J. COMBIN. MATH. COMBIN. COMPUT. 124 (2025) 377–387

Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on thematic clustering and text mining of chinese modern and contemporary literary texts in the network era

Yiling Sun[1,✉]

[1] *Faculty of Art and Social Science, National University of Singapore, 119077, Singapore*

ABSTRACT

This paper aims at resolving the issue that the conventional literature study can't deal with the large amount of data, the author proposes a research method for theme clustering and text mining of Chinese modern and contemporary literary texts in the network era. The author studied how to effectively improve the thematic clustering performance of literary texts based on keyword clustering ensemble method. Comparing two clustering ensemble methods (K-means based data ensemble and incremental clustering based algorithm ensemble) and four keyword extraction methods (TF-ISF CSI, ECC, TextRank), the effects of various keywords on the results of thematic clustering were analysed. Experiments indicate that the clustering algorithm can greatly increase the topic clustering efficiency, and it is more stable when the key words are less. The author's research provides new technological means for text mining and thematic clustering in contemporary Chinese literature, which helps to promote the development of digital humanities research.

Keywords: internet era, text mining, cluster analysis

1. Introduction

Network literature relies on the development of the Internet, with the development of the network, the network of modern and contemporary literature dissemination speed as well as in the life of the reading occupied by the proportion of increasing, the impact on the reading population is also increasing day by day, the network of modern and contemporary literature has long become an indispensable part of people's lives [14, 5]. Early network modern and contemporary literature works are mostly related to reality, such as emotional life class. However, with the depletion of realistic creative resources and the market's pursuit of fantasy themes, online contemporary literature has

✉ Corresponding author.
E-mail address: sunyiling2024@163.com (Y. Sun).

Received 10 September 2024; Accepted 22 January 2025; Published Online 18 March 2025.

DOI: 10.61091/jcmcc124-25

shown a tendency to turn its back on reality [7, 15].

With the creation of most of the network works with non-realistic themes, the phenomenon of fish-eye mixed pearls has appeared in the network present and contemporary literature, which makes it tricky for readers to select reading works, and there is no better method in selecting the network present and contemporary literature, and at the same time, there are fewer text mining researches conducted for the network present and contemporary literature. In this paper, it is necessary to combine the theme mining content to show the main content and ideas of the works more intuitively [9, 11, 6]. This method can be more applied in the preliminary screening and presentation process of future literary works, which has certain theoretical and practical significance, provides convenience for readers' reading, and promotes the further development of network modern and contemporary literature in the current new era [4, 10, 8].

The rapid development of network applications makes the number of texts increasing. Text topic clustering is an unsupervised machine learning method, which can automatically complete the tasks of effective organization of text information, automatic categorization, topic discovery and so on by providing a meaningful classification, helping users to quickly obtain useful information under a specific topic from a huge amount of text [13, 12].

Text similarity clustering is an important method of text topic clustering, is in the case of the document set without category labeling, based on specific criteria will be divided into a number of clusters, each cluster represents a topic, text characterization is the key to text similarity clustering [1, 2]. Effective analysis of large text is a challenging problem, long text due to more diverse semantics, the text contains the theme is not unique and there are redundancy and noise and other problems, increasing the difficulty of clustering [3].

The traditional vector space model extracts text features by One-Hot, TF-IDF and other methods, but it does not consider word order and lacks semantic features [17]. Semantically related textual representations can be obtained by using neural network language models, which are mainly based on word embedding approach and pre-trained language model (PTM) approach. The PTM represented by BERT adopts a deep bidirectional Transformer structure with strong feature extraction, but BERT requires that the length of the input sequence is no more than 512 characters, so it cannot directly obtain semantic representations of long documents.Doc2Vec model can be applied to phrases, sentences, large documents, and other text fragments of any length, in sentiment analysis, text classification, topic clustering, personalized recommendation and other NLP tasks have been effectively applied [16].

