Mathematical Biosciences and Engineering

MBE, 21(4): 5411–5429.
DOI: 10.3934/mbe.2024238
Received: 08 October 2023
Revised: 30 December 2023
Accepted: 08 January 2024
Published: 08 March 2024

http://www.aimspress.com/journal/MBE

Research article

Co-occurrence word model for news media hotspot mining-text mining method design

Xinyun Zhang[1,*] and Tao Ding[2]

[1] School of Arts and Creative Technologies, The University of York, York, United Kingdom
[2] Department of Statistical Science, University College London, London, United Kingdom

* **Correspondence:** Email: chichi0707@163.com.

Abstract: Currently, with the rapid growth of online media, more people are obtaining information from it. However, traditional hotspot mining algorithms cannot achieve precise and fast control of hot topics. Aiming at the problem of poor accuracy and timeliness in current news media hotspot mining methods, this paper proposes a hotspot mining method based on the co-occurrence word model. First, a new co-occurrence word model based on word weight is proposed. Then, for key phrase extraction, a hotspot mining algorithm based on the co-occurrence word model and improved smooth inverse frequency rank (SIFRANK) is designed. Finally, the Spark computing framework is introduced to improve the computing efficiency. The experimental outcomes expresses that the new word discovery algorithm discovered 16871 and 17921 new words in the Weibo Short News and Weibo Short Text datasets respectively. The heat weight values of the keywords obtained by the improved SIFRANK reaches 0.9356, 0.9991, and 0.6117. In the Covid19 Tweets dataset, the accuracy is 0.6223, the recall is 0.7015, and the F1 value is 0.6605. In the President-elects Tweets dataset, the accuracy is 0.6418, the recall is 0.7162, and the F1 value is 0.6767. After applying the Spark computing framework, the running speed has significantly improved. The text mining news media hotspot mining method based on the co-occurrence word model proposed in this study has improved the accuracy and efficiency of mining hot topics, and has great practical significance.

Keywords: tot news topic; co-occurrence word model; text mining; theme word extraction; hot spot discovery

1. Introduction

Hots news topic mining (HNTM) refers to extracting hot topics from Internet news data according to certain rules, and analyzing and mining them to obtain the content that users are interested in to meet user needs [1,2]. When mining hot topics in news, it is necessary to consider the data information provided by different information sources, and comprehensively consider the relationship between user needs, media hot topics, and online public opinion. At present, research on HNTM at home and abroad mainly focuses on multi-granularity information extraction technology, text similarity calculation technology, and topic detection technology based on clustering [3,4]. There are complex relationships and thematic structures between texts, and co-occurrence word models (CWMs) can fully consider these text features. At the same time, the model can also accurately mine the correlation between news reports and events [5]. However, traditional models did not take into account the mutual influence between keywords in calculating their similarity, and their computational efficiency was slow. Aiming at the above problems, this paper proposes a HNTM method based on the CWM. This method comprehensively considers the interaction between keywords and the relationship between texts, which can improve the accuracy and effectiveness of HNTM.

The co-occurrence word model proposed in this study can overcome the problems of insufficient carrying information, sparse data and high feature dimension in text; its extraction effect of short text hot words is no worse than that of traditional SIFRANK algorithm; and the extraction speed is 5 to 8 times faster. Through the design of heat weight model, an improved algorithm Hot topic mined by Co-Occurrence Word Model and HSIFRANK (HCH) is proposed to improve the traditional co-occurrence word model. A Sliding window algorithm and HSIFRANK are used to effectively capture heat transition topics and dig hot sub-topics. The Spark distributed computing framework is used to improve the single-node algorithm to run in a distributed environment, speeding up the algorithm running and improving the algorithm running efficiency.

This paper carries out the research through four parts. The first part is an overview of the research status of HNTM methods. The second part is the research of HNTM methods based on the CWM - text mining (TM). The third part is to verify the performance of the system designed by the research. The fourth part is the conclusion.

2. Related works

Hot spot mining algorithm has always been an important direction of academic research. It can efficiently extract hot topics from massive data, and plays a very important role in public opinion monitoring, recommendation system, big data analysis, etc. Jia et al. [6] developed an internet hotspot event mining and analysis technology with the advantages of integrating redundant information and extracting core information. This technology utilized PAT-Tree technology to extract high-frequency keywords, and then used event triplets as candidate elements to extract key sentences in hot topics. Based on this, a main service channel model based on word graphs was constructed. After experimental analysis, this technology could predict the trend of topic popularity [6]. To more accurately recommend hot news articles to users, Manoharan et al. [7] proposed an algorithm based on fuzzy logic, which predicted and classified users' interests by analyzing user characteristics, and then recommended them in a variety of ways. The findings denoted that the performance of this algorithm in measuring the

overall user interest of all categories reached 84.238% [7]. To prevent news from having more negative effects on people, De et al. conducted emotional analysis on hot news, classified Twitter data through four emotional analysis tools such as TextBlob, and found that March 2020 had the largest negative polarity and January 2020 had the largest enthusiasm [8]. Wang et al. [9] improved the k-means++ algorithm to solve the problems such as insufficient clustering accuracy and long algorithm time in the passenger transport hotspot clustering research. First, they established a dynamic adjustable region, then used the Gaussian Mixture model for data distribution statistics, and finally used the k-means++ algorithm to complete the clustering of local regions. The results indicated that the algorithm could provide higher accuracy in the same time [9]. He et al. [10] designed an improved LeaderRank algorithm to better control network public opinion and maintain network security. By mining key nodes in the network community, combined with the topology properties and average performance of the network, it could reflect indicators such as the number of likes. The outcomes denoted that the algorithm could reflect the quality of nodes well, indicating its practicality [10].

In the era of big data, people are also increasingly focusing on HNTM, and related research is also getting more attention. At present, researchers have proposed some related technologies and algorithms in this field. To promote the development of the cultural industry, Park S D et al. collected and analyzed the keyword policy discourse of China's creative industries from 2006 to 2020 through TM technology, and found that China's cultural industry was accelerating its development, constantly diversifying and humanizing. At the same time, they pointed out that attention should be paid to the development of the industrial model based on cultural experience to prevent the imbalance of regional development [11]. To quickly analyze the causes of accidents and determine the key factors, Xu et al. [12] designed a TM method that could automatically classify and predict the causes of accidents through in-depth learning. This method used the potential Dirichlet distribution model to extract keywords in accidents, and then extracted the key causes of accidents through convolutional neural network. Managers developed measures to reduce accidents through the extracted key causes [12]. Macêdo et al. [13] designed a prevention guarantee system for the problem that harmful substances may be released out of control. First, quantitative risk analysis was carried out through existing data, and then key risk factors were extracted by TM and fine-tuning training bidirectional encoder from transformer model. After practical verification, the system was very promising [13]. To understand and discover the research direction of future model-based system engineering, Akundi et al. [14] identified articles related to model-based system engineering through TM technology, classified the published literature and identified six topics. They found that "SysML", "network Physical system" and "production" were the most commonly used terms in model-based system engineering. To study the new trend of commodity development and formulate relevant marketing strategies, Muoz-Leiva et al. [15] analyzed the common words and subject networks of relevant articles in the Scopus database and Web of Science database through TM technology. The results indicated that virtual reality and augmented reality based on neural science methods were increasingly used in the field of commodity marketing, which is of great significance to the development of marketing discipline.

