Research Article

Jinye Li*

A comparative study of keyword extraction algorithms for English texts

https://doi.org/10.1515/jisys-2021-0040
received March 14, 2021; accepted June 15, 2021

Abstract: This study mainly analyzed the keyword extraction of English text. First, two commonly used algorithms, the term frequency–inverse document frequency (TF–IDF) algorithm and the keyphrase extraction algorithm (KEA), were introduced. Then, an improved TF–IDF algorithm was designed, which improved the calculation of word frequency, and it was combined with the position weight to improve the performance of keyword extraction. Finally, 100 English literature was selected from the British Academic Written English Corpus for the analysis experiment. The results showed that the improved TF–IDF algorithm had the shortest running time and took only 4.93 s in processing 100 texts; the precision of the algorithms decreased with the increase of the number of extracted keywords. The comparison between the two algorithms demonstrated that the improved TF–IDF algorithm had the best performance, with a precision rate of 71.2%, a recall rate of 52.98%, and an F_1 score of 60.75%, when five keywords were extracted from each article. The experimental results show that the improved TF–IDF algorithm is effective in extracting English text keywords, which can be further promoted and applied in practice.

Keywords: English text, keyword extraction, TF–IDF algorithm, KEA

1 Introduction

With the development of society, there are more and more ways to express information, among which natural language text is the most important one [1] and one of the largest information sources [2]. With the rapid growth of the amount of text information, how to process and retrieve these massive texts has become a more and more important problem [3]. Information retrieval [4], text classification [5], emotional analysis [6], and topic identification [7] have been widely concerned by researchers. In the process of information retrieval, users can find the corresponding web pages by inputting keywords. However, if the keywords input by users are not accurate or the keywords do not appear on the corresponding page, the retrieval effect of information will be greatly affected. Therefore, keywords are very important in text processing [8]. Beliga et al. [9] introduced keyword extraction methods related to supervised and unsupervised methods, analyzed and compared various graph-based methods, and encouraged the development of new graph-based methods for keyword extraction. Biswas et al. [10] studied the extraction of Twitter keywords, proposed an unsupervised graph-based method, extracted keywords using the collective node weight, carried out experiments on five data sets, and found that this method was better than other methods. Hu et al. [11] designed a patent keyword extraction algorithm based on a distributed skip-gram model and conducted experiments on standard data sets and self-made data sets. They found that the method was useful in extracting keywords from patent texts. Zhang et al. [12] studied the TextRank algorithm and conducted experiments on Hulth 2003 and Krapivin 2009 datasets. The results showed that the TextRank

* **Corresponding author: Jinye Li,** Department of Applied Foreign Languages, Dongguan Polytechnic, No. 3, Daxue Road, Dongguan, Guangdong 523808, China, e-mail: jinba93@163.com

algorithm had an excellent performance, and the results were independent of the text length. Machine learning is a science of artificial intelligence, and its theory and method have been extensively applied in solving complex problems in the engineering application and science field; moreover, it also has good applications in natural language processing [13]. With the explosive growth of information, how to find out the needed information rapidly has become more and more difficult; thus, research on keyword extraction becomes increasingly important. This study compared three keyword extraction algorithms, the term frequency–inverse document frequency (TF–IDF) algorithm, keyphrase extraction algorithm (KEA), and the improved TF–IDF algorithm, and made an experimental analysis in the corpus to verify the effectiveness of the improved algorithm. The improved TF–IDF algorithm is beneficial to improving the extraction effect of keywords, realizing effective information mining, and helping works such as text retrieval and classification, providing a more efficient method for information processing. In practical application, the improved TF–IDF algorithm can be utilized to extract keywords from English texts to provide services for the sorting and retrieval of English questions and English literature and enhance the efficiency and level of text processing.