2. Research methods

2.1. Main techniques of text mining

Text mining is a large framework with many branches, and text clustering is one of its branches. Text data is complex unstructured data that cannot be directly recognized and calculated by computers, and cannot be implemented using algorithms. So, in order to achieve text data mining, it is necessary to transform complex unstructured forms into structured forms, so that computers can recognize and calculate them. The expression of text is also the essence of text mining. A crucial step in the transformation process of text mining is text preprocessing, which ensures high accuracy in text clustering. When representing text in a computable structured form, it is important to note that the transformed form must fully retain the information of the original text and be able to clearly

distinguish the text. Preprocessing has two steps, namely word segmentation and stopping words. Figure 1 shows the mining process using Chinese text as an example.

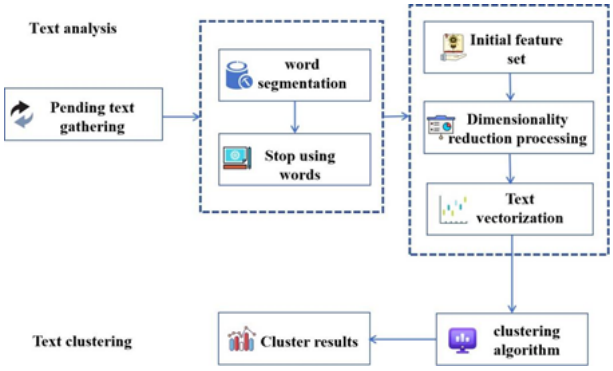


Fig. 1. Process of text mining

The text analysis section describes the process of analyzing Chinese text, while the analysis of English text differs from that of Chinese text in the preprocessing process due to tense and voice issues in English.

2.1.1. Text representation. The conversion process mentioned earlier is the representation of text and the expression of text features. Reduce a text to a vector representing its significance, which is a characteristic vector made up of many characteristic words. The calculation of characteristic vector is a kind of text expression. Text is a kind of natural language made up of many words, such as words, punctuation, digits and so on. It is a complex unstructured form that cannot be recognized and processed by existing computers, and cannot be directly implemented for clustering analysis using algorithms. The bag of words model is a typical conversion method because words are the smallest semantic expressions. There are many ways to convert this, including various text representation methods. There are many representation models mentioned in academic papers, and the commonly used one is the Boolean logic model (Boolean), which uses 0 and 1 to determine weights. This model is simple and easy to implement. There is also the vector space model VSM, which is a commonly used model in text mining and has better representation performance than other models. In addition, there are probability models, etc.

2.1.2. Text preprocessing. Most of the raw files are written in natural language and are made up of lots of words, punctuation, digits, etc., and there are no obvious distinguishing marks between words. It is unstructured data that must be transformed into form first. A document is composed of a large number of words and phrases, so words are the smallest components that can reflect the content of the text, constructing spatial vectors based on words. So to cluster text, it is necessary to first break down a large number of documents into individual words or phrases, so that the original continuous articles can be broken down into independent strings, retain useful words, delete words and numerical symbols that have no actual meaning, and collectively refer to these meaningless words as stop words. The reserved words form the initial feature item, which is the set of feature words.

The first step in preprocessing is word segmentation, which breaks down the text into independent words and restores it to the initial form of the document. A complete text is formed by combining words, numbers, punctuation, and other strings. Due to the limited information that a single word can convey and the fact that documents are composed of a large number of words, some of which

have multiple meanings, using words as a criterion for feature selection can lead to a very high vector dimension, resulting in a difficult problem of dimensionality disaster in data processing. Although phrases can convey more information than words, some phrases may contain multiple characters and appear infrequently in documents, making it difficult to represent the text effectively. Therefore, in the process of word segmentation, choose words as feature items. Word segmentation means separating words from each other. During the segmentation process, there are differences between Chinese and English. English has a rich voice, while Chinese does not have such a rich voice. Apart from the diversity of meanings, there is no change in form. Characters are separated by punctuation marks, without any separators. Punctuation marks can be filtered out as stop words after word segmentation. These features can help the document automatically segment during word segmentation.