In summary, although researchers have proposed many methods to improve the mining algorithm of hot news topics, and have also made some achievements, they are still lacking in accuracy, effectiveness and efficiency. Therefore, through the mining technology of news media hot information based on the CWM-TM, it is expected that the hot information of news media can be quickly and accurately controlled.

3. Research on news media hotspot information mining method based on CWM-TM

By improving SIFRANK algorithm, the research designs a hot spot acquisition method based on co-occurrence word model-text mining (CWM-TM). First, it developed a new word discovery algorithm to discover new words, and then used the pre-training model ELMo to achieve high-quality key phrase extraction. To enhance the effect of mining hot words in the social network media environment, it also designs a HNTM algorithm based on CWM and improved SIFRANK (HCS), and finally introduces the Spark computing framework (SCF) to improve computational efficiency.

3.1. Design of short text HNTM method based on improved SIFRANK algorithm

Massive short texts have characteristics such as high dimensionality, sparsity, insignificant sentence structure, high noise, and unpredictable number of topics. Existing topic discovery algorithms have difficulty in meeting the requirements of fast and unsupervised hot topic discovery [16]. A hot word extraction method based on the improved SIFRANK algorithm is designed for massive short texts. The SIFRANK algorithm reduces the weight of high frequency words by performing inverse word frequency smoothing on the word vector. It has excellent performance in extracting keywords from text, especially on short texts that are more affected by high word frequency and invalid words. The basic framework for improving the SIFRANK algorithm is shown in Figure 1. After preprocessing such as word segmentation and stop word removal, the short text gets phrases with parts of speech, and then uses the ELMo training model to turn these phrases into word vectors, uses SIF method to weight the word vectors, to transform them into sentence vectors. Finally, the word and sentence vectors are compared for cosine similarity, and sorted according to the similarity to get short text hot words.

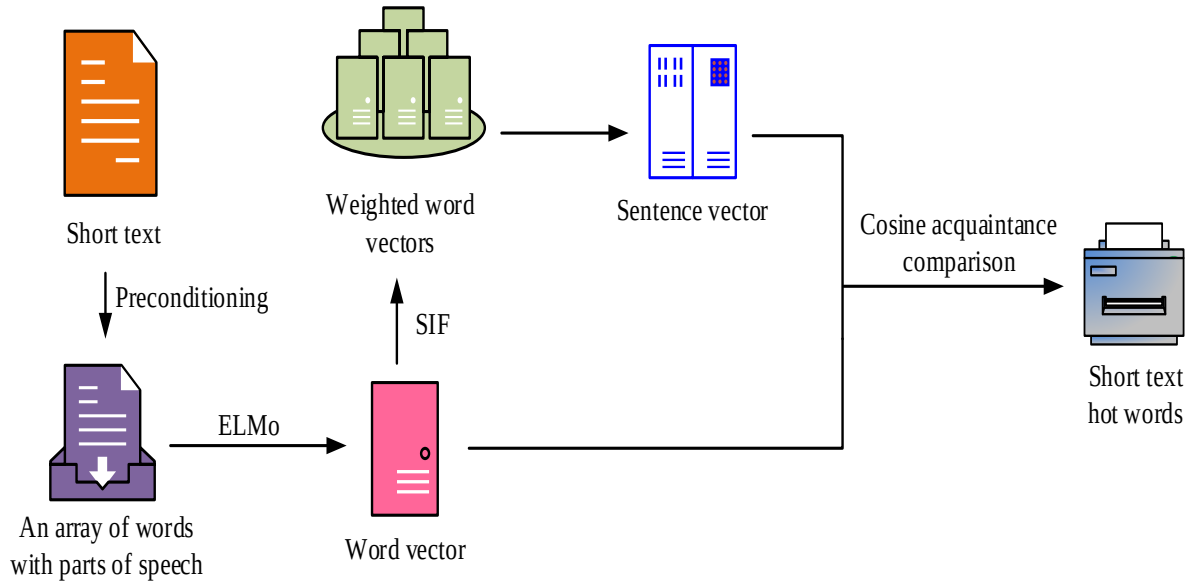


Figure 1. Basic framework of SIFRANK algorithm.

ELMo, as an unsupervised pre-training model, can train the same word using different statements to obtain different word vectors, effectively distinguishing the different meanings represented by the same word in different contexts. It can also save corpus and significantly improve the performance of downstream tasks [17,18]. In real social networks, there are often a large number of abbreviations, colloquial expressions, and network phrases that do not appear in formal word databases. Therefore, these words can interfere with the SIFRANK algorithm's extraction of short text hot words, and the same word has different levels of criticality in different sentences. To achieve better hot word extraction results, it is also necessary to enhance the semantic vector of the word.

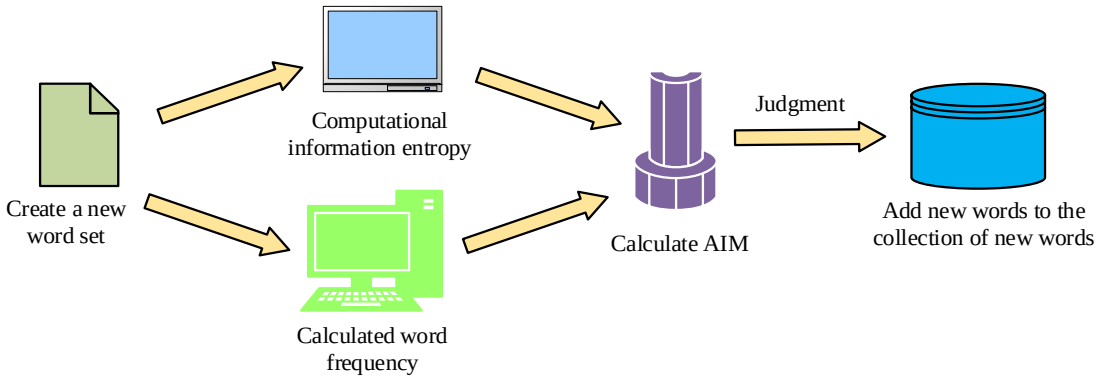


Figure 2. Neologism algorithm framework.

The study defines words that do not appear in the word database as new words, whether online, spoken or specialized. In order to make the subsequent tasks as free as possible from the interference of this new word, it is necessary to identify the new word in advance. The specific algorithm flow is shown in Figure 2. A key feature of new words is their rich left and right collocations, which means they have a rich amount of information. Therefore, information entropy is introduced here, and the specific calculation method is shown in Eq (1).

$$\text{Entropy}(w) = -\sum_{w_n \in W_{Neighbor}} P(w_n|w) log\, P(w_n|w). \tag{1}$$

In Eq (1), $\text{Entropy}(w)$ means information entropy, w denotes words, $W_{Neighbor}$ expresses the set of adjacent words around the words, w_n represents word frequency. At the same time, due to the frequent use of words such as "de" and "shi" in Chinese, and their frequent mixing with other words, the frequency of coexisting with other words is too high. This should also be avoided as much as possible. With the mutual information between points as the statistical criterion, the feature combination is quantitatively analyzed, and the specific mathematical expression is Eq (2).