2 Common keyword extraction algorithms

The TF–IDF algorithm is a classic keyword extraction method [14], which mainly evaluates the importance of a word or a phrase to the text. The importance is related to two factors, TF and IDF. TF refers to the frequency of a word appearing in the document; the higher the frequency is, the more important the word is. The meaning of IDF is as follows. The document where a word is located belongs to a corpus; in this corpus, if the frequency of a word is high, it means that the word is not highly representative and of low importance; if the frequency of a word is low, it means that the word is highly representative and important. Suppose that there is a text d whose keyword needs to be extracted. After preprocessing, such as word segmentation, the content is regarded as the set of feature words, i.e., d_i, expressed as $d_i = (t_1, t_2, \ldots, t_j, \ldots, t_D)$. For the importance of every feature word (t_j) in text d_i, it is written as weight w_i. w_j is calculated according to the TF and IDF of feature words:

$$\mathrm{TF-IDF}(d_i, t_j) = \mathrm{TF}(d_i, t_j) \times \mathrm{IDF}(t_j) = \mathrm{TF}(d_i, t_j) \times \lg\left(\frac{M}{\mathrm{DF}(t_j)}\right), \tag{1}$$

where $\mathrm{TF}(d_i, t_j)$ stands for the frequency of appearance of t_j in d_i, $\mathrm{DF}(t_j)$ stands for the number of texts in which t_j appears in the text set, and M stands for the number of texts in the text set.

After calculating the weight, t_j is sorted according to the size of w_j, i.e., $\mathrm{Sort}(d_i)$. The top k feature words are taken as the final keywords.

KEA is targeted at English texts [15], and it predicts which keywords are good through machine learning. Its principle is to train a model with documents with marked keywords through a naive Bayes classifier [16] and extract keywords from new documents with the new model. It mainly uses two features. One is the TF–IDF value, and the other is the first appearance position of words. For a feature word, if it appears in the title, abstract, or introduction of a text, it is more likely to be a keyword. The calculation of the first appearance position of words is the ratio of the number of words in front of the feature word when it first appears in the text to the total number of words in the text. When training a Bayesian model, it is assumed that the text to be classified is expressed as follows:

$$x = \{a_1, a_2, \ldots, a_m\}. \tag{2}$$

The category collection is expressed as follows:

$$C = \{y_1, y_2, \ldots, y_n\}. \tag{3}$$

The probability of every category, $P(y_i)$, is calculated. The probability of every category to a_j, $P(y_i|a_j)$, is calculated. After the model training, in the application stage, the probability of x to every category y_i, $P(x|y_i)$, is calculated. The category that x belongs to is y_i with the highest probability.

3 Improved TF–IDF algorithm

To improve the performance of the TF–IDF algorithm, this study improved it from two aspects. First, in the calculation of TF, the original TF–IDF algorithm treats all keywords equally and gives them higher weight when they appear more frequently. However, considering the actual situation, in the actual English text, there may be some situations. For example, a word appears more frequently in the text, but it is given a high weight though it belongs to the daily use of words and has low discrimination for the text; a word is a rare word that seldom appears in the text, but it is given a low weight though it has a great contribution to the distinction of the text. Given these situations, this study improved the calculation of TF, i.e., adjusting the weight according to the relative appearance times of keywords, and the corresponding formula is expressed as follows:

$$\mathrm{TF} = e^{N_{d,t}-\bar{N}_t}, \tag{4}$$

where $N_{d,t}$ stands for the appearance times of keyword t in text d and $\bar{N}_t$ stands for the average appearance times of t in a text set. If the appearance times of a word in the text is less than its average times, then the TF value is smaller than 1, and the weight of the keyword will decrease; otherwise, it will increase. In this way, the influence of common words and rare words on weight can be reduced.

For an English text, the possible positions of candidate keywords are title, introduction, text, and so on. This study mainly divides the positions of keywords into three types. The first type is the title. Title is the summary of the text content and theme. If a word appears in the title, it is most likely to be the final keyword with the highest importance. The second type is the chapter title. Chapter titles can summarize the content of a chapter, and the candidate keywords appearing in the title of a chapter are also very important. The third type is the main body. There is much content in the main body; therefore, the possibility of keywords appearing in the main body is relatively large, but its importance is slightly less than that of the title and chapter title.

