability matrix, and extract keywords. In terms of the representation of the semantic features of words, we use a new Bidirectional Encoder Representation from Transformers (BERT) model to generate word vectors. This new model which adopts bidirectional encoder method similar to Long Short-Term Memory (LSTM) solves the polysemy problem that cannot be solved by Word2Vec and can accurately represent word semantics[18–20].

3 Semantic Clustering News Keyword Extraction Based on TextRank

In general, a text corresponds to multiple keywords, and the main content of the news describes a current event. Thus, keywords often belong to multiple topics, and keywords that belong to different topics usually have large semantic differences. From another aspect, the general TextRank algorithm builds a word graph model through the co-occurrence relationship of words, that is, it focuses on the frequency of words. The iterative calculation process is inclined to give a high weight to words with high frequency. As a result, keywords with low frequency are easy to miss. Therefore, we use the semantic difference of words as an innovation point. On the basis of the TextRank algorithm, clustering is performed according to the semantic difference between words, and the clustering results are used to calculate the weight of edges in the word graph and adjust the transition probability matrix. Finally, the final weight of the word is iteratively calculated, and sorting is performed to obtain the keywords. The general process is shown in Fig. 1.

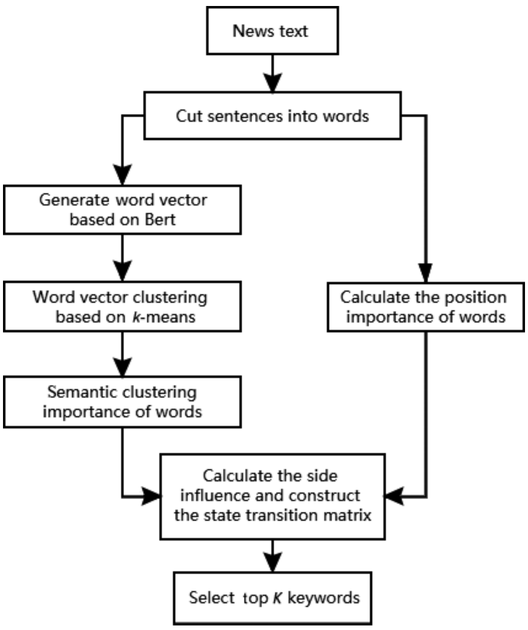

Fig. 1 Algorithm flow.

3.1 Word graph construction of candidate words

According to the basic idea of the TextRank algorithm, the words in a text can be regarded as nodes, so that the target text is constructed into a word graph model according to the co-occurrence relationship between the words, so that the weight of words is calculated according to the word structure in the word graph.

The word graph is constructed as follows:

(1) Split the current text into whole sentences to obtain a set of sentences $S = [s_1, s_2, \ldots, s_m]$.

(2) Subject each sentence S to preprocess, such as word segmentation, part-of-speech tagging, and removal of stop words, and set candidate words $W = [w_1, w_2, \ldots, w_n]$, $w_i \in W$ as the candidate keywords after processing (n is the number of all words in the text).

(3) Treat candidate words as nodes, and construct a word graph of candidate words $G = (V, E)$, where V is the set of candidate keyword nodes, $V = [v_1, v_2, \ldots, v_n]$, and E is the edge set of candidate keywords. The presence or absence of an edge is determined by the co-occurrence relationship between candidate words. The co-occurrence has an edge; otherwise, it does not. For example, when the words w_i and w_j are simultaneously in the co-occurrence window of length l, if the word w_i appears before the word w_j, then an edge $v_i \to v_j$ exists; otherwise, an edge $v_j \to v_i$ exists.

After the initial construction of the word graph, iterative calculation is performed according to Eq. (1):

$$\mathrm{WS}(V_i) = (1-d) + d \times \sum_{V_j \in \mathrm{In}(V_i)} \frac{w_{ji}}{\sum_{V_k} \in \mathrm{Out}(V_j) w_{jk}} \mathrm{WS}(V_j) \tag{1}$$

Let $p_{ji} = \dfrac{w_{ji}}{\sum_{V_k} \in \mathrm{Out}(V_j) w_{jk}}$ represent the jump probability from node j to i, and then Eq. (1) can be expressed as

$$\mathrm{WS}(V_i) = (1-d) + d \times \sum_{V_j \in \mathrm{In}(V_i)} p_{ji} \mathrm{WS}(V_j) \tag{2}$$

where $\mathrm{WS}(V_i)$ is the weight of the node V_i. d is the damping coefficient, and the value range is 0–1, representing the probability of pointing from a particular point to any other point in the word graph. The meaning is to allow the weight to be stably passed to the convergence, generally 0.85. For node V_i, $\mathrm{In}(V_i)$ is the set of points pointing to the point V_i, and $\mathrm{Out}(V_i)$ is the set of points pointed by point V_i.