$$PMI = p(x,y) log \frac{p(x,y)}{p(x)\,p(y)}. \tag{2}$$

In Eq (2), PMI means mutual information between points, $p(x,y)$ indicates the probability of the simultaneous occurrence of the words x and y, $p(x)$ expresses the probability of the occurrence of the word x, $p(y)$ refers to the probability of the occurrence of the word y. From Eq (2), the higher

the frequency of word combinations appearing together, the greater the PMI. If the frequency of individual words is too high, the PMI will decrease. To determine whether a word combination is a new word, it is necessary to first calculate the left and right information entropy of each candidate word, and then comprehensively determine the internal cohesion of each candidate word. The specific calculation formula is shown in Eq (3).

$$L(w) = log \frac{LE + RE + 1e-8}{|LE - RE| + 1e-8}. \tag{3}$$

In Eq (3), LE means the left information entropy, RE refers to the right information entropy, $|LE - RE|$ can be expressed as the possibility of word formation. When the length of candidate words is longer, the PMI will be higher. To accurately express the internal cohesion of each candidate word, the research introduces mutual information between average points, and the mathematical expression is Eq (4).

$$AMI = \frac{1}{n} log \frac{p(W)}{p(c_1)\,p(c_2)\ldots p(c_n)}. \tag{4}$$

In Eq (4), AMI is the Mutual information between the average points, and W is the word combination. Finally, the calculation equation for determining whether a candidate word is a new word is Eq (5).

$$score = \alpha L(w) + \beta AMI. \tag{5}$$

In Eq (5), $score$ stands for the score; α and β represent the coefficients considered to be set to control the importance of $L(w)$ and AMI respectively. By calculating the scores of the word and its sub words $score_{w_1}$ and $score_{w_2}$, it can be determined whether the word is a new word. When $score_W \geq score_{w_1} + score_{w_2}$ is used, the word can be considered a new word.

After the determination of new words, it is also necessary to eliminate the unusable data such as garbled code, expressions and symbols, and then segment the text, remove the stop word, and finally get the usable text phrases. Using Bert as a pre-training model, word vectors are generated from words in text phrases, and SIF is used to weight the word vectors. Considering the importance of heat measurement, word importance measurement is added to the weighting formula, as shown in Eq (6).

$$Weight_w = \alpha TF - IDF + \beta p(w) + \gamma log\left(k_1 n_{like} + k_2 n_{relay} + k_3 n_{fans}\right). \tag{6}$$

In Eq (6), $Weight_w$ denotes the weight of the word; $TF - IDF$ expresses the word frequency inverse document frequency; $p(w)$ represents the frequency of the word appearing in the text; n_{like} means the number of likes; n_{relay} refers to the number of reposts; n_{fans} means the number of followers of the topic publisher, α, β, γ, k_1, k_2, and k_3 represent the weight coefficient. By weighting and adding the word vectors, we can obtain the sentence vector. Finally, it compares the cosine similarity between the word vector and the sentence vector, sets the threshold point ε, and extracts the words with cosine similarity greater than ε or the words with the highest cosine similarity as the final hot word extraction result.

In general, the specific steps for key phrases are as follows. First of all, the corresponding left and right neighbors of all words are recorded to create a new word set. The left and right neighbors of each

word were recorded, as well as the number of occurrences of each neighbor word, and the left and right information entropy of each word was calculated according to the proportion of each word in the left and right neighbors of each word. The information entropy combined with two adjacent words w_1 and w_2 is recorded, that is, the left information entropy of w_1 and the right information entropy of w_2. Then, the word frequency of two adjacent words and the word frequency of adjacent words are calculated, AMI is calculated by word frequency, and $L(w)$ is calculated by information entropy. If $AMI(w)+L(w)>AMI(w_1)+L(w_1)+AMI(w_2)+L(w_2)$ is satisfied, the new word set is added; Otherwise, repeat the previous operation.

3.2. News hotspot mining algorithm based on CWM

To enhance the mining effect of SIFRANK algorithm on news media hotspots, a CWM is introduced in the study. Traditional algorithms can quickly process and obtain results in long texts with complete sentence structures [19,20], while HCS algorithm is more suitable for extracting hot words in short texts. However, in the modern news media and social network environment, there are more short texts with short length, more confounding and less information. When traditional algorithms process such texts, they cannot meet the performance and efficiency requirements. Figure 3 is the HCS algorithm framework.

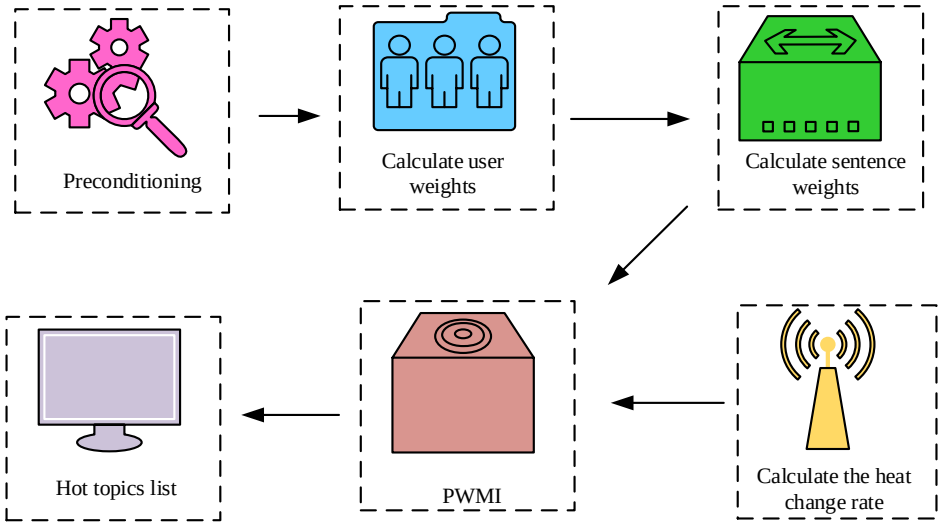


Figure 3. HCS algorithm framework.

The HCS algorithm needs to determine that the extracted hot topic phrase is indeed being frequently discussed or used by people and has a certain degree of representativeness, so it is necessary to calculate its popularity [21,22]. The specific expression is shown in Eq (7).

$$PIM = log\frac{p(w_1,w_2,\ldots w_n)}{p(w_1)p(w_2)\ldots p(w_n)}. \tag{7}$$

In Eq (7), $p(w_1,w_2,\ldots w_n)$ means the frequency at which w_1, w_2, and $\ldots w_n$ appear in the same sliding window, while $p(w_n)$ denotes the frequency at which words appear in the entire text corpus.

Eq (7) has to some extent filtered out some versatile words, thus retaining words with strong relevance and uniqueness.

Table 1. Weight classification of users' fans and followers.