Stop is a term that is removed from a document preprocess, that is, one that does not make a significant contribution to analyzing the main message. Pause is one of the most common expressions in IR, and Hans Peter Luhn is one of the founders of IR. Stop is a manual input or definition according to the analysis, instead of automatic generation. A term that is defined as a "stop" will produce a list of stop words. It is important to note that there are no explicit or general rules for the use of all types of data searches, and each term is regarded as a "stop". Through the filtering of stop words, it is possible to eliminate the limitation of dimension, decrease the noise, and increase the efficiency of text mining. There are three types of stop words. The first category is the function of the language, which is used for emotional expression or for the fluency of sentences. The functions of these terms are quite ordinary, without any practical significance. Such as prepositions, conjuncts, "Yes", "Da" and "Ma". The other is the special words, which are useful but occur more than once in the analysis of a particular text. It is not useful for conveying messages, nor does it serve as a basis for differentiating the text. Thus, it is regarded as a stop word filtering. The third kind are punctuation and figures. This kind of language has no meaning in analyzing the text and is used as a stop. The stop word list can be subjectively created based on analysis needs, and can also be searched on Baidu. There are also built-in stop words in the TM package of R language.

Unlike Chinese, English has rich tense and voice issues. The same root can derive many different words by adding prefixes and suffixes, but the meaning remains unchanged. For example, "intersection," "section," and "dissect" have the same root word "sect," but words in different tenses express the same meaning. Therefore, when doing document preprocessing, Stemming is needed to convert the segmented feature words into the same voice and root word. The specific process is shown in Figure 2.

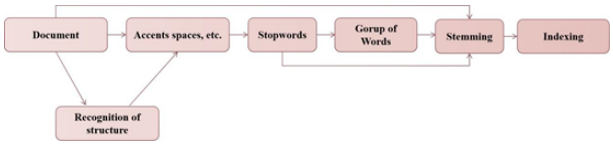


Fig. 2. Process of text mining

2.2. Text clustering

Clustering is an unsupervised machine learning algorithm in statistics, and clustering methods for numerical structured data types have become very mature both domestically and internationally. There are now a large number of algorithms, and most of them have been improved. However, there is still no algorithm specifically designed for unstructured data. Therefore, text clustering mainly

consists of two steps. The first step is text analysis, which preprocesses the text and transforms it into an expression. The second is to use clustering algorithms to cluster the expressed text. The basis for text clustering is the same as traditional data clustering, which is to minimize the distance between texts of the same type and maximize the distance between different classes. Below is an overview of the text clustering process:

(1) *Text representation.* The representation of text has been introduced in detail in the previous text, which is also the foundation and important step of text mining and text clustering. Convert the original document, calculate feature weights, and select features to form a feature set for the next step of calculation.

(2) Cluster the processed text matrix, i.e., the feature set, using specific clustering algorithms, select appropriate clustering algorithms and apply them.

(3) *Analyze the clustering effect.* Compare the results of various algorithm implementations and analyze to determine the optimal algorithm suitable for this dataset. The following are the concepts involved in text clustering: measurement of similarity and clustering methods.

2.2.1. Measurement of Text Similarity. To achieve document clustering, it is necessary to measure the similarity between documents. Various methods exist for this purpose, and this section primarily discusses distance-based similarity measures. By representing a document as a feature vector, several common methods are used to measure the similarity between two document vectors D_i and D_j.

The Euclidean distance is a widely used metric for measuring similarity and is defined as:

$$\mathrm{dis}(d_i, d_j) = \sqrt{\sum_{k=1}^{n} (w_{ik} - w_{jk})^2}. \tag{1}$$

In Eq. (1), the Euclidean distance represents the sum of squared differences between corresponding feature weights of two documents. Here, w_{ik} and w_{jk} denote the weights of the k-th feature in documents d_i and d_j, respectively.