In consideration that the actual operation process of the TextRank algorithm uses matrix operation, Eq. (2) can be transformed into Eq. (3):

$$B_i = (1 - d) + d \times M \times B_{i-1} \tag{3}$$

where B_i represents the word weight vector composed of the $\mathrm{WS}(V_i)$ of each word node in the i-th iteration and M represents the transition probability matrix, as shown below:

$$M = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix} \tag{4}$$

In the principle of the TextRank algorithm, iterative calculation is actually a Markov process; that is, under the condition of knowing the current state, its future evolution does not depend on its past evolution. Therefore, the iteration result is affected only by the edge weight and transition probability in the word graph and not by the initial weight of the candidate keywords.

Initially, the original TextRank algorithm regards the weight of each side as 1, that is, the jump probability p_{ji} of any node j to the adjacent node i is $\frac{1}{|\mathrm{Out}(V_j)|}$, but this algorithm only focuses on the words in the text. The frequency of appearance simply sets the weight of the word node to spread equally to neighboring nodes. Therefore, the jump probability is calculated according to the node importance set in advance, so that the node weights are non-uniformly propagated, to improve the iterative calculation of the word node weights in TextRank. This paper introduces the semantic clustering results based on BERT word vectors.

3.2 Importance of semantic clustering

Word vectors are vector forms that represent words on a computer. Among the methods used to generate word vectors, Word2Vec proposed by Mikolov et al.[14] uses a shallow neural network to train the corpus to generate each word vector. In 2019, Devlin et al.[18] proposed BERT, which is a bidirectional language model that uses transformers as a feature extractor. Compared with previous vector models, BERT solves the polysemy problem and expresses the semantics of words more accurately. The model structure of BERT is shown in Fig. 2.

The input of BERT is a sentence pair. For a given word, its input representation is formed by summing three parts of Embedding: Token Embedding represents a word vector, segment Embedding represents a text vector, and position Embedding represents a position vector. In this way, the grammar and context information of the word are well integrated.

The middle layer is composed of a multi-layer

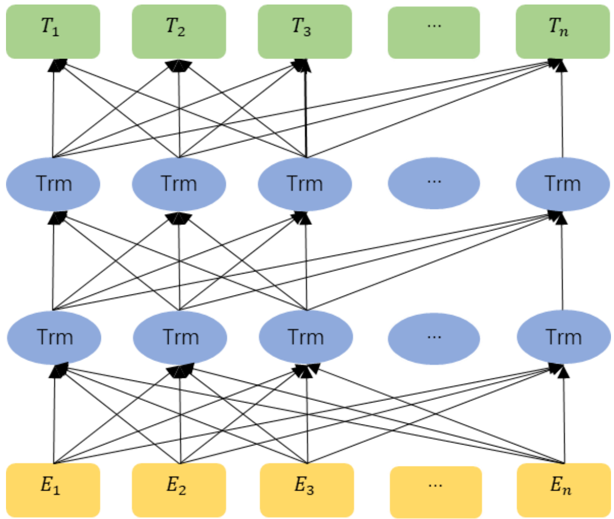

Fig. 2 BERT model structure.

bidirectional transformer (Trm). A model with strong generalization ability can be obtained by pretraining with a large amount of data as shown in Fig. 2. After training the above model, the vector representation T_i of each word (token) can be obtained.

In summary, this article uses word vectors generated by BERT to represent the semantics of words. Observation of abundant news and its keywords shows that news usually describes a current event. Thus, the semantic difference between multiple keywords used to express its core content is large, that is, the keywords of news are always various themes. The LDA theme model also uses this feature to extract keywords. Therefore, this study uses word vector clustering to represent semantic clustering according to the semantic differences between different keywords and assigns high importance to the word nodes at the center of the cluster, so that it plays a role in the iteration of TextRank higher jump probability.