Number of followers or followers	Weight
$x \leq 10$	1
$10 < x \leq 100$	2
$100 < x \leq 500$	3
$500 < x \leq 1000$	4
$1000 < x \leq 10000$	5
$10000 < x \leq 50000$	6
$50000 < x \leq 200000$	7
$200000 < x \leq 500000$	8
$500000 < x \leq 1000000$	9
$1000000 \leq x$	10

If word frequency is used as a statistical variable, it cannot rule out malicious screen swiping and forwarding by online water armies and marketing accounts [23]. The posts of these users are often different from those of ordinary users. They have stronger written content, more single content, and are easy to get high scores in heat calculation. Therefore, research has improved the calculation equation. The more active a user is, the greater the likelihood of spontaneously participating in hot topics. The more followers a user has, the less likely they are to become an online influencer or marketing account. The weight grading of the user's fan count and followers is shown in Table 1. By taking the number of non-repetitive blog posts posted by a user on a certain social platform over the past 180 days as a weight, the user is evaluated using the mathematical calculation expression shown in Eq (8).

$$weight_{user} = log(1+D). \tag{8}$$

In Eq (8), D is the amount of non-repetitive blog posts posted on a certain social platform in the past 180 days. For more accurate judgment, the number of fans and followers of the user can also be considered. Therefore, the weight calculation expression for the user is Eq (9).

$$weight_{user} = \alpha\, log(1+D) + \beta\, log\left(1+weight_{fans}+weight_{follower}\right). \tag{9}$$

In Eq (9), $weight_{fans}$ is the weight of the number of followers, $weight_{follower}$ refers to the weight of the amount of followers, α and β are the weight coefficients, and $\alpha+\beta=1$. If a certain word corresponds to a sentence that mentions these words, and the expression is exactly the same, or even similar, then this popular word is likely to be maliciously swiped by navy, marketing accounts, or advertisements. So it also needs to consider the weight of the sentence, and the calculation method is as shown in Eq (10).

$$weight_{sentence} = 1+\frac{D_{sentence}}{C_{sentence}}. \tag{10}$$

In Eq (10), $D_{sentence}$ stands for the amount of sentences that do not involve candidate wordsand $C_{sentence}$ is the amount of sentences that involve candidate words. The closer $\frac{D_{sentence}}{C_{sentence}}$ approaches to 1, the more realistic the topic of discussion. The same sentence is spoken by different people and represents different weights, so Eq (11) can be derived for calculating the weight of words can be derived.

$$weight_{word} = log\left(1 + k_1 SIFRANK + k_2 weight_{user}\right). \tag{11}$$

In Eq (11), k_1 and k_2 are proportional coefficients, and $k_1 + k_2 = 1$. By combining the above weights, an expression for the comprehensive weight of words can be obtained. The calculation method is shown in Eq (12).

$$score = \alpha weight_{user} + \beta weight_{sentence} + \gamma weight_{word}. \tag{12}$$

In Eq (12), α, β, and γ are all proportional coefficients, and $\alpha + \beta + \gamma = 1$. At the same time, the calculation equation of mutual information of weights between points can be obtained, as shown in Eq (13).

$$PWMI = log \frac{score_{w_1 w_2 \ldots w_n}}{score_{w_1} score_{w_2} \ldots score_{w_n}}. \tag{13}$$

When mining hot topics, it is important to note that in a specific social network media, some topics that are already prone to arousing enthusiastic discussions among users usually pay more attention to the popularity of the topic, other related subtopics, or in the heat count. Although not as high as traditional topics, the heat has significantly increased, so it is necessary to add the heat change rate, as shown in Eq (14).

$$Ratio(words) = \sum_{1}^{n-1} t\left(\frac{PWMI_{words_{t+1}} - PWMI_{words_t}}{PWMI_{words_t} + 1e-7} - \frac{PWMI_{words_t} - PWMI_{words_{t-1}}}{PWMI_{words_{t-1}} + 1e-7}\right). \tag{14}$$

In Eq (13), n denotes the size of the time window. Using $Ratio(words)$ as a proportional coefficient, the expression for calculating the heat of the topic can be obtained, as shown in Eq (15).

$$PWMI_{nenw} = PWMI_{word} * Ratio(words). \tag{15}$$

In the final output result, if there are multiple candidate statements from the same short text, selecting the highest candidate statement to output can obtain the high freshness hot topic popularity ranking list. If the output candidate statements are not from the same short text, it can obtain the hot topic popularity ranking list.

When processing a large amount of data, the HCS algorithm may reduce its operational efficiency and result in the loss of timeliness in the final results. To avoid this situation, the research utilizes the distributed multi-core computing capability under Spark to perform parallel calculations on user weights, word weights, sentence weights, and other indicators. This method utilizes the characteristics of the original calculation results to accelerate the operational efficiency of the HCS algorithm.

In HSIFRANK, the subsidiary task Bert has been used to generate the word vector table to speed up the running of the main algorithm [24]. However, when calculating the popularity of hot words, if the data set is too large, then the number of cycles may be too many, the calculation is too complicated,

and the running speed of the algorithm is slow [25]. In HSIFRANK, the main performance consumption is concentrated in the calculation of the adjacent words, word frequency, word vector and other statistical indicators after word segmentation. Therefore, HSIFRANK is applied under the distributed computing framework Spark. With the distributed multi-core computing capability under Spark, text preprocessing can be carried out in parallel, and indicators such as information entropy, word frequency, user weight and tweet forwarding likes can be counted. Meanwhile, due to the sequential relationship between steps, such as word frequency calculation, new word discovery, and TF-IDF, this data parallel computing mode is very conducive to the utilization of Spark's memory-based computing features, which can reuse the original calculation results, and accelerate the operation efficiency of the algorithm.

During the execution of the algorithm, Spark will divide the data set into multiple subsets and store them in the RDD. After that, the operators provided by Spark are used to preprocess and segment the text using map operation, and groupByKey is used to calculate and record the adjacent words and word frequency of each word. On this basis, the score of new words is calculated to get the dictionary of new words. In the new word discovery task, the new word and word vector dictionary are obtained at the end. All the previous operation calculation steps can be distributed calculation using RDD to speed up the calculation. In the task of hot word mining, word frequency can be quickly calculated based on the previously discovered new words and the created word vector dictionary. Then, various operators of RDD can be operated according to the corresponding topological relationship. For example, when calculating the corresponding weight of each user, the number of followers and fans of each user can be counted first, and then the formula can be calculated. Thus, you can get the corresponding results quickly.

In order to test the HSIFRANK algorithm proposed in Chapter 3, this paper selects three benchmark algorithms SIFRANK, TextRank and TF-IDF, all of which are common algorithms for keyword extraction. Through comparative experiments, the performance of the new word discovery module is tested first, including the number of large-scale new words discovered, the accuracy of word segmentation after the discovery of new words, the recall rate and F1 value. Besides, the performance of HSIFRANK in hot word extraction after the discovery of new words is tested. The performance test includes the accuracy rate of hot word extraction, the recall rate and F1 value. The concept of measurement index is derived from the confusion matrix shown in Table 2. The formula for accuracy, recall and F1 value is as follows Eq (16).

$$\begin{cases} \mathrm{Pr}\,ecision = \dfrac{TP}{TP + FP}, \\ \mathrm{Recall} = \dfrac{TP}{TP + FP}, \\ F1 = \dfrac{2 * \mathrm{Pr}\,ecision * \mathrm{Recall}}{\mathrm{Pr}\,ecision + \mathrm{Recall}}. \end{cases} \tag{16}$$

Table2. Confusion matrix.

Confusion matrix		True value	
		Positive	Negative
Predicted value	Positive	*TP*	*FP*
	Negative	*FN*	*TN*

4. Performance verification of news media hotspot mining system

The study selected four different types of experimental datasets and compared and analyzed the advantages and disadvantages of the improved SIFRANK algorithm with other algorithms in terms of runtime, mining accuracy, recall, and F1 value. Then the acceleration ratio of the improved SIFRANK and HCS algorithms was analyzed when using the SCF.

