Hindawi
Advances in Multimedia
Volume 2022, Article ID 7923262, 11 pages
https://doi.org/10.1155/2022/7923262

Research Article

Research on Text Similarity Measurement Hybrid Algorithm with Term Semantic Information and TF-IDF Method

Fei Lan

School of Electronics and Internet of Things, Chongqing College of Electronic Engineering, Chongqing 400000, China

Correspondence should be addressed to Fei Lan; lanfei@cqcet.edu.cn

Received 26 January 2022; Revised 24 February 2022; Accepted 3 March 2022; Published 23 April 2022

Academic Editor: Qiangyi Li

TF-IDF (term frequency-inverse document frequency) is one of the traditional text similarity calculation methods based on statistics. Because TF-IDF does not consider the semantic information of words, it cannot accurately reflect the similarity between texts, and semantic information enhanced methods distinguish between text documents poorly because extended vectors with semantic similar terms aggravate the curse of dimensionality. Aiming at this problem, this paper advances a hybrid with the semantic understanding and TF-IDF to calculate the similarity of texts. Based on term similarity weighting tree (TSWT) data structure and the definition of semantic similarity information from the HowNet, the paper firstly discusses text preprocess and filter process and then utilizes the semantic information of those key terms to calculate similarities of text documents according to the weight of the features whose weight is greater than the given threshold. The experimental results show that the hybrid method is better than the pure TF-IDF and the method of semantic understanding at the aspect of accuracy, recall, and F1-metric by different K-means clustering methods.

1. Introduction

Text similarity measurement is some way to measure the degree of semantic similarity between two texts, and it is a very important task for natural language processing. Text similarity measures have extremely widespread application in many fields, such as text duplicate detection field, image retrieval, information retrieval, the automatic generation of text areas, and text classification. There are statistical ways and semantic analysis algorithm in traditional text similarity measurement methods. In text similarity measurement method based on statistics, the whole text is regarded as a set of words. By analyzing the occurrences number of each term, the text model vector is constructed in terms of effective word frequency information. Moreover, the similarity of text vectors is calculated by cosine similarity or Jaccard coefficient. The model based on attribute theory, semantic index model, and vector space model belong to statistics method. For statistical similarity measurement, it expresses the text as a vector to simplify the complex relationship between the keywords in the text, by which the model is calculated easily [1]. However, the method ignored the meaning and semantic relationship of word item; it needs large scale word corpus to support. Due to the large number of words and texts, the vector dimension in the text representation model is extremely high so that it is difficult to handle directly. TF-IDF method is a traditional statistics-based text similarity measure algorithm, which constructed model by text word frequency vector, and the similarity of texts is calculated through cosine similarity measurement.

For text similarity measurement method based on semantic analysis, the semantic relationship of text word (e.g., synonym, redundancy, and inclusion) is set up by specified domain knowledge [2], and it also determines texts similarity degree. The advantage of this method is that the algorithm accuracy is very high and it does not depend on a large corpus to support. However, it is very easy to establish a knowledge base, which needs large scale and complex work. Thus, the current research generally adopts a complete dictionary with words rather than a knowledge base. Literature [3] introduced resolving process of text similarity based on WordNet and HowNet. Literature [4] put forward text similarity measurement by sememe space of HowNet. Literature [5] introduced text similarity measurement in terms of weighting semantic web. These methods considered the semantic information of word terms, but they ignored the different degree of importance to the various texts. The methods improved the vector dimension of the text representation, and they cannot reflect the similarity between the two texts.

According to the defects of the above method, a method that can effectively reduce the dimension of the text representation model and combine the semantic information of words terms is proposed. The algorithm proposed can efficiently and automatically calculate the similarity of the semantic texts, and there is a broad application prospect for the hybrid similarity measurement method.

2. Related Works

TF-IDF method is the most typical text similarity measure algorithm, and it represents the text as a vector composed of n weighted words terms that appear in the text by following empiric observation [6].

(1) Term Frequency. The more frequently a word appears in a text, the more relevant it is to the topic of the text. There are many specific words in specific linguistic environment that do not have this property and should be excluded, such as “a” and “an”.

(2) Inverse Document Frequency. The more times a term appears in multiple text in a text collection, the worse the term is. For example, in a collection including 10000 texts, if a term A is present in 1000 texts and another term B appears only in 10 texts, then term B is better discrimination than A.

By using the above concept, the TF-IDF value of every term ω_{i} can be calculated according to equations (1)–(3).

$$\mathrm{TF}-\mathrm{IDF}(\omega_i) = tf(\omega_i) \times idf(\omega_i) = tf(\omega_i) \times \log\left(\frac{N}{df(\omega_i)}\right), \tag{1}$$

$$tf(\omega_i) = \frac{n_{ij}}{\sum_{k=1}^{m} n_{kj}}, \tag{2}$$

$$df(\omega_i) = \left|\left\{j : \omega_i \in d_j\right\}\right|, \tag{3}$$

where $tf\,(\omega_i)$ is occurrence frequency of current term ω_i in text j, and N is total number of all text in text collection $\{d_j\}$. $df\,(\omega_i)$ indicates how many texts show term ωi in the text collection. n_{ij} is occurrence frequency of ith term appearing in jth document. n_{kj} is occurrence frequency of kth term appearing in jth document. $|\{j: \omega_i \in d_j\}|$TF-IDF is number of document including ith term. Value of each term in every text can be acquired by analyzing every term in text collection, and vector model of each text is constructed by the term TF-IDF value. Thus, the similarity of texts can be determined by calculating cosine similarity or Jaccard coefficient among vectors.