Cosine similarity is another commonly used measure, which calculates the cosine of the angle θ between two document vectors. A smaller angle implies higher similarity, with a value of 1 indicating that the two documents are identical. It is given by:

$$\mathrm{sim}(d_i, d_j) = \frac{\sum_{k=1}^{n} (w_{ik} w_{jk})}{\sqrt{\sum_{k=1}^{n} w_{ik}^2} \times \sqrt{\sum_{k=1}^{n} w_{jk}^2}}. \tag{2}$$

Eq. (2) computes the cosine similarity by normalizing the dot product of the document vectors with their magnitudes. A cosine similarity value close to 1 indicates high similarity, whereas a value near 0 signifies dissimilarity.

2.2.2. Cluster ensemble method. First, we give the Cluster ensemble method based on K-means. The clustering ensemble problem can be divided into the following four steps:

1) *Data subset generation:* Based on the original dataset, different data subsets are generated;

2) *Cluster center generation:* Using clustering methods to generate cluster centers for each subset of data;

3) *Clustering result generation:* i.e., generating different clustering results;

4) *Result conversion and merging:* that is, converting the category labels of different clustering results and effectively merging different clustering results.

Data subset generation: Use Bootstrap random sampling algorithm to split the original dataset into b subsets $\{S_1, S_2, \cdots, S_i \cdots, S_b\}$ $(i \in [1, b])$.

Cluster center generation: Use K-means to obtain each sub dataset s; The cluster centers represent one category and can generate b groups of cluster centers. K-means is an efficient clustering method. In this method, the clustering results are obtained by iteratively calculating the positions of the samples and the clustering centers. When the clustering results reach the convergence condition, the algorithm stops iterating and obtains the final clustering result. The convergence conditions include: 1) The cluster center no longer changes; 2) Two iterations before and after, where the sum of distances within all categories is less than the specified threshold; 3) Iteration exceeds a certain number of times. The details of the K-means algorithm include the following steps: 1) Randomly or strategically selecting K samples as initial cluster centers, with each cluster center representing a category; 2) Each sample is assigned to the category represented by the nearest cluster center; 3) Recalculate the cluster centers for each category; 4) If the convergence condition is not met, return to step 2.

Cluster result generation: The top m samples with the highest similarity values between the original dataset and the cluster center are assigned to the class represented by that cluster center, resulting in a total of b cluster results $\left\{\lambda^1, \lambda^2, \cdots \lambda^i, \cdots, \lambda^b\right\}$. The author uses the cosine similarity formula to calculate the similarity between the samples in the original dataset and the cluster centers.

Result conversion and merging: The clustering results of group b obtained by the above method cannot be directly merged. Firstly, record the number of identical objects covered by each pair of clustering labels C_i^a and C_j^b in the kxk matrix. Then, we choose the most similar group tags to build up the relation, and delete the relevant clustering tags from the matrix. Repeat this procedure until appropriate relations are set up for all of the clustering markers. After converting the clustering results, the author used a weight based selective voting strategy to merge the various clustering results. This strategy uses mutual information to calculate the weight of each clustering result. If two clustering results λ 2 and λ, where the original dataset has a total of n samples, there are ni texts in C_i^a, nj texts in C_j^b, and nij identical texts in C_i^a and C_j^b, then mutual information (MI) can be expressed as

$$\phi^{NMI}\left(\lambda^a, \lambda^b\right) = \frac{2}{n} \sum_{i=1}^{k} \sum_{j=1}^{k} n^{ij} \log_{k^2}\left(\frac{ni^j n}{n_i n_j}\right). \tag{3}$$

For each clustering result, its average mutual information value can be defined as:

$$\beta_m = \frac{1}{b-1} \sum_{I=1, I \neq m}^{b} \phi^{NMI}\left(\lambda^m, \lambda^I\right) (m \in [1, b]). \tag{4}$$

Among them, b represents the number of clustering results. In the clustering result λ^m, the larger the β m, the smaller the weight. The weight of λ^m can be defined as

$$W_m \frac{1}{\beta_m Z}, W_m > 0 (m \in [1, t]), \sum_{m=1}^{b} W_m = 1. \tag{5}$$

Among them, z is used to standardize the weights. Mutual information weights can be used to effectively select clustering results. This can exclude clustering results with mutual information below the threshold. The author's threshold is set to 1/b.