4.1. Performance analysis of short text hot word extraction method

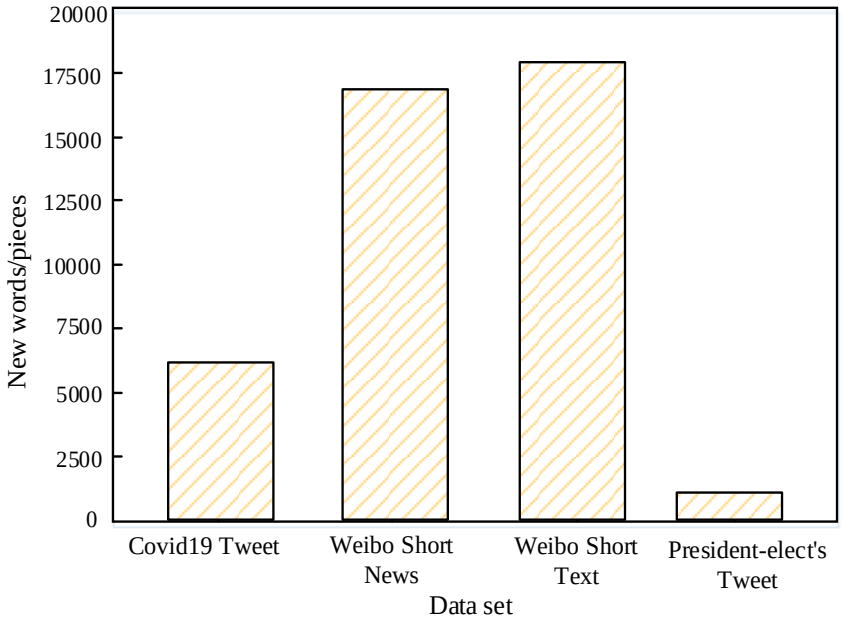
The CPU used in the experiment was Core (TM) i5-4790K CPU@4.00GHZ. The GPU was Nvidia GeForce GTX1060Ti, and the computer operating system was Windows11 64 bit. Four datasets were selected for testing in the experiment, and their specific parameters are shown in Table 3. Among them, the COVID-19 tweet is the data obtained from Twitter, and the time span is divided into before March 12, 2020 and from March 12, 2020 to March 22, 2020. It belongs to the topic focused data set with a short time span. The data set contains text, Twitter-related information such as sending time, and the following information: The number of likes, etc., user information such as the number of user fans, etc. Weibo Short News data set is a data set that releases short news through microblog accounts. The topics are relatively sparse, and the data set contains text and text content summary. The Weibo Short Text data set is a massive short text data set randomly crawled from microblog, belonging to a large sparse topic data set, which contains text and text content abstract. The President-elect's Tweet data set is a data set of tweets discussing the two candidates of the US presidential election: Trump and Biden. It is a data set of medium topic intensity in a short and medium time span. The data set includes text, Twitter-related information such as sending time and number of likes, and user information such as the number of users' followers.

Table3. Details of data set parameters.

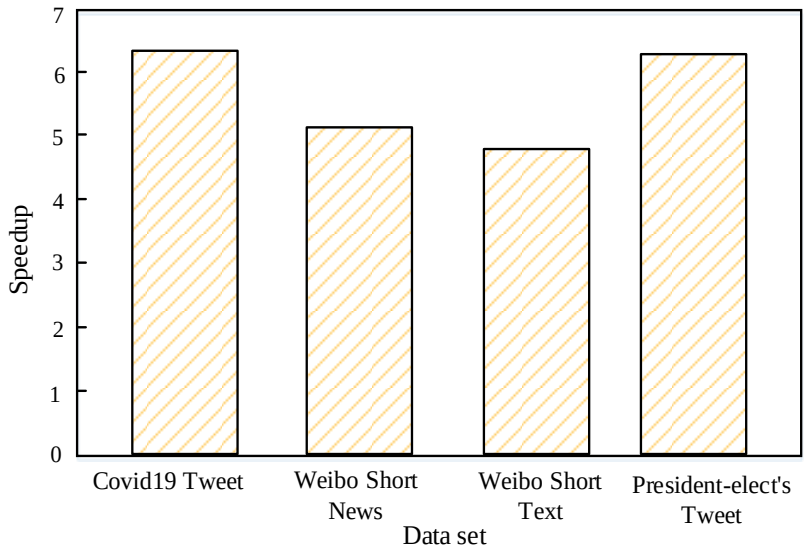
Data set	Number of entries	File size /KB
Covid19 Tweet	2880000	1522908
Weibo Short News	600000	221669
Weibo Short Text	1000000	699817
President-elect's Tweet	800000	626784

Figure 4 shows the performance analysis of the new word discovery algorithm. Figure 4(a) shows the results of the amount of new words discovered by the new word discovery algorithm in four datasets. As shown in the figure, the new word discovery algorithm found far more new words in the Weibo Short News and Weibo Short Text datasets, reaching 16871 and 17921, respectively. This was because English text had spaces as the basis for word segmentation, so only some combination words

could be found. Another reason was that in the Chinese dataset, its unique network vocabulary, spoken language, and abbreviations and other words would be frequently used, and these would be separated by new word algorithms, which traditional word segmentation tools could not achieve. The number of new words in Covid19 Tweet was 6521, which was higher than that in President-elect Tweet, mainly because the Covid19 Tweet dataset had a higher amount of items. Figure 4 (b) shows the acceleration ratio between the original new word algorithm and the new word algorithm using the SCF when the amount of computing cores was 8. From the figure, the acceleration of Covid19 Tweet and President-elect Tweet was relatively large. Overall, the application of the SCF has significantly improved the efficiency of the new word algorithm.



(a) Comparison of the number of new words discovered by neologism discovery algorithm



(b) The acceleration ratio of new word discovery algorithm under Spark framework

Figure 4. Performance diagram of new word discovery algorithm.

The experiment extracted 20000 texts from Covid19 Tweets and President-elect's Tweets, and set the keyword extraction number to 3. The accuracy, recall, and F1 values of the improved SIFRANK, TF-IDF, TEXTRANK, SIFRANK, and SIFRANK+ algorithms were compared in mining keywords. Figure 5 shows the performance comparison of algorithms based on Covid19 Tweets. The traditional TF-IDF algorithm was not as good as the other four algorithms in various indicators. The accuracy, recall, and F1 value of the SIFRANK algorithm were the highest, reaching 0.6735, 0.7550, and 0.7118, respectively. The improved SIFRANK algorithm was slightly worse than the SIFRANK and SIFRANK+ algorithms, with an accuracy of 0.6223, a recall of 0.7015, and an F1 value of 0.6605.