With the development of the Internet, how to acquire more accurate information from massive amounts of text data is a challenge to the approach (e.g., TF-IDF) of ignoring the terms semantics. We should analyze, capture, and characterize the meaning of the text more precisely rather than only term occurrence frequency. For example, there is an article about gift (present) and another article about gift (talents). The two articles will be regarded as similar things, if the articles are measured based on term frequency method. On the other side, an article about girl and another article about boy are regarded as dissimilar papers for their different term (boy and girl). Therefore, term similarity is researched gradually. The similarity measurement in terms requires organizing all words to form a semantic network (e.g., WordNet), and it is realized by determining information of edges and vertexes in terms.

Literature [7] described an approach for domain-specific WSD by selecting the predominant sense (sunset from WordNet) of ambiguous words. To achieve it the method uses two corpora: the domain-specific test corpus and a domain-specific auxiliary corpus. Literature [8] put forward method considering vertex information and edge relationship, which is helpful to similarity application of noun or verb. However, it is difficult to organize hierarchical relationships like nouns for adjective or adverb. Literature [9] discussed local correlation information to determine similarity of texts by WordNet. Literature [10] defined similarity among terms by applying information theory on the premise of text vocabulary in specified probability distribution. Literature [11] put forward semantic similarity measurement method to improve the traditional term frequency-based text similarity measure result, but the method does not reduce the dimension of the text model. Literature [12] discussed a method of determining sentence similarity, and text automatic summary is applied in the method. Literature [13] recalculated text correlation of results returned by the search engine by ontology. Literature [14] introduced term similarity measurement method combined with WordNet and applied it to improve the vector representation model of the text. By analyzing text concept, synonym, and term hyponymy relation, it improved more extensive frequency vector including text concept, synonym, and term hyponymy relation and realized text clustering by computing the cosine similarity among the vectors. Literature [15] designed a supply chain information oriented mining model based on TF-IDF algorithm to obtain the required supply chain information. Literature [16] put forward a new method integrating the advantages of TF-IDF and semantic information from HowNet, and the method worked out the value of text similarity by hamming distance to avoid direct processing of high-dimensional sparse matrix. Literature [17] proposed the scientific research project TF-IDF (SRP-TF-IDF) model, which combined TF-IDF with a weight balance algorithm designed to recalculate candidate keywords. Literature [18] improved Bayes algorithm with TF-IDF method, and it introduced decentralized word frequency factor and feature word position factor to enhance the accuracy of feature weights. Literature [19] proposed a method based on the combination of contents and their semantic similarities, and the method is a collection of synonyms and inverse document frequency combining semantic similarity by WordNet synonyms set.

These methods did not reduce text representing vector dimension, and its calculating method of text similarity is also traditional cosine similarity between vectors. By analyzing the above methods about the text similarity, the paper firstly preprocesses the text in natural language processing techniques. After that, key terms with high TF-IDF value in text are searched in terms of TF-IDF method. Besides, the similarity of two texts is calculated by external dictionary terms analysis, term similarity weighting tree structure, and text semantic definition. The method in the paper can make text similarity measures more efficient and accurate, and it also decreases the dimension of the text similarity model. By text clustering experiments with the benchmark data set, the algorithm discussed in the paper is better than the pure TF-IDF and the method of semantic understanding at the aspect of accuracy, recall, and F1-metric by different K-means clustering methods.

3. Text Similarity Measurement Hybrid Algorithm with Term Semantic Information and TF-IDF Method

3.1. Text Preprocessing. Current natural language processing techniques cannot deal with full original information of text easily. The text data is usually unstructured or semi-structured; then machine cannot handle it directly. Therefore, it is very necessary to properly preprocess the text first and then to establish the frequency vector of the terms in the text to finally transform the text into a structured form.

3.1.1. Text Segmentation. The first key point of text preprocess is text segmentation, root reduction, and stop words deletion. English contains different tenses, and words are also divided into single and plural forms, and then most words also appear in different forms. If the words based on the same root appear in different entries as different forms, then it is possible that the text with the same theme has a very low similarity, which directly affects the quality of the text clustering, so the root reduction processing is required. Stop words are words that have little significance to identify text content but appear very frequently, and they will lead to large errors in calculating text similarity or in training the model to obtain parameters, and they are usually regarded as a noise. For example, definite articles “a” and “an” will appear in almost any text, but there is little substantial contribution to the expression of the textual meaning. Therefore, it is very necessary to remove these stop words from the original text, and the process is called deleting stop words. The deletion of stop words is achieved by establishing a list of stop words. The list of stop words is a query process to delete stop words. By querying each item by item, then all terms in the list are deleted. Text preprocessing typical example is shown in Figure 1.

3.1.2. Special Word Deletion. The method in the paper requires semantic analysis of term, and then the following three preprocessing steps are necessary based on stop word deletion.

(1) Special terms (such as people’s name, place name, organize name, etc.) in the text need to be processed. These special terms always have high TF-IDF value in TF-IDF computing process, and they are incorrectly selected as text key term. In addition, this special word term makes a great impact on similarity result. In the paper, this special word term is processed by name entity recognition technology [20], and the special word items recognized are replaced with specific string. In order to avoid the possible adverse effects of these word terms on text clustering during feature selection, the special word terms are ignored in selecting feature terms.

(2) Synonyms may appear simultaneously in a document, and then synonyms appearing in the text should be treated consistently. In other words, the same meaning word terms should be combined, and they are represented by a single name to reduce costs on calculating the semantic similarity of texts.

(3) Since the most important thing about characterizing the meaning of the text is substantive in the text, the final step is to perform a verbal analysis of all the terms in the text. The semantic properties of all terms should be judged to distinguish nouns, verbs, adjectives, and adverbs, etc.

3.1.3. Special Word Deletion

(1) Text Preprocessing Process. By comprehensively considering comprehensive word segmentation, root reduction, stop words deletion, and special term filtering technology, text preprocessing process is shown in Figure 2.