Based on calculating the TF–IDF value and the position weight p_i, the feature value of candidate keywords is the product of the position weight and TF–IDF value. The specific calculation method is as follows. It is assumed that there is a text containing 10,000 words; the keyword has been labeled; the title, chapter title, and content of the text are named text 1, text 2, and text 3, respectively; the candidate keywords appearing in these content are named place 1, place 2, and place 3, respectively; and the labeled keyword is named text 4. The formula for calculating the frequency of keywords in different positions is expressed as follows:

$$p = \frac{m}{n}, \tag{5}$$

where m refers to the number of keywords in this part of the content and n refers to the total number of words in this part of the content. If the statistical text information is shown in Table 1, then the probability of the keyword appearing in the title is:

Table 1: Statistical results of text information

Name	Frequency of words	Name	Frequency of words
Place 1	15	Text 1	76
Place 2	29	Text 2	498
Place 3	102	Text 3	9,426

$$P1 = \text{place } 1/\text{text } 1 = 15/76 = 0.1974 \tag{6}$$

$$P2 = \text{place } 2/\text{text } 2 = 29/498 = 0.0582 \tag{7}$$

$$P3 = \text{place } 3/\text{text } 3 = 102/9426 = 0.0108 \tag{8}$$

According to the results, the position weight of the keyword in the title, chapter title, and main body can be set as 19.7, 5.8, and 1.1%, respectively.

4 Experimental analysis

In this study, 100 English literature was randomly selected from the British Academic Written English (BAWE) Corpus [17] as experimental texts. A total of 672 keywords were given by the 100 literature. The content of the literature is presented in Table 2.

Table 2: Experimental data

Subject	Number of documents
Biology	12
Military	6
Power engineering	10
Electrical engineering	9
Chemical engineering	7
Mechanical engineering	11
Water conservancy project	8
Mineral engineering	10
Natural science	15
Architecture science	12

The texts were analyzed by StopAnalyzer that is specially used for analyzing English texts. The system can filter out space in the text, convert capital and lowercase letters, and remove stop words. After preprocessing, the remaining words in the text are mainly nouns and verbs, which is more conducive to keyword extraction.

English literature generally provides three to six keywords, no more than 15 at most. To better confirm the performance of the algorithms, the number of keywords to be extracted was set as 5–30.

The results of keyword extraction are presented in Table 3, and the algorithms were evaluated by three indicators, as shown below.

Table 3: Results of keyword extraction

	Manually labeled as the keyword	Not manually labeled as the keyword
Extracted as the keyword by the algorithm	A	B
Not extracted as the keyword by the algorithm	C	D

(1) Precision:

$$P = \frac{A}{A + B}. \tag{9}$$

(2) Recall rate:

$$R = \frac{A}{A + C}. \tag{10}$$

(3) F_1 score:

$$F_1 = \frac{2PR}{P + R}. \tag{11}$$

The performance of three algorithms, TF–IDF, KEA, and improved TF–IDF algorithms, in extracting keywords was compared. First, the running time of the algorithms under different numbers of texts is presented in Table 4.

Table 4: Comparison of the running time between algorithms

Number of texts	TF–IDF algorithm (s)	KEA algorithm (s)	Improved TF–IDF algorithm (s)
10	1.12	1.36	0.78
20	1.56	1.89	1.21
30	2.08	2.21	1.79
40	2.51	2.78	2.33
50	3.01	3.21	2.64
60	3.46	3.81	2.89
70	3.98	4.23	3.46
80	4.31	4.76	3.89
90	4.86	5.21	4.21
100	5.27	5.73	4.93

It was seen from Table 4 that the running time of the algorithms increased gradually with the increase of the number of texts. The running time of the Kea algorithm was the longest, followed by the TF–IDF algorithm and the TF–IDF algorithm. When the number of texts was ten, the running time of the improved TF–IDF algorithm was 30.36% shorter than the TF–IDF algorithm and 42.65% shorter than the KEA algorithm; when the number of texts was 100, the running time of the improved TF–IDF algorithm was 6.45% shorter than the TF–IDF algorithm and 13.96% shorter than the KEA algorithm. The reason for the aforementioned result was because the KEA algorithm needed training.