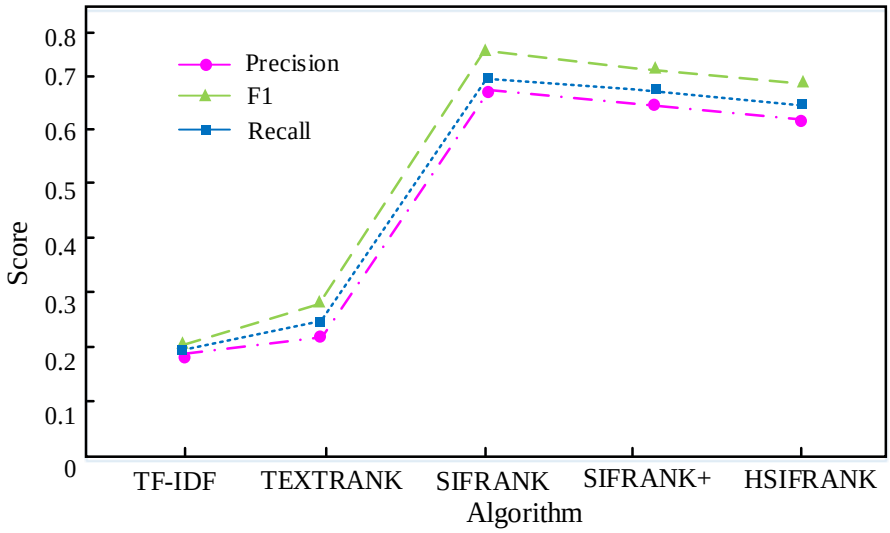


Figure 5. Improved SIFRANK keyword mining performance test based on the Covid19 Tweet dataset.

Figure 6 is a comparison chart of algorithm performance based on President-elect Tweet. As shown in the figure, the performance of the traditional TF-IDF algorithm was still the worst in this dataset, while the SIFRANK algorithm was still superior to other algorithms, with accuracy, recall, and F1 values reaching 0.6735, 0.7550, and 0.7118, respectively. Through comprehensive analysis of Figures 5 and 6, the improved SIFRANK algorithm was slightly inferior to the original SIFRANK algorithm. This was because the ELMo model used by SIFRANK was obtained in the context of this dataset, and the word vector of each word would change with the change of the sentence, thus achieving the effect of solving polysemy of a word. However, the word vectors in HSIFRANK were obtained through a pre-trained Bert model, which did not fully fit the context of the original text. Therefore, its performance was slightly worse than SIFRANK in this regard. From the overall results, the performance of the two algorithms was not significantly different.

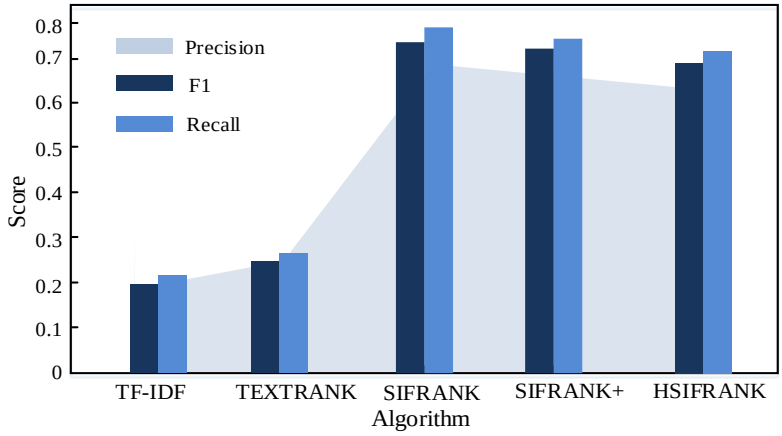


Figure 6. Improved SIFRANK keyword mining performance test based on the President-elects Tweet dataset.

The experiment selected the three words with the highest heat intensity from the Covid Tweet dataset and compared their heat weight values under five algorithms. The specific results are shown in Table 4. From the table, compared to SIFRANK, the improved SIFRANK could better mine popular vocabulary, and the keywords obtained from it had the highest heat weight values, reaching 0.9356, 0.9991, and 0.6117, respectively. This was mainly because the improved SIFRANK algorithm increased the weight of word importance and occurrence frequency. Meanwhile, because the improved SIFRANK algorithm obtained word vectors by segmenting each subtask, the algorithm only needed to use table lookup when performing operations. This improvement greatly improved the execution efficiency of the improved SIFRANK algorithm, which could better meet the requirements of real-time performance.

Table 4. Keyword mining performance test.

Algorithm	Covid19	Coronavirus	Coronavirus outbreak
TF-IDF	0.0772	0.0524	0.0103
TEXTRANK	0.1482	0.1573	0.0823
SIFRANK	0.84491	0.9586	0.6049
SIFRANK+	0.8026	0.8772	0.5987
HSIFRANK	0.9356	0.9991	0.6117

4.2. Performance verification of news hotspot mining algorithm

Figure 7 shows the popularity changes of two topics mined by the HCS algorithm from the Covid Tweet dataset from 12th to 18th. From the figure, the keyword social distance, which was not mentioned too much from the 12th to the 17th, suddenly rose in popularity on the 18th, because with the spread of COVID-19, people have realized the danger of this disease and started to advocate social distancing. Coronavirus, as a topic related to the epidemic itself, has always had a high popularity. One important reason was that the HCS algorithm could timely identify topics with relatively low popularity but high change rates.

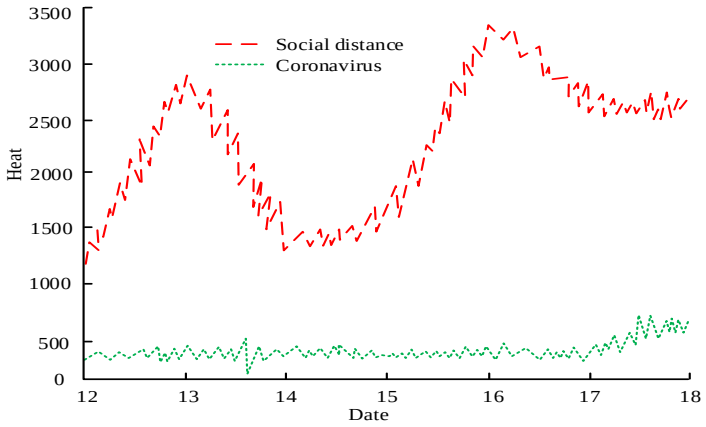
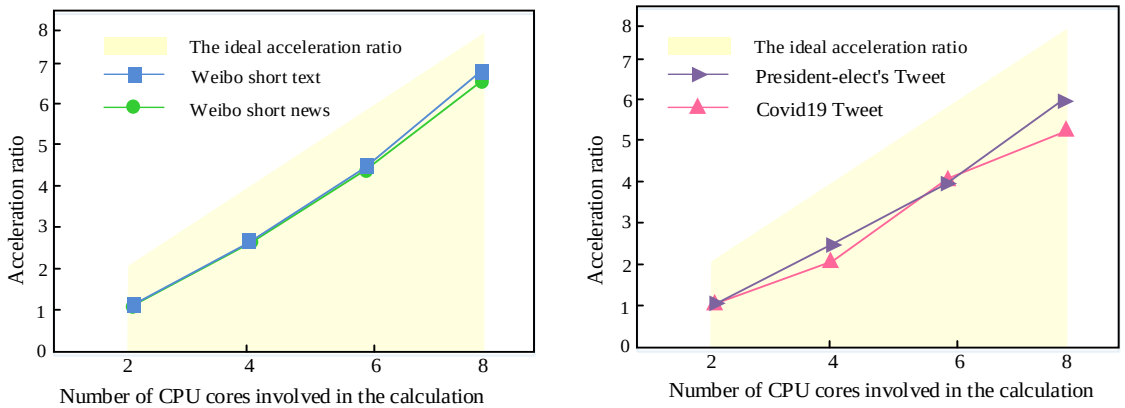


Figure 7. Heat variation in coronavirus and social distance.