3.2. Key Terms Selection. After the text preprocessing is completed, the TF-IDF values of terms in one text should be calculated, and each term TF-IDF value in text is represented as a vector to support texts similarity computing. The text vector is high-dimensional and extremely sparse. According to information theory, the value of IDF is cross entropy of term probability distribution in special condition, and TF is used to increase the weight of words to describe the information features of words in text. Thus, several important words from each text can be selected to represent the text. This can reduce the text feature vector representation without affecting text feature extraction. The approach in detail is shown below:

(1) All terms in the text are sorted according to their TF-IDF value.

(2) Nouns and verb terms are selected as key word terms, if their TF-IDF value is greater than p (p is the percentage). Besides, the selected key term is add to the vector.

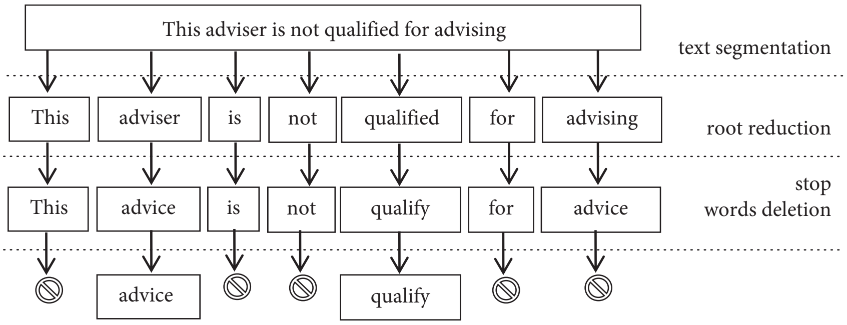

Figure 1: Text preprocessing typical example.

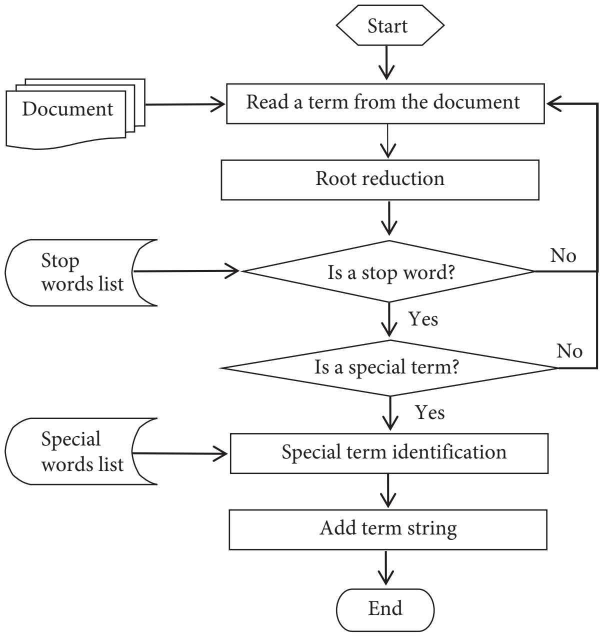

Figure 2: The process of text preprocessing.

(3) The last key term vector is regarded as feature representation of the text. Compared with traditional TF-IDF method, the key term vector dimension decreases by 1-p, and it is a large increase in efficiency.

3.3. Text Similarity Calculation. After the eigenvector of each text is determined, the following problem to resolve is how to calculate similarity of two texts. The most important information in an article is the characteristic word term, and the text similarity measurement can be translated into similarity calculation between the vectors of the feature terms. As a result, the similarity between the original texts can be regard as the similarity between vectors of feature word terms. To ensure that the vector of the similarity meets the basic similarity measure, the dimension influence must be removed.

If Sim (x, y) is similarity of data points x and y, the following conditions should be satisfied.

If and only if $x = y$, $Sim\ (x, y) = 1$ ($Sim\ (x, y) \in [0, 1]$).

$\forall x$ and y, $\mathrm{Sim}(x, y) = \mathrm{Sim}(y, x)$.

Let $\mathbf{v}_i$ and $\mathbf{v}_j$ represent key term vector, where $\mathbf{v}_i = (w_{i1}, w_{i2}, \ldots, w_{ik}, \ldots, w_{im})$, $\mathbf{v}_j = (w_{j1}, w_{j2}, \ldots, w_{jk}, \ldots, w_{jm})$, $\mathbf{v}_j = (w_{j1}, w_{j2}, \ldots, w_{jk}, \ldots, w_{jm})$, and the similarity of two texts is defined as below:

$$\mathrm{textSim}\left(v_i, v_j\right) = k_w \cdot \mathrm{vectSim}\left(v_i, v_j\right), \tag{4}$$

where k_w is weight coefficient of key term vectors $\mathbf{v}_i$ and $\mathbf{v}_j$. Similar terms can determine the TF-IDF value in the document. The more the similar terms, the higher the TF-IDF value. It indicates that these terms can reflect their importance better in the text. Thus, weighting is determined by the proportion of the TF-IDF values of the keyword terms in the sum of the whole text TF-IDF values in the keyword vector. Weight coefficient k_w is calculated by equations (5) and (6).

$$k_w = 1 + \mathrm{ave}(i, j) \cdot \left(\sqrt{\mathrm{vectSim}\left(v_i, v_j\right)} - \mathrm{vectSim}\left(v_i, v_j\right)\right), \tag{5}$$

$$\mathrm{ave}(i, j) = \frac{1}{2}\left(\frac{\sum_{k\in\Lambda_i} \mathrm{TFIDF}(w_{ik})}{\sum_{k=1}^{m} \mathrm{TFIDF}(w_{ik})} + \frac{\sum_{l\in\Lambda j} \mathrm{TFIDF}\left(w_{jl}\right)}{\sum_{n=1}^{n} \mathrm{TFIDF}\left(w_{jl}\right)}\right), \tag{6}$$

where TFIDF (w_{ik}) is TF-IDF value of key term w_{ik}, and ave (i, j) represents the proportion of the TF-IDF values of the keyword terms in the sum of the whole text TF-IDF values in the keyword vector. Sets Λ_i and Λ_j are defined as below.