The extraction results of the algorithms when the number of keywords extracted was different are presented in Table 5.

Table 5: Results of keyword extraction under different algorithms

	The number of keywords extracted from each text	Total number of keywords extracted	Correct number of keywords
TF–IDF algorithm	5	500	198
	10	1,000	241
	15	1,500	307
	20	2,000	322
	25	2,500	356
	30	3,000	409
KEA algorithm	5	500	219
	10	1,000	263
	15	1,500	321
	20	2,000	398
	25	2,500	411
	30	3,000	489
Improved TF–IDF algorithm	5	500	243
	10	1,000	289
	15	1,500	367
	20	2,000	421
	25	2,500	478
	30	3,000	563

According to the results of keyword extraction, the P value, R value, and F_1 value of the algorithms were calculated, and the results are shown in Table 6.

It was seen from Table 5 that the P value of the algorithms decreased with the increase of keywords extracted from each text. The decrease of the P value might be because the number of keywords marked by originally was far less than the number of keywords extracted by the algorithms. With the increase of the

Table 6: Performance comparison between algorithms

	The number of keywords extracted from each text	Precision (P) (%)	Recall rate (R) (%)	F_1 score (F_1) (%)
TF–IDF algorithm	5	57.80	43.01	49.32
	10	30.80	45.83	36.84
	15	21.40	47.77	29.56
	20	16.75	49.85	25.07
	25	13.88	51.64	21.88
	30	12.27	54.76	20.04
KEA algorithm	5	63.80	47.47	54.44
	10	31.20	46.43	37.32
	15	22.20	49.55	30.66
	20	17.85	53.13	26.72
	25	15.84	58.93	24.97
	30	14.23	63.54	23.26
Improved TF–IDF algorithm	5	71.20	52.98	60.75
	10	37.20	55.36	44.50
	15	25.60	57.14	35.36
	20	22.80	67.86	34.13
	25	18.88	70.24	29.76
	30	18.40	82.14	30.07

number of keywords extracted, the R value of the algorithms increased, which indicated that the more keywords were extracted, the more marked keywords were included. Due to the rapid decline of the P value, the F_1 value also showed a downward trend. To compare the performance of different algorithms, five keywords extracted from each text were taken as an example to compare the P value, R value, and F_1 value. The results are shown in Figure 1.

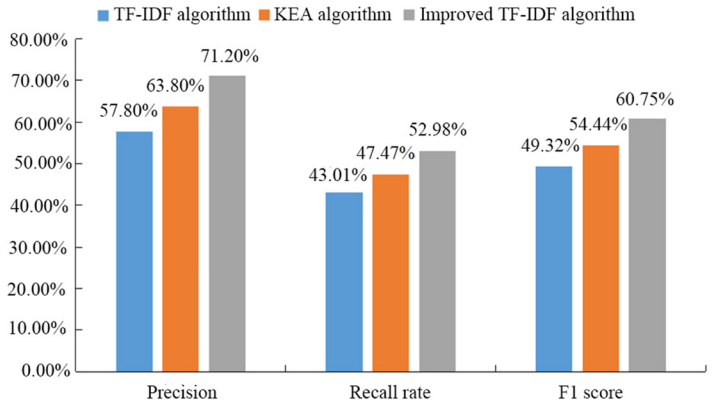

Figure 1: Performance comparison between algorithms when extracting five keywords from each text.