Figure 8 shows the acceleration ratios obtained when the improved SIFRANK and SIFRANK algorithms were tested on different datasets under the SCF. From Figure 8(a), in the Chinese dataset, as the number of CPU cores participating in the calculation increased, the acceleration ratio gradually increased, and the algorithm's acceleration ratio was relatively close to the ideal acceleration ratio. From Figure 8(b), in the Chinese dataset, as the number of CPU cores involved in the calculation increased, the acceleration ratio gradually increased. However, the algorithm's acceleration ratio was not as close to the ideal value as in the Chinese dataset. The reason was that the topics in the Chinese dataset were relatively sparse, and sparse topics could better divide the data into smaller and more detailed blocks, accelerate Spark's calculation speed, and further improve the algorithm's acceleration ability. Overall, the acceleration ratio of algorithms could not increase exponentially as the number of CPU cores increased like the ideal acceleration ratio did. This was because when the computing power reached a certain upper limit, the communication consumption between work nodes, data structure construction consumption, and other limitations limited the increase in acceleration ratio.



(a) Acceleration ratio based on Chinese data set

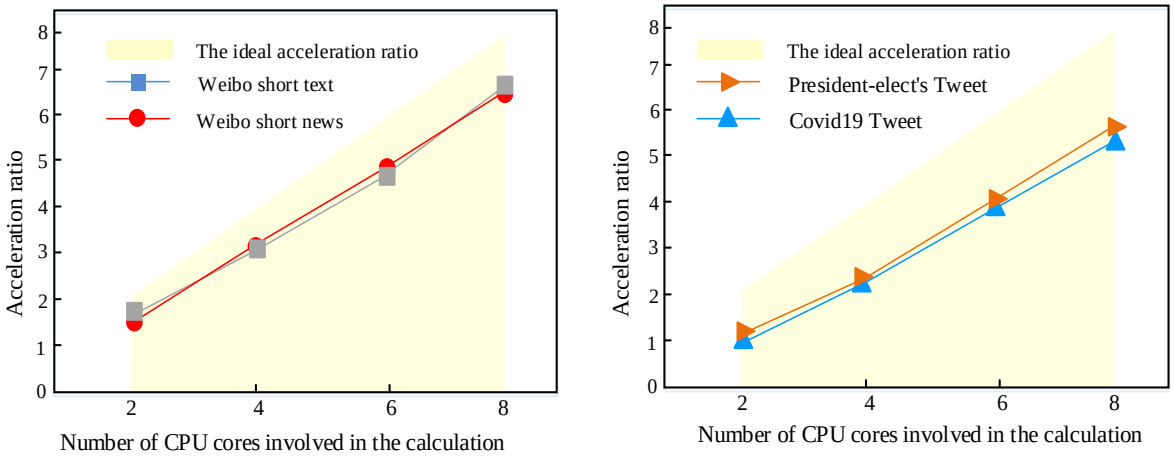
(b) Acceleration ratio based on English data set

Figure 8. Spark-HSIFRANK algorithm acceleration ratio.

Figure 9 shows the growth trend of the Spark-HCH algorithm's acceleration ratio compared to the HCS algorithm. From Figures 9(a) and (b), the acceleration ratio of the Spark-HCH algorithm could increase with the increase of CPU cores, regardless of whether it was a Chinese dataset or an

English dataset. Because the Spark-HCH algorithm itself included more aggregation operations, such as deduplication, group counting, and group scoring for the same topic, it could achieve better acceleration on Chinese datasets with high topic sparsity, thus fully utilizing these additional CPU cores to accelerate the algorithm.

In order to verify the computational performance of the proposed method in processing large-scale data, we compare it with Lda2vec algorithm [26] and LDA topic classification visualization method [27]. Figure 10 is a comparison of CPU usage in the Covid19 Tweet dataset with different integrity levels. As can be seen from Figure 10, the CPU usage of the three algorithms increases as the integrity of the data set increases. However, compared with the other two algorithms, the increase trend of the improved method is slower. In general, the proposed improvement method uses less computing resources and can meet the requirements of general use.



(a) Acceleration ratio based on Chinese data set

(b) Acceleration ratio based on English data set

Figure 9. Spark-HCS algorithm acceleration ratio.

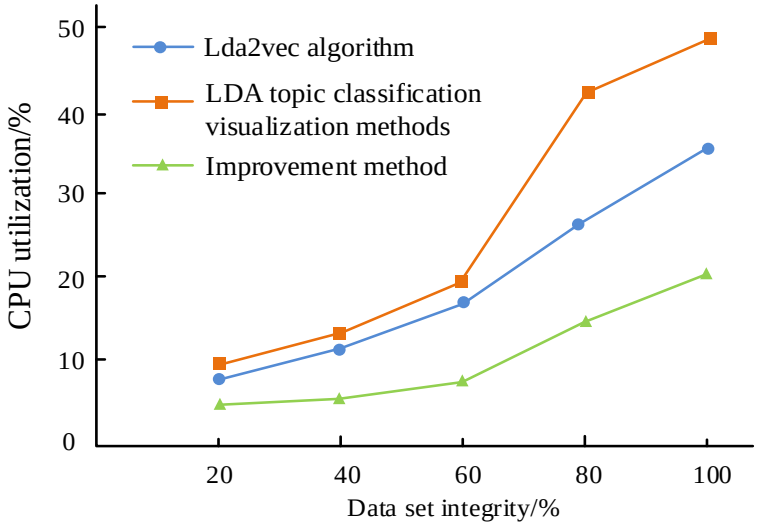


Figure 10. Computing resource comparison results.

5. Conclusions

To improve the accuracy and effectiveness of hots news topic mining, this paper studied a method of news media hot information mining based on co-occurrence word model-text mining. On this basis, the Spark computing framework was introduced to improve the computing efficiency. The experiment outcomes expressed that the new word discovery algorithm found 16871 and 17921 new words in the Weibo Short News and Weibo Short Text datasets, much more than in the English dataset. The acceleration of the new word algorithm using the Spark computing framework had significantly increased. In the Covid19 Tweets dataset, the improved SIFRANK algorithm had an accuracy of 0.6223, a recall rate of 0.7015, and an F1 value of 0.6605. In the President-elect Tweets dataset, the SIFRANK algorithm still outperformed other algorithms, with accuracy, recall, and F1 values of 0.6735, 0.7550, and 0.7118, respectively. The heat weight values of the keywords obtained by the improved SIFRANK reached 0.9356, 0.9991, and 0.6117. After applying the Spark computing framework, the running speed maintained a linear improvement, but the acceleration of the Chinese dataset was closer to the ideal value than the English dataset. Under the social network media, the improved SIFRANK and HCS algorithms had better performance than the traditional algorithm, and they were more suitable for completing the task of text mining under the social network media. In practical scenarios, the algorithm proposed in this study is difficult to distinguish the same topic represented by synonyms, and it is easy to remember them as different topics. In the future, it is considered to use the combination of Embedding, community discovery and so on to mine synonyms to construct the synonym thesaurus in the same context. On the basis of obtaining a thesaurus, the output topics are combined with similar topics to make the final output results less similar.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. B. Dadashova, C. Silvestri-Dobrovolny, J. Chauhan, M. Perez, R. Bligh, Hot-spot analysis of motorcyclist crashes involving fixed objects using multinomial logit and data mining tools, *J. Transp. Saf. Secur.*, **36** (2021), 10–29. https://doi.org/10.1080/19439962.2021.1898070
2. M. Saeed, M. R. Ahmad, A. U. Rahman, Refined pythagorean fuzzy sets: Properties, set-theoretic operations and axiomatic results, *J. Comput. Cogn. Eng.*, **2** (2022), 10–16. https://doi.org/10.47852/bonviewJCCE2023512225
3. S. Choudhuri, S. Adeniye, A. Sen, Distribution alignment using complement entropy objective and adaptive consensus-based label refinement for partial domain adaptation/artificial intelligence and applications, **1** (2023), 43–51. https://doi.org/10.47852/bonviewAIA2202524