$$\begin{gathered}\Lambda_i = \left\{k:\ 1 \le k \le m,\ \max_{1\le s\le n}\left\{\mathrm{Sim}\left(w_{\mathrm{ik}}, w_{\mathrm{js}}\right)\right\} \ge \theta\right\}, \\ \Lambda_j = \left\{l,:\ 1 \le l \le m,\ \max_{1\le s\le m}\left\{\mathrm{Sim}\left(w_{jl}, w_{is}\right)\right\} \ge \theta\right\}.\end{gathered} \tag{7}$$

In key term vector $\mathbf{v}_i$, keyword w_{ik} is put into set Λ_i, if similarity of key terms w_{ik} and w_{js} exceeds setting threshold value. Sim (w_{ik}, w_{js}) is semantic similarity of keywords w_{ik} and w_{js}.

$$\begin{aligned}\mathrm{vectSim}\left(v_i, v_j\right) = \frac{1}{2}\Bigg(&\frac{1}{m}\sum_{k=1}^{m} \max_{1\le s\le n}\left\{\mathrm{Sim}\left(w_{ik}, w_{js}\right)\right\} \\ &+ \frac{1}{n}\sum_{k=1}^{n} \max_{1\le s\le n}\left\{\mathrm{Sim}\left(w_{is}, w_{jk}\right)\right\}\Bigg),\end{aligned} \tag{8}$$

where vectSim $(\mathbf{v}_i, \mathbf{v}_j)$ is determined by term similarity of vectors $\mathbf{v}_i$ and $\mathbf{v}_j$, and terms with high similarity degree must appear in similar vector, and the vectors including low similarity degree are obviously dissimilar. The weighting coefficient is calculated according to term similarity weighing hierarchical tree data structure and sememe similarity formula. There are leaf nodes and nonleaf nodes in three-layer weighting tree, and all terms with similarity exceeding threshold value θ are sorted in order from large to small, and they are saved in leaf nodes. Figure 3 is construction process of weighting tree.

(1) The initialization of TSWT

The three-layer similarity weighting tree of feature terms is constructed by user concrete task, and the feature term is put into each leaf node, whose similarity is greater than special threshold value.

(2) The weight and update of TSWT

In the calculation of the eigenvector similarity process, the similarity result of eigenvectors $\mathbf{v}_i$ and $\mathbf{v}_j$ is disposed, if a certain pair of feature terms (w_{ik} and w_{js}) satisfies following one condition.

(a) w_{ik} and w_{js} belong to ordered queue of terms for a certain leaf node in a weighted tree.

(b) If w_{ik} belongs to ordered queue of terms for a certain leaf node, w_{js} distinguishes foreign from the queue, and there is a high similarity above the threshold k_w. According to similarity of w_{js} and other terms, the sequence location of w_{js} in ordered queue including w_{ik} is determined.

(c) If w_{ik} and w_{js} do not belong to ordered queue of terms for a certain leaf node in a weighted tree, there is maximum and minimum similarity value with w_{ik} and w_{js}. If the similarity degree is less than threshold k_w, a branch with terms of maximum and minimum similarity should be constructed, and w_{ik} and w_{js} are inserted into the new branch.

(d) If w_{ik} and w_{js} do not belong to ordered queue of terms for a certain leaf node in a weighted tree, there is maximum and minimum similarity value with w_{ik} and w_{js}. If the similarity degree is less than threshold k_w and exceeds threshold k_w, the sequence location of w_{ik} and w_{js} is determined by the similarity of other terms with w_{ik} and w_{js}.

(3) Text similarity calculation

Similarity of two key term vectors is calculated by equation (4) and TSWT.

Text similarity measurement hybrid algorithm with term semantic information and TF-IDF method is shown as below.

Input: Feature term vector $\mathbf{v}_i$ and $\mathbf{v}_j$; term similarity weighting tree; the threshold value k_w.

Output: Similarity Sim $(\mathbf{v}_i, \mathbf{v}_j)$ of key terms $\mathbf{v}_i$ and $\mathbf{v}_j$.

Step 1. The term similarity weighting tree is initialized.

Step 2. Starting from w_{il} in the vector $\mathbf{v}_i$, the most similar term w_{jk} to w_{il} in the vector $\mathbf{v}_j$ is searched by sememe similarity equation, and the similarity of w_{il} and w_{jk} is recorded.

Step 3. The weighting coefficient k_w is calculated by TSWT weighting principle, and determine whether w_{il} and w_{jk} are added to weighting tree according to TSWT updating principle.

Step 4. Repeat the procedure of Steps 2 and 3 for other terms in vector $\mathbf{v}_i$ until all terms in vector $\mathbf{v}_i$ find the corresponding most similar term in vector $\mathbf{v}_j$.

Step 5. The similarity value of Step 2, Step 3, and Step 4 is accumulated, and the result divided by the number of all terms in vector $\mathbf{v}_i$ is the dimension of vector $\mathbf{v}_i$; thus the similarity of vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ Sim $(\mathbf{v}_i, \mathbf{v}_j)$ is determined.

Step 6. Start from w_{jl} in the vector $\mathbf{v}_j$, and repeat from Step 2 to Step 5. Thus, the similarity of vectors $\mathbf{v}_j$ and $\mathbf{v}_i$ Sim $(\mathbf{v}_j, \mathbf{v}_i)$ is determined. The goal of this step is to keep the vector $\mathbf{v}_i$ dimension the same as $\mathbf{v}_j$.

Step 7. The average of Sim $(\mathbf{v}_i, \mathbf{v}_j)$ and Sim $(\mathbf{v}_j, \mathbf{v}_i)$ is calculated, and vectSim $(\mathbf{v}_i, \mathbf{v}_j)$ is regarded as semantic similarity of vectors $\mathbf{v}_i$ and $\mathbf{v}_j$.