k-means clustering based on Euclidean distance, which is an unsupervised learning method that divides vectors with high similarity into a cluster, is used in this study. Given document, its corresponding word set $W = [w_1, w_2, \ldots, w_n]$ is obtained, and the BERT word vector model is used to generate the corresponding word vector set $\mathrm{WT} = [T_1, T_2, \ldots, T_n]$. The result of k-means clustering of this word vector set is $C = [c_1, c_2, \ldots, c_k]$, and c_k represents the set of word vectors in cluster k and each cluster can find a centroid. For any word a from c_k, the clustering importance of word a can be calculated using Formula (5):

$$w_{\mathrm{vec}}(a) = \begin{cases} t, & a \text{ is centeroid;} \\ 1, & a \text{ is not centeroid} \end{cases} \tag{5}$$

where t is the preset weight, which is set to 5 in this study. Formula (5) shows that the word node selected as the cluster center of a cluster will be given the highest clustering importance, and the importance of other nodes is considered to be 1.

3.3 Construct transition probability matrix and extract keywords

w_{vec} has been calculated above, and let w_{fre} and w_{loc} denote the frequency importance of words and the position importance of words, respectively.

For any word a, the calculation equation of w_{fre} is as follows:

$$w_{\text{fre}}(a) = \frac{F_a}{n} \tag{6}$$

where F_a is the frequency of word a in the text.

The calculation equation of w_{loc} is as follows:

$$w_{\text{loc}}(a) = \begin{cases} t, & a \text{ in title;} \\ 1, & a \text{ not in title} \end{cases} \tag{7}$$

When a appears in the title, its positional importance is given to weight.

The influence of three important degrees of words is calculated as follows:

$$w(i,j) = \frac{w_j}{\sum_{V_k \in \text{Out}(V_i)} w_k} \tag{8}$$

According to Eq. (9), three influences are weighted and summed in a certain proportion ($\alpha = 0.25$, $\beta = 0.5$, $\gamma = 0.25$) and a new transition probability is obtained:

$$p_{ij} = \alpha \times w_{\text{loc}}(i,j) + \beta \times w_{\text{vec}}(i,j) + \gamma \times w_{\text{fre}}(i,j) \tag{9}$$

This calculated transition probability is used to construct a new transition probability M, which is iteratively calculated using Eq. (3) until the current iteration result B_i converges, after which the iteration is stopped. The words are sorted in descending order according to the final weight of the word nodes, and the top K words are selected as keywords to complete the extraction of keywords.

4 Experiment and Analysis

4.1 Experimental material and method

The test text library of this experiment was extracted from the news articles of *Nanfang Daily* from November to December in 2019. When each news released, the news editor marked the original manuscript with keywords, which were then used as the standard keywords for each news article in this experiment. Then, the accuracy and recall of the keyword extraction algorithm were calculated according to these keywords.

In this study, the crawler used to obtain news from the website was developed based on the Scrapy framework. It is aimed at the secondary development of the hierarchical structure of each page in the target news website and the front-end code structure of the news page to accurately obtain the news text and keywords from the news page. After about 2500 news articles were obtained from the website, 500 news articles with more than 500 words and 5 corresponding keywords were selected as the test set of this experiment.

The keyword extraction algorithms mentioned in this paper were all implemented by Python. When dealing with news text, we firstly used the open source word segmentation tool *jieba* to segment the sentence and define the characteristic of the word and then filter some useless words to obtain candidate words. At the same time, the word vector was generated by the pretrained BERT model in Chinese corpus provided by Google. The word vector corresponding to each word in the sentence was obtained by inputting sentences, and it can accurately represent the semantic meaning of words in the context.

In the present study, common precision and recall were used to evaluate the effectiveness of this algorithm. The calculation method is as follows.

In Table 1, TP and FP indicate the number of keywords that belong to and not belong to the extracted words, respectively, and FN and TN indicate the number of keywords that belong to and not belong to the words that have not been extracted, respectively. Then, the calculation equations of precision (P) and recall (R) are as follows:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

Equations (10) and (11) reflect the extraction effect of the algorithm from two different aspects. Equation (10) indicates whether the algorithm is accurate, and Eq. (11) indicates whether the effect of the algorithm extraction is comprehensive. Therefore, in order to evaluate the extraction effect of the algorithm more evenly, this experiment uses the F1 value:

Table 1 Results of keyword extraction.

Type of keywords	Number of keywords	Number of non-keywords
Extracted	TP	FP
Not extracted	FN	TN

$$\text{F1} = \frac{2 \times P \times R}{P + R} \tag{12}$$

4.2 Experimental result and analysis

In the experiment, the semantic clustering TextRank is referred to as SCTR, and the original TextRank is referred to as TR. The extraction effects of SCTR, TR, and TF-IDF were compared.