4. S. Oslund, C. Washington, A. So, T. Chen, H. Ji, Multiview robust adversarial stickers for arbitrary objects in the physical world, *J. Comput. Cogn. Eng.*, **1** (2022), 152–158. https://doi.org/10.47852/bonviewJCCE2202322
5. X. Wang, M. Cheng, J. Eaton, C. J. Hsieh, S. F. Wu, Fake node attacks on graph convolutional networks, *J. Comput. Cogn. Eng.*, **1** (2022), 165–173. https://doi.org/10.47852/bonviewJCCE2202321
6. Y. Jia, S. B. Tsai, Digital media hotspot mining algorithm implementation with complex systems in the mobile internet environment, *Complexity*, **4** (2021), 71–82. https://doi.org/10.1155/2021/3471168
7. S. Manoharan, R. Senthilkumar, An intelligent fuzzy rule-based personalized news recommendation using social media mining, *Comput. Intell. Neurosci.*, **2020** (2020), 3791541–3791550. https://doi.org/10.1155/2020/3791541
8. H. De, K. Deb, Does social media follow news media? A comparative sentiment analysis during the COVID-19 pandemic, *Int. J. Inform. Commun. Tech. Hum. Dev.*, **13** (2021), 72–82. https://doi.org/10.4018/IJICTHD.2021100102
9. Y. Wang, J. Ren, Taxi passenger hot spot mining based on a refined k-means++ algorithm, *IEEE Access*, **9** 2021, 66587–66598. https://doi.org/10.1109/ACCESS.2021.3075682
10. Y. He, T. Wang, J. Xie, M. Zhang, Research on mining key nodes of complex web-based communities based on mining algorithm, *Int. J. Web Based Commun.*, **16** (2020), 202–210. https://doi.org/10.1504/IJWBC.2020.107155
11. S. D. Park, Policy discourse among the chinese public on initiatives for cultural and creative industries: text mining analysis, *SAGE Open*, **12** (2022), 45–65. https://doi.org/10.1177/21582440221079927
12. H. Xu, Y. Liu, C. M. Shu, M. Bai, M. Motalifu, Z. He, et al., Cause analysis of hot work accidents based on text mining and deep learning, *J. Loss Prevent. Proc. Ind.*, **2** (2022), 104747–101458. https://doi.org/10.1016/j.jlp.2022.104747
13. J. B. Macêdo, M. das Chagas Moura, D. Aichele, I. D. Lins, Identification of risk features using text mining and BERT-based models: Application to an oil refinery, *Process Saf. Environ. Prot.*, **158** (2022), 382–399. https://doi.org/10.1016/j.psep.2021.12.025
14. A. Akundi, O. Mondragon, Model based systems engineering—A text mining based structured comprehensive overview, *Syst. Eng.*, **25** (2022), 51–67. https://doi.org/10.1002/sys.21601
15. F. Muñoz-Leiva, M. E. Rodriguez Lopez, F. Liebana-Cabanillas, S. Moro, Past, present, and future research on self-service merchandising: A co-word and text mining approach, *Eur. J. Marketing*, **55** (2021), 2269–2307.
16. X. M. Long, Y. J. Chen, J. Zhou, Development of AR experiment on electric-thermal effect by open framework with simulation-based asset and user-defined input, *Artif. Intell. Appl.*, **1** (2023), 52–57. https://doi.org/10.47852/bonviewAIA2202359
17. A. Islam, F. Othman, N. Sakib, H. M. H. Babu, Prevention of shoulder-surfing attacks using shifting condition using digraph substitution rules, *Artif. Intell. Appl.*, **1** (2023), 58–68. https://doi.org/10.47852/bonviewAIA2202289
18. A. M. Usman, M. K. Abdullah, An assessment of building energy consumption characteristics using analytical energy and carbon footprint assessment model, *Green Low-Carbon Econ.*, **1** (2023), 28–40. https://doi.org/10.47852/bonviewGLCE3202545

19. Y. Wang, Y. Liu, W. Feng, S. Zeng, Waste haven transfer and poverty-environment trap: Evidence from EU, *Green Low-Carbon Econ.*, **1** (2023), 41–49. https://doi.org/10.47852/bonviewGLCE3202668
20. V. D. Gazman, A new criterion for the ESG Model, *Green Low-Carbon Econ.*, **1** (2023), 22–27. https://doi.org/10.47852/bonviewGLCE3202511
21. J. Machicao, E. A. Corrêa Jr, G. H. B. Miranda, D. R. Amancio, O. M. Bruno, Authorship attribution based on life-like network automata, *Plos One*, **13** (2018), 1371–1381. https://doi.org/10.1371/journal.pone.0193703
22. L. V. C. Quispe, J. A. V. Tohalino, D. R. Amancio, Using virtual edges to improve the discriminability of co-occurrence text networks, *Physica A*, **562** (2021), 125344–125357. https://doi.org/10.1016/j.physa.2020.125344
23. J. Qiang, Z. Qian, Y. Li, Y. Yuan, X. Wu, Short text topic modeling techniques, applications, and performance: a survey, *IEEE Trans. Knowl. Data Eng.*, **34** (2020), 1427–1445. https://doi.org/10.1109/TKDE.2020.2992485
24. D. R. Amancio, O. N. Oliveira Jr, L. da F Costa, Using complex networks to quantify consistency in the use of words, *J. Stat. Mech.*, **2012** (2012), P01004. https://doi.org/10.1088/1742-5468/2012/01/P01004
25. H. Che, B. Pan, M. F. Leung, Y. Cao, Z. Yan, Tensor factorization with sparse and graph regularization for fake news detection on social networks, *IEEE Trans. Comput. Social Syst.*, **14** (2023), 1–11. https://doi.org/10.1109/TCSS.2023.3296479
26. M. Zhang, H. Su, J. Wen, Analysis and mining of internet public opinion based on LDA subject classification, *J. Web Eng.*, **20** (2021), 2457–2472.
27. Y. Qian, Z. Ni, W. Gui, Y. Liu, Exploring the landscape, hot topics, and trends of electronic health records literature with topics detection and evolution analysis, *Int. J. Comput. Intell. Syst.*, **14** (2021), 744–757. https://doi.org/10.2991/ijcis.d.210203.006

AIMS Press