Step 8. According to above steps cumulation, the sum weighting coefficient ω_f is determined.

Step 9. The similarity of vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ is processed in weight by text similarity definition, and text similarity of vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ is determined.

4. Case Study

In order to verify the effectiveness of the hybrid algorithm in solving text similarity measurement problem, this paper collected 500 article papers of HowNet as data set, and text set involves multiple fields, including computer, economy, organism, physics, and mechanics, etc. The total number of five-class text is, respectively, 131 (computer), 117 (economy), 113 (organism), 91 (physics), and 73 (mechanics). The feature of each data set is shown as Figure 4.

The above text set is preprocessed firstly by natural processing language software *LinPipe* of *Alias* company. Segment and word class tagging of each text is realized by *LinPipe*, and then the relevant person names, place names, and organization names involved in the text collection are identified. The weight of terms in text is calculated by TF-IDF algorithm, and specific percentages top value is selected from computing result. The similarity of the experimental text is calculated by the hybrid method in the paper, and text

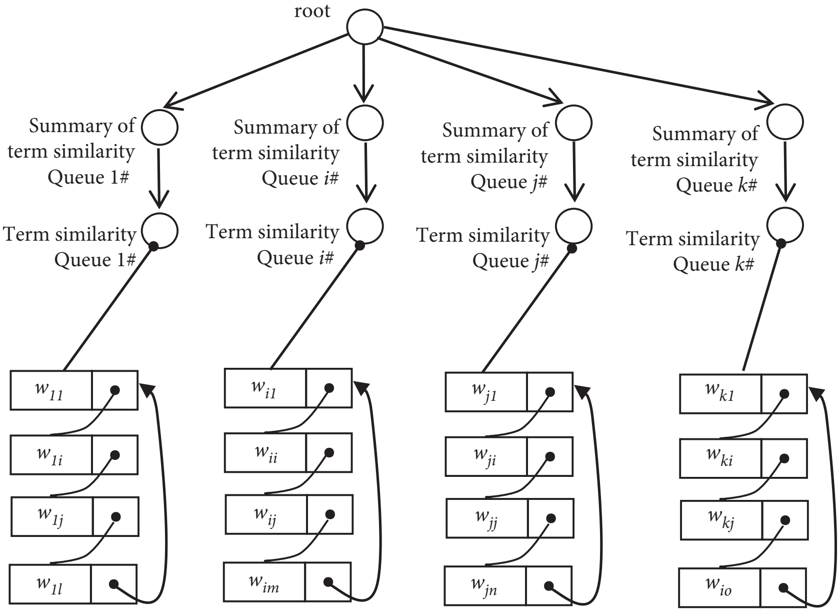

Figure 3: Term similarity weighing tree. Term similarity weighting tree (TSWT) can be constructed automatically according to flowing method.

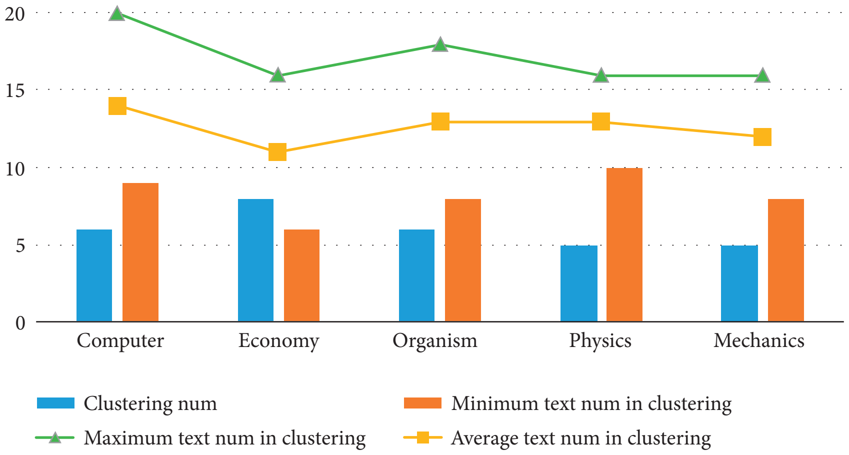

Figure 4: The feature statics and analysis of data set.

similarity matrix is determined. According to text similarity matrix and TF-IDF matrix, clustering experiment is realized, and the results of direct K-means algorithm (DKM), binary K algorithm (BKM), aggregation K-means algorithm (AKM), and hybrid algorithm of the paper are analyzed and compared. In order to make the experimental results more objective, this paper measures text similarity by multiple indexes, and indexes include accuracy, recall ratio, F1-metric, and macroaveraging.

It is necessary to select different percentage of top characteristic terms in the similarity calculation to understand how top characteristic terms impact similarity calculation. From this, characteristic terms similarity threshold value k_w is set as zero to ensure that all characteristic terms are equally important. Figures 5–7 describe influence of various percentage top characteristic terms on similarity result. If percentage top characteristic terms are located in the interval of 30% and 50%, the accuracy of computer, economy, organism, physics, and mechanics is the highest, and it is about 6 percentage points higher than other top characteristic terms' percentage. For recall ratio, there is also best interval in [0.4, 0.5]. The value of F1-metric is inflection point at the 40 percentage top characteristic terms. According to the above statistical analysis, the text term clustering result is the best, when percentage TOP characteristic terms are selected as about 40%.

In order to determine influence of threshold value k_w for similarity computing, the experiment selects 40% top characteristic terms as text feature vector, and DKM is also selected as clustering algorithm. k_w distributes in 0.6 and 0.9, and Figures 8–10 describe influence of threshold value on similarity result.