Results of repeated experiments show that when the co-occurrence window of length $l = 5$ and the damping coefficient $d = 0.85$, the extraction effect of the two algorithms reaches a relatively good level. Therefore, the co-occurrence window of length $l = 5$ and the damping coefficient $d = 0.85$. Moreover, when the clustering importance of the clustering center is 30, the overall importance of the algorithm spreads well. Thus, in this case, the extraction effects of the two algorithms are compared.

(1) Before the iterative calculation of TextRank, the algorithm firstly calculates the importance of clustering, and the selection of the number of clustering centers affects the final distribution of word weights. Thus, the first experiments are conducted based on different numbers of cluster centers, as shown in Fig. 3.

Figure 3 shows that when the number of cluster centers is 3, the precision of keyword extraction is the highest. Thus, the following experiments and analysis are performed with the number of cluster centers being 3.

(2) When the number of keyword extraction is different, the precision, recall, and F1 value of SCTR, TR, and TF-IDF are compared. The experimental results are shown in Figs. 4–6.

Figures 4–6 show that when the number of keyword extraction is small, the precision, recall, and F1 value of three keyword extration algorithms are basically close to each other. This result can be ascribed to the fact that the three algorithms deem the core words that

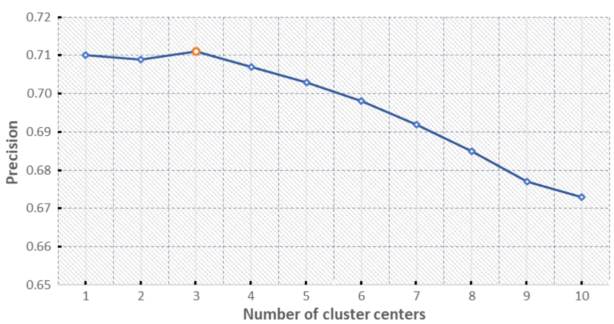

Fig. 3 Comparison of precision of keyword extraction under different numbers of cluster centers.

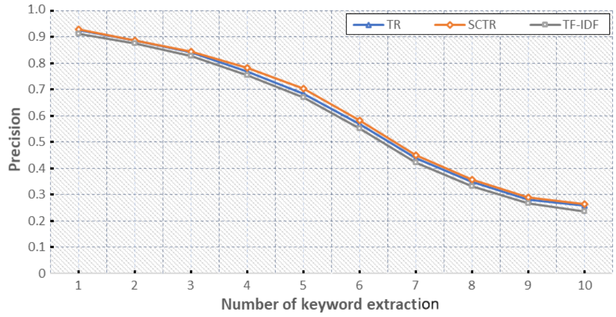

Fig. 4 Comparison of precision under different numbers of keyword extraction.

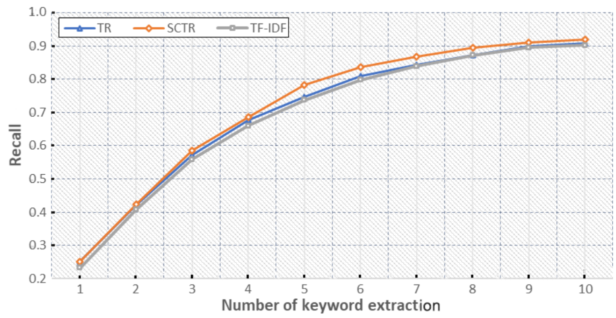

Fig. 5 Comparison of recall under different numbers of keyword extraction.

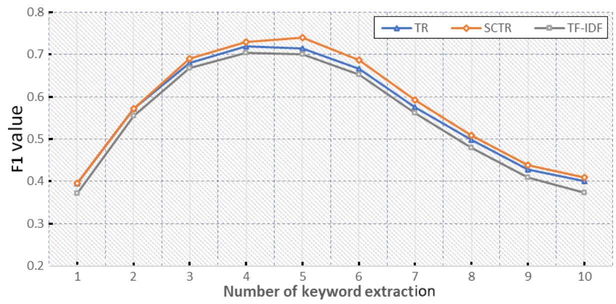

Fig. 6 Comparison of F1 value under different numbers of keyword extraction.

appear frequently in the article to be similar. When the number of keyword extraction reaches about 5, the SCTR algorithm designed in this paper is obviously better than the TF-IDF algorithm in the above three indicators and has a greater improvement compared with the traditional TR algorithm. This finding is because the semantic clustering of words gives a large weight to words with low frequency and meets the central theme of the article, thus improving the extraction effect.