Figure 1 shows that the performance of the improved TF–IDF algorithm was significantly better; its accuracy was 71.2 and 13.4% higher than the TF–IDF algorithm and 7.4% higher than the KEA algorithm; its recall rate was 52.98 and 9.97% higher than the TF–IDF algorithm and 5.51% higher than the KEA algorithm; its F_1 score was 60.75 and 11.43% higher than the TF–IDF algorithm and 6.31% higher than the KEA algorithm. Therefore, the improved TF–IDF algorithm had good reliability in extracting keywords from English texts.

5 Discussion

In text processing, machine learning methods have been widely used. Onan et al. [18] designed a method that integrated Bayesian logistic regression, naive Bayes, linear discriminant analysis, logistic regression, and support vector machine. They studied sentiment classification of texts and found through experiments that the method showed the highest classification accuracy, 98.86%. Onan and Tocoglu [19] provided an effective sarcasm recognition framework for social media data by pursuing the paradigm of neural language models and deep neural networks, evaluated it on the corpus, and obtained a classification accuracy of 95.3%. Keywords play a very important role in many fields of text processing [20]. In document management [21], the efficiency of document management can be improved by collecting keywords of documents in different fields and establishing corresponding indexes; in text classification [22], the relevance of keywords to texts can be calculated to divide texts with similar semantics into one category; in automatic abstracting [23], the weight of each sentence can be calculated after keyword extraction to determine whether the sentence can become a part of the abstract to realize the automatic abstracting.

This article mainly compared three keyword extraction algorithms, TF–IDF, KEA, and improved TF–IDF algorithms. This study selected 5–30 keywords from each text, and there were 100 texts. The running time of the improved TF–IDF algorithm was the shortest, 4.93 s, which was 6.45% shorter than the TF–IDF algorithm and 13.96% shorter than the KEA algorithm. The precision, recall rate, and F1 score of the algorithms were compared under different numbers of keywords extracted. It was found that the P value and F1 score of the three algorithms decreased with the increase of the number of keywords extracted, while the R value increased, which was because the number of keywords labeled in the text originally was less than the number of keywords extracted by the algorithms. The improved TF–IDF algorithm had significantly improved performance after improving the calculation method of TF and introducing the concept of the position weight; its P value, R value, and F_1 score were significantly higher than those of TF–IDF and KEA algorithms. It showed that the improved TF–IDF algorithm had strong applicability in extracting keywords from English texts.

It was found from the experimental results that the TF–IDF algorithm only used word frequency to measure the importance of a word and did not take into account the location information of the word. For KEA, if the structure of a text was poor, the contribution of the first occurrence of the word to the keyword extraction is small; then, the results of keyword extraction will also be affected. In addition, KEA has a strong dependence on the Bayesian algorithm. Bayes assumes that the features are independent of each other. If this hypothesis fails, the trained classifier will have obvious defects, and the effect of keyword extraction will become worse. Table 6 shows that the performance of the TF–IFD algorithm and KEA was general, but this study improved the calculation method of word frequency and combined the position weight to greatly improve the keyword extraction performance of the TF–IDF algorithm. The effectiveness of the improved algorithm was verified. In practical application, the improved TF–IDF algorithm can be used for keyword extraction of English literature to help readers understand the content of the literature better, and it can also be used for keyword extraction of English education resources to improve the intelligent level of teaching and meet the needs of learners.

6 Conclusion

This study compared the performance of three algorithms in extracting keywords from English texts. The results showed that the improved TF–IDF algorithm had the best performance and the shortest running time. The precision, recall rate, and F_1 score of the improved TF–IDF algorithm were 71.2, 52.98, and 60.75%, respectively, significantly better than TF–IDF and KEA algorithms. The method can be further promoted and applied in practice. This study also has some defects. In future work, the performance of keyword extraction algorithms will be further improved, and experiments will be carried out in more corpora to perfect the experimental results.

Conflict of interest: The author states no conflict of interest.