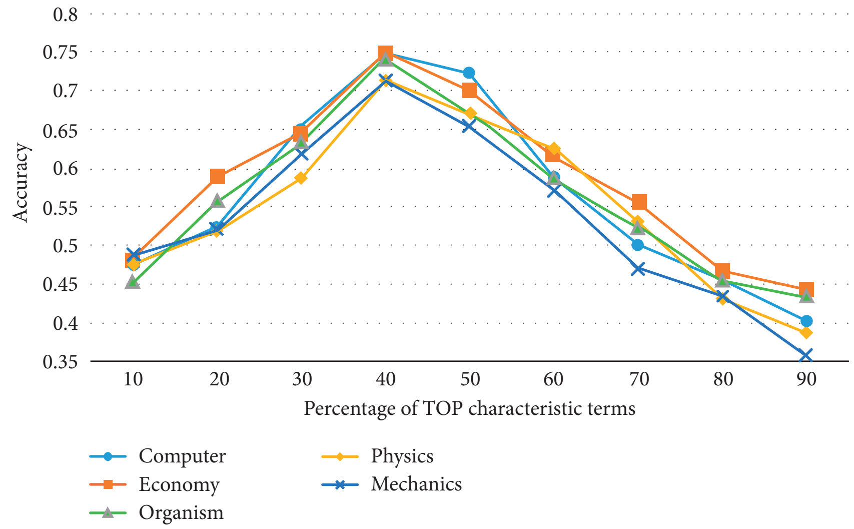

Figure 5: Accuracy comparison of various percentage top characteristic terms.

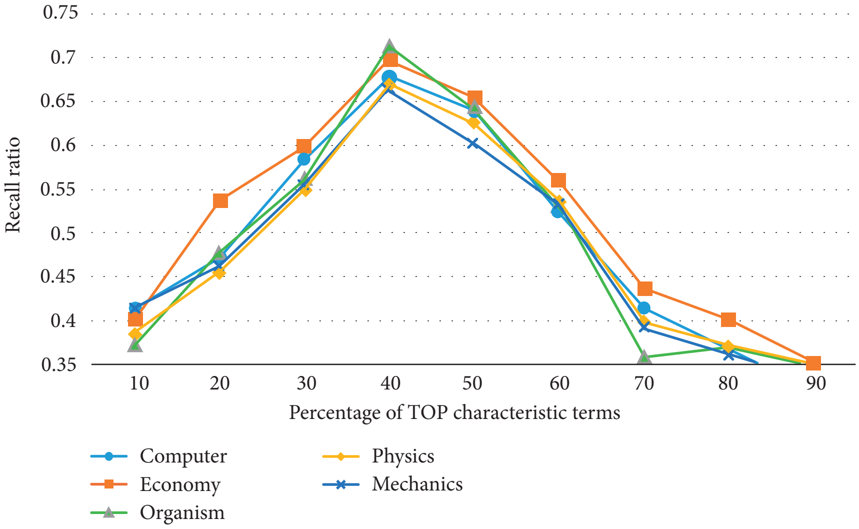

Figure 6: Recall ratio comparison of various percentage top characteristic terms.

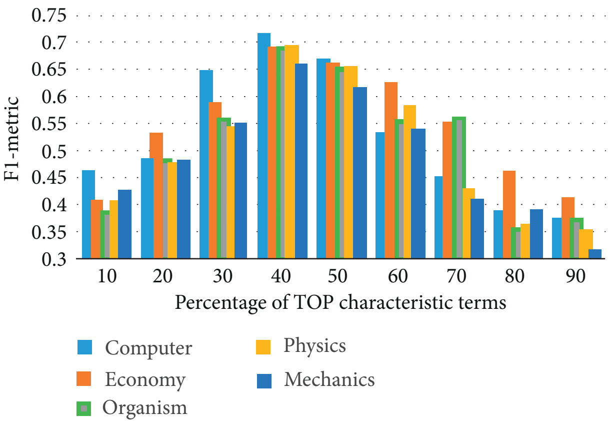

Figure 7: F1-metric comparison of various percentage top characteristic terms.

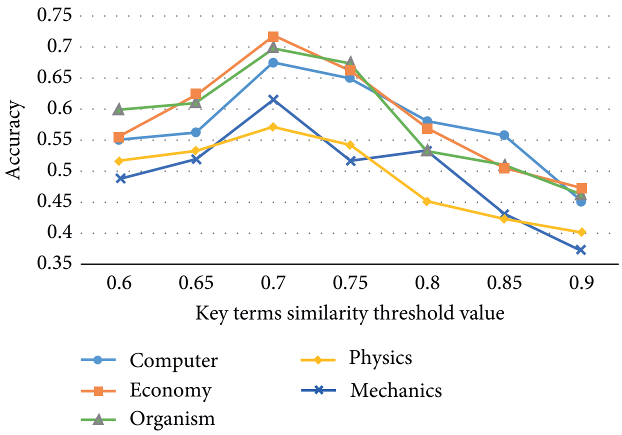

Figure 8: Accuracy comparison of various key similarity threshold value.

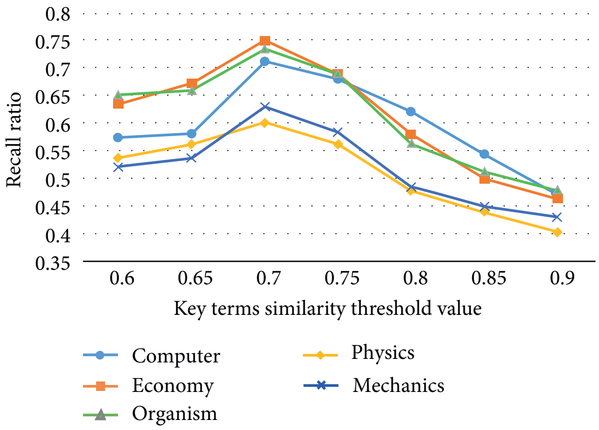

Figure 9: Recall ratio comparison of various key similarity threshold value.