The F1 value of the three algorithms basically reaches the maximum when the number of keyword extraction is 5 mainly because the number of keywords in the news text is 5. Thus, when the number of keyword extraction by the algorithm is 5, the precision and recall can reach relatively large values. Thus, F1 also reaches the peak value.

The precision-recall histogram in Fig. 7 shows that the precision and recall have a negative correlation. In addition, it can be seen from Fig. 7 that with the improvement of precision, the recall of SCTR algorithm is always greater than that of TR and TF-IDF algorithms.

The above experimental results show that the precision, recall, and F1 value of SCTR are always higher than those of TR and TF-IDF as the number of keyword extraction increases from 1 to 10.

5 Conclusion and Future Work

The traditional TextRank algorithm selects candidate words with high frequency in the text as keywords to optimize the extraction. The word vector is introduced to represent the semantics of the word, so that the weight of the word is calculated in many ways, thus improving the extraction of keywords. Experimental results show that the proposed keyword extraction algorithm based on semantic clustering is superior to the traditional TextRank and TF-IDF algorithms in terms of precision, recall, and F1 value.

TextRank is a classic algorithm in keyword extraction. In this paper, we introduce the semantics of the word vector into the algorithm, which can obviously improve the keyword extraction of the algorithm.

Experimental data in this study can still be improved. In the next work, we can introduce text and keywords with a large amount of data and a wide field for experiment and analysis. The TF-IDF algorithm will be studied. This algorithm focuses on the use of word frequency and the characteristics of the whole text database into the algorithm. The effect of keyword extraction can be improved by adjusting the importance of words relative to the text database.

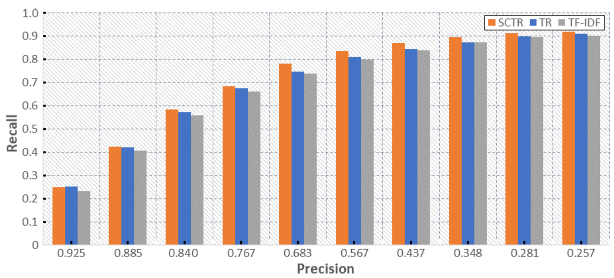

Fig. 7 Precision-recall histogram of three algorithms.

Acknowledgment

This work was supported by the National Key R&D Program of China (No. 2018YFE0205502) and the National Natural Science Foundation of China (No. 61672108).

References

[1] A. Z. Guo and T. Yang, Research and improvement of feature words weight based on TFIDF algorithm, presented at 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conf., Chongqing, China, 2016, pp. 415–419.

[2] J. Z. Li, Q. N. Fan, and K. Zhang, Keyword extraction based on tf/idf for Chinese news document, *Wuhan Univ. J. Nat. Sci.*, vol. 12, no. 5, pp. 917–921, 2007.

[3] H. X. Wei, G. L. Gao, and X. D. Su, LDA-based word image representation for keyword spotting on historical Mongolian documents, in *Proc. 23rd Int. Conf. Neural Information Processing*, Kyoto, Japan, 2016, pp. 432–441.

[4] R. Mihalcea and P. Tarau, TextRank: Bringing order into text, in *Proc. 2004 Conf. Empirical Methods in Natural Language Process.*, Stroudsburg, PA, USA, 2004, pp. 404–411.

[5] H. Chen, C. Yin, R. Li, W. Rong, Z. Xiong, and B. David, Enhanced learning resource recommendation based on online learning style model, *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 348–356, 2020.

[6] X. Y. Zhang, Y. B. Wang, and L. Wu, Research on cross language text keyword extraction based on information entropy and TextRank, presented at 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conf. (ITNEC), Chengdu, China, 2019, pp. 16–19.

[7] Y. Luo, S. L. Zhao, X. C. Li, Y. H. Han, and Y. F. Ding, Text keyword extraction method based on word frequency statistics, (in Chinese), *J . Comput. Appl.*, vol. 36, no. 3, pp. 718–725, 2016.

[8] H. T. Geng, Q. S. Cai, K. Yu, and P. Zhao, A kind of automatic text keyphrase extraction method based on word co-occurrence, (in Chinese), *J. Nanjing Univ.* (*Nat. Sci.*), vol. 42, no. 2, pp. 156–162, 2006.