References

[1] Perovek M, Kranjc J, Erjavec T, Cestnik B, Lavrac N. TextFlows: a visual programming platform for text mining and natural language processing. Sci Comput Program. 2016;121:128–52.
[2] Onan A. Hybrid supervised clustering based ensemble scheme for text classification. Kybernetes. 2017;46(2):330–48.
[3] Onan A. Sentiment analysis on twitter based on ensemble of psychological and linguistic feature sets. Balkan J Electr Comput Eng. 2018;6:1–9.
[4] Berger A, Lafferty J. Information retrieval as statistical translation. ACM SIGIR Forum. 2017;51(2):219–26.
[5] Onan A. An ensemble scheme based on language function analysis and feature engineering for text genre classification. J Inf Sci Prin Pract. 2018;44(1):28–47.
[6] Onan A, Korukoglu S. A feature selection model based on genetic rank aggregation for text sentiment classification. J Inf Sci. 2017;43(1):25–38.
[7] Onan A, Toolu MA. Weighted word embeddings and clustering-based identification of question topics in MOOC discussion forum posts. Comput Appl Eng Educ. 2020.
[8] Firoozeh N, Nazarenko A, Alizon F, Daille B. Keyword extraction: issues and methods. Nat Lang Eng. 2019;26(3):1–33.
[9] Beliga S, Meštrović A, Martinčić-Ipšić S. An overview of graph-based keyword extraction methods and approaches. J Inform Organ Sci. 2015;39(1):1–20.
[10] Biswas SK, Bordoloi M, Shreya J. A graph based keyword extraction model using collective node weight. Expert Syst Appl. 2017;97:51–9.
[11] Hu J, Li SB, Yao Y, Yu LY, Yang GC, Hu JJ. Patent keyword extraction algorithm based on distributed representation for patent classification. Entropy. 2018;20(2):104.
[12] Zhang MX, Li XM, Yue SB, Yang LQ. An empirical study of TextRank for keyword extraction. IEEE Access. 2020;8:178849–58.
[13] Onan A. Mining opinions from instructor evaluation reviews: a deep learning approach. Comput Appl Eng Educ. 2020;28(1):117–38.
[14] Chin K, Zhang Z, Long J, Zhang H. Turning from TF–IDF to TF-IGM for term weighting in text classification. Expert Syst Appl. 2016;66:45–260.
[15] Qiu Q, Xie Z, Wu L, Li WJ. Geoscience keyphrase extraction algorithm using enhanced word embedding. Expert Syst Appl. 2019;125:157–69.
[16] Onan A, Korukoglu S, Bulut H. LDA-based topic modelling in text sentiment classification: an empirical analysis. Int J Comput Linguist Appl. 2016;7(1):101–19.
[17] Farahani M. Metadiscourse in academic English texts: a corpus-driven probe into British academic written english corpus. Stud About Lang. 2019;34:56–73.
[18] Onan A, Korukoğlu S, Bulut H. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification – ScienceDirect. Expert Syst Appl. 2016;62:1–16.
[19] Onan A, Tocoglu MA. A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification. IEEE Access. 2016;4:1–23.
[20] Onan A, Korukoğlu S, Bulut H. Ensemble of keyword extraction methods and classifiers in text classification. Expert Syst Appl. 2016;57:232–47.
[21] Schaefer MB, Lima EDS. Evaluation and classification of documents: an analysis of its aplication in a digital archival document management system. Perspect Ciênc Inf. 2012;17(3):137–54.
[22] Otaibi JA, Hassaine A, Safi Z, Jaoua A. Inconsistency detection in islamic advisory opinions using multilevel text categorization. Adv Sci Lett. 2017;23(5):4591–5.
[23] Hernandez-Castaneda A, Garcia-Hernandez RA, Ledeneva Y, Millán-Hernández CE. Extractive automatic text summarization based on lexical-semantic keywords. IEEE Access. 2020;8:49896–907.