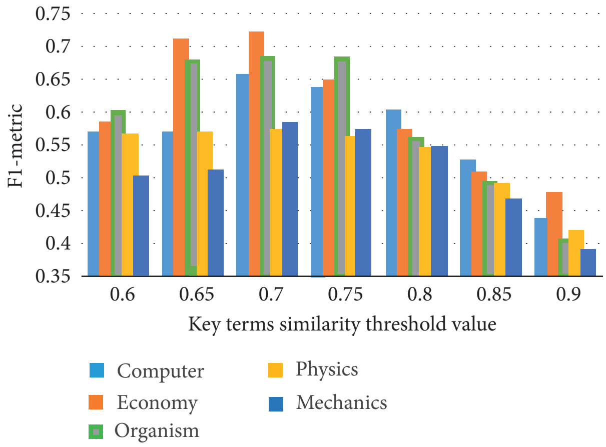

Figure 10: F1-metric comparison of various key similarity threshold value.

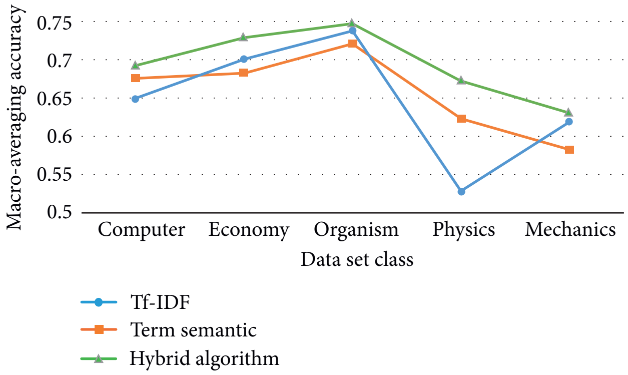

Figure 11: Macroaverage accuracy analysis of three methods in DKM algorithm.

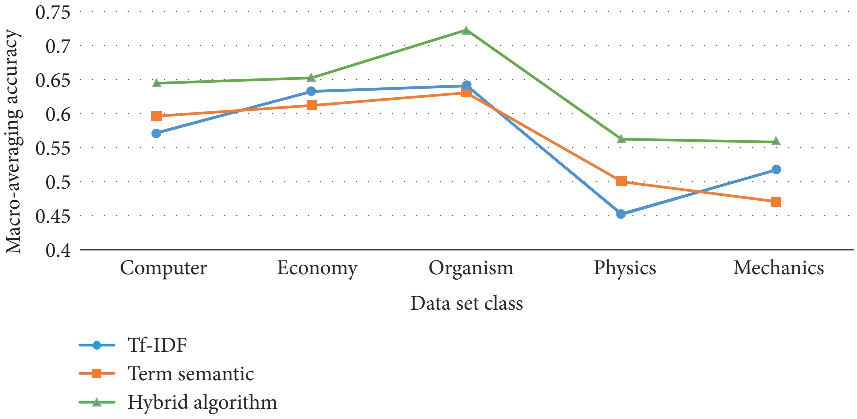

Figure 12: Macroaverage recall ratio analysis of three methods in DKM algorithm.

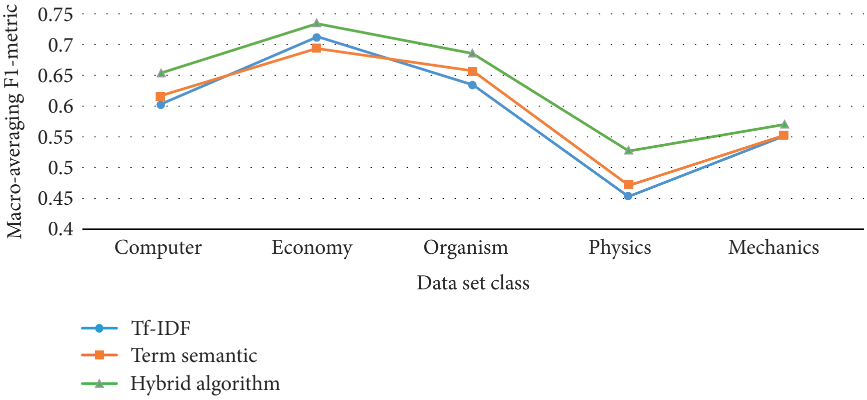

Figure 13: Macroaverage F1-metric analysis of three methods in DKM algorithm.

In conclusion, it can be seen that the clustering effect also increases gradually as the similarity threshold rises gradually. This is mainly due to the gradual rise of the similarity threshold, the discrimination between texts becomes larger and larger, and the clustering effect is naturally getting better and better. In particular, the best clustering effect was achieved at the interval of 0.7 and 0.75, and if the threshold value is improved unceasingly, the clustering effect began to decline gradually.

In terms of above statistical analysis, the initial parameter is set as below. The threshold value $k_w = 0.7$, and top characteristic terms percentage is 40%. The hybrid algorithm in the paper is compared with traditional TF-IDF method and term semantic method. Furthermore, clustering method is also realized by algorithm of DKM, AKM, and BKM. The experiment result is evaluation by multiple indexes, including accuracy, recall ratio, and F1-metric. Macroaverage is a comprehensive index, which considers concurrently accuracy, recall ratio, and F1-metric. Macroaverage assigns the same weight to each category, and it is calculated as the following formula to prove the method validity.

$$\begin{aligned} \text{MacroAVG} - \text{accuracy} &= \frac{\sum_{i=1}^{|c|} \text{accuracy}_i}{|C|}, \\ \text{MacroAVG} - \text{recallRatio} &= \frac{\sum_{i=1}^{|c|} r\text{ecallRatio}_i}{|C|}, \\ \text{MacroAVG} - \text{F1 Metric} &= \frac{\sum_{i=1}^{|c|} \text{F1 Metric}_i}{|C|}. \end{aligned} \tag{9}$$

The three-method text similarity measurement in DKM clustering algorithm is shown as Figures 11–13. For macroaveraging accuracy, the hybrid algorithm of the paper is optimal, and t is two percentage points higher than TF-IDF and term semantic method in algorithm accuracy for the hybrid algorithm. The hybrid algorithm put forward in paper also has the same advantages in both macroaveraging recall ratio and F1-metric. The results of three-method text similarity in AKM and BKM are analyzed, and the conclusion is the same as DKM experimental result. This shows that the method used in this paper has a better clustering effect than the two traditional algorithms and effectively avoids the disadvantages of the traditional methods to some extent and confirms the validity of the method used in this paper.