[9] Y. J. Gu and T. Xia, Study on keyword extraction with LDA and TextRank combination, (in Chinese), *Data Anal. Knowl. Discov.*, vol. 30, no. 7, pp. 41–47, 2014.

[10] F. Jiang, G. H. Li, and X. Yun, Semantic-based keyword extraction method for document, *International Journal of u-and e-Service, Science and Technology*, vol. 8, no. 5, pp. 37–46, 2015.

[11] H. Li, C. L. Tang, X. Yang, and W. T. Shen, TextRank keyword extraction based on multi feature fusion, (in Chinese), *J. Intell.*, vol. 36, no. 8, pp. 183–187, 2017.

[12] L. Tian and S. Wang, Improved bag-of-words model for person re-identification, *Tsinghua Science and Technology*, vol. 23, no. 2, pp. 145–156, 2018.

[13] Z. Shen, B. Yong, G. Zhang, R. Zhou, and Q. Zhou, A deep learning method for Chinese singer identification, *Tsinghua Science and Technology*, vol. 24, no. 4, pp. 371–378, 2019.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space,

https://arxiv.org/abs/1301.3781, 2013.

[15] Y. P. Li, C. Jin, and J. C. Ji, A keyword extraction algorithm based on Word2vec, (in Chinese), *E-Sci. Technol. Appl.*, vol. 6, no. 4, pp. 54–59, 2015.

[16] J. Z. Zhou and X. H. Cui, Keyword extraction method based on word vector and TextRank, (in Chinese), *Appl. Res. Comput.*, vol. 36, no. 4, pp. 1051–1054, 2019.

[17] Y. J. Wen, H. Yuan, and P. Z. Zhang, Research on keyword extraction based on Word2Vec weighted TextRank, presented at 2016 2nd IEEE Int. Conf. Computer and Communications (ICCC), Chengdu, China, 2016, pp. 2109–2113.

[18] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, https://arxiv.org/abs/1810.04805, 2019.

[19] D. Miller, Leveraging BERT for extractive text summarization on lectures, https://arxiv.org/abs/1906.04165, 2019.

[20] L. Zhang, C. B. Xu, Y. H. Gao, Y. Han, X. J. Du, and Z. H. Tian, Improved Dota2 lineup recommendation model based on a bidirectional LSTM, *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 712–720, 2020.

Ao Xiong received the PhD degree in computer science from Beijing University of Posts and Telecommunications (BUPT) in 2013. He is currently an associate professor at State Key Laboratory of Networking and Switching Technology, BUPT. He has authored about 30 SCI/EI index papers and received 14 national and provincial scientific and technical awards. His research interests include communication network and communication software, internet routing and applications, and network management.

Derong Liu received the BEng degree from Central South University in 2018. He is currently pursuing the master degree at State Key Laboratory of Networking and Switching Technology, BUPT. His research interest is keyword extraction in natural language processing.

Hongkang Tian received the BEng degree from Xi'an University of Posts and Telecommunications in 2019. He is currently pursuing the master degree at State Key Laboratory of Networking and Switching Technology, BUPT. His research interests include keyword extraction in natural language processing and SLA management based on blockchain.

Zhengyuan Liu received the BEng degree from Changchun University of Science and Technology in 2020. He is currently pursuing the master degree at State Key Laboratory of Networking and Switching Technology, BUPT. His research interests include autonomic network management for 5G/6G networks and smart grid communications.

Peng Yu received the BEng and PhD degrees from Beijing University of Posts and Telecommunications in 2008 and 2013, respectively. He is an associate professor at State Key Laboratory of Networking and Switching Technology, BUPT. His research interests are autonomic network management for 5G/6G networks and smart grid communications.

Michel Kadoch is a full professor at École de Technologie Supérieure, Université du Québec. He obtained the PhD degree from Concordia University in 1992. He is the author of the book *Protocoles et réseaux locaux*: *Accs̀ à Internet* (PUQ Press 2012) and co-author of many other books. He has established the research lab "LAGRIT: Laboratoire de gestion informatique et de telecommunication", which performs research on data communication networking. His current work includes 5G, SON, NFV, LTE, HetNet, WMN, resource allocation, and performance analysis. He served as the TPC member and editor for several conferences and journals. He has also been the director of the master of engineering program since 2001. He is supervising post doctorates, PhD, and master students as well as undergraduate students working at the lab. He has been a senior member of IEEE since 2004.