5. Conclusion

The terms with high TF-IDF value are selected as feature keywords in hybrid algorithm, and the method reduces the impact of the high dimensions of the traditional vector representation. Besides, it also decreases computing time. It fully combines the similarity of the feature keywords semantics in the text with external dictionary word analysis to realize semantic similarity degree computing between two texts by terms similarity weighting tree structure. Based on TF-IDF model, at the same time keywords in the text analysis of semantic information, a new method of text similarity measure is discussed. The probability distribution of the terms in the text is fully discussed, and experiment result shows that the clustering method in the paper is better than traditional method, such as TF-IDF or semantic method at the aspect of accuracy, recall rate, and F1-metric. The work of this paper has some improved effects on the traditional two types of text similarity measures; however there are still many shortcomings to be overcome. The cosine angle problem is not fully considered when calculating the cosine similarity of texts, and there are many works to mine semantic characteristics contained in the analysis of text similarity, such as semantic information of statements, paragraphs, and chapters in the text.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The author declares no competing interests.

Acknowledgments

This study was sponsored by the special general projects for technological innovation and application development of Chongqing Science and Technology Commission of China (no. cstc2019 jscx-msxm0060).

References

[1] H. Zhang, M. Huang, and W. Li, "Emotional change detection oriented speech emotion database," *Computer Simulation*, vol. 38, no. 9, pp. 448–455, 2021.

[2] C. Li, L. Zhao, X. Li, and L. Wang, "Text sentiment classification model based on TF-IDF weighted convolutional neural network," *Journal of Chongqing University of Technology (Natural Science)*, vol. 35, no. 11, pp. 109–115, 2021.

[3] S. Lee and J. Nang, "A near-duplicate image detection system for design contents using SIFT," *KIISE Transactions on Computing Practices*, vol. 25, no. 5, pp. 257–262, 2019.

[4] Z. Xiao and G. Feng, "Text similarity calculation based on 'HowNet' original space," *Computer Science and Engineering*, vol. 13, no. 29, pp. 8651–8656, 2013.

[5] K. Liao and B. Yang, "Research on text similarity calculation based on weighted semantic network," *Journal of Information*, vol. 31, no. 7, pp. 186–182, 2012.

[6] L. Zhang, Y. Jiang, and L. Sun, "An improved TF-IDF text clustering method," *Journal of Jilin University (Science Edition)*, vol. 59, no. 5, pp. 1200–1204, 2021.

[7] I. Lopez-Arevalo, V. J. Sosa-Sosa, F. Rojas-Lopez, and E. Tello-Leal, "Improving selection of synsets from WordNet for domain-specific word sense disambiguation," *Computer Speech & Language*, vol. 41, pp. 128–145, 2017.

[8] J. J. Jiang and D. W. Conrath, *Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy*, p. 11512, ROCLING, 1997.

[9] D. Ramage, A. N. Rafferty, and C. D. Manning, "Random walks for text semantic similarity," in *Proceedings of the Workshop on Graph-based Methods for Natural Language Processing DBLP*, Singapore, August 2009.

[10] X. Ran, X. Zhou, M. Lei, W. Tepsan, and W. Deng, "A novel k-means clustering algorithm with a noise algorithm for capturing urban hotspots," *Applied Sciences*, vol. 11, no. 23, Article ID 11202, 2021.

[11] W. Yong and J. E. Hodges, "Document clustering with semantic analysis," in *Proceedings of the Hawaii International Conference on System Sciences IEEE Computer Society*, Kauia, HI, USA, January 2006.

[12] M. A. Ramiz, "A new sentence similarity measure and sentence based extractive technique f or automatic text summarization," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7764–7772, 2009.

[13] L. Gang, Z. Cheng, and Z. Li, "Text information retrieval based on concept semantic similarity," in *Proceedings of the 2009 Fifth International Conference on Semantics, Knowledge and Grid*, IEEE, Zhuhai, China, October 2009.

[14] M. AlMousa, R. Benlamri, and R. Khoury, "Exploiting non-taxonomic relations for measuring semantic similarity and relatedness in WordNet," *Knowledge-Based Systems*, vol. 212, Article ID 106565, 2021.

[15] J. Lu, Z. Fan, and C. Liu, "Supply chain information oriented mining model based on TF IDF algorithm," *Computer Simulation*, vol. 38, no. 7, pp. 153–156, 2021.

[16] W. Hu, B. Qian, and K. Li, "The research about text similarity measuring through Hamming-distance and semantics," *Journal of Hangzhou Dianzi University (Natural Sciences)*, vol. 36, no. 3, pp. 36–41, 2016.

[17] Z. Wang, D. Wang, and Q. Li, "Keyword extraction from scientific research projects based on SRP-TF-IDF," *Chinese Journal of Electronics*, vol. 30, no. 4, pp. 652–657, 2021.

[18] T. Xu and M. Wu, "An improved Naive Bayes algorithm based on TF-IDF," *Computer Technology and Development*, vol. 30, no. 2, pp. 75–79, 2020.

[19] Y. Zhou and M. Dai, "News recommendation technology combining semantic analysis with TF-IDF method," *Computer Science*, vol. 40, no. S2, pp. 267–269, 2013.

[20] S. Yildirim and T. Yildiz, "A comparative analysis of text classification for Turkish language," *Pamukkale University Journal of Engineering Sciences*, vol. 24, no. 5, pp. 879–886, 2018.