Now, for cluster ensemble method based on incremental clustering, there are two main steps in this approach: (1) *Generating basic clustering results:* In the group, we can get basic clustering

results by using various parameters. The writer creates a variety of basic clusters with parameter settings. (2) Transform and combine the classification tags of basic clustering results, and combine them efficiently. Generating elementary Clustering Results: Varying the Parameters of Incremental Clustering Algorithm can get a variety of Text Clustering Results. Incremental clustering algorithm considers the relationship between existing clusters and text, and is an incremental clustering algorithm. This algorithm consists of the following two steps: (1) *Similarity calculation:* Calculate the similarity between a new text and an existing cluster, and find the cluster with the highest similarity. The similarity is obtained by calculating the cosine similarity between the new text and the center of the cluster; (2) *Threshold setting:* Manually set a similarity threshold of m. If the similarity is higher than m, the new text will be classified into the cluster with the highest similarity; On the contrary, create a new class cluster for the new text. In order to obtain multiple different base clustering results, the author selected four different thresholds m, namely 0.1, 0.2, 0.3, and 0.4. The algorithm ultimately generates cluster results $\left\{\lambda^1, \lambda^2, \cdots, \lambda^b\right\}$ for group b, with the number of clusters set to 5.

The clustering result conversion and merging method is the same as that in the K-means based clustering ensemble method.

3. Results analysis

The author analyzed the performance of K-means based clustering ensemble (ECKM) and incremental clustering based clustering ensemble (ECIC) in academic text classification. In order to analyze whether clustering ensemble improves the clustering performance of academic texts, the author uses K-means and incremental clustering as benchmark methods to compare the performance differences between K-means and ECKM, as well as incremental clustering and ECIC. In addition, the author analyzed the impact of keywords on the classification of academic texts from the perspectives of different keyword extraction methods and different numbers of keywords.

3.1. Performance analysis of clustering ensemble method

This section mainly analyzes the performance differences between base clustering algorithms and clustering ensemble methods in academic text classification, comparing the F1 values of each dataset with the average F1 value of all datasets. The author conducted an academic text classification experiment using eight datasets. Table 1 shows the average F1 values of K-means and ECKM clustering methods on eight data subsets based on four keyword extraction methods: CSI, ECC, TextRank, and TFISF. Table 2 shows the average F1 values and standard deviations of incremental clustering (IC) and ECIC clustering methods on eight data subsets based on four keyword extraction methods: CSI, ECC, TextRank, and TFISF.

From Tables 1 and 2, it can be seen that the performance of clustering ensemble methods is generally higher than that of base classifiers. In the comparison between K-means and ECKM, when CSI and TFISF are used for keyword extraction, the F1 value of ECKM is higher than that of K-means; When ECC and TextRank are used as keyword extraction methods, some K-means have higher F1 values than ECKM, but the proportion of this part is relatively small, accounting for 32% and 41% of the total results, respectively. In order to conduct a detailed analysis of the performance differences between K-means and ECKM, the author used t-test to analyze the average F1 values of the two methods. The results (t=7.01, P=0.000) showed a significant difference in F1

Table 1. The average F1 value and standard deviation (mean ± SD) of K-means and ECKM on eight subsets of data

Clustering method	number	CSI	ECC	TextRank	TFISF
K-means	5	0.220±0.004	0.236±0.008	0.274±0.040	0.235±0.015
	10	0.223±0.003	0.302±0.047	0.324±0.042	0.281±0.066
	15	0.220±0.002	0.330±0.068	0.353±0.072	0.244±0.020
	20	0.225±0.003	0.340±0.071	0.344±0.053	0.303±0.047
	25	0.223±0.004	0.355±0.055	0.377±0.072	0.315±0.067
	30	0.220±0.002	0.372±0.051	0.383±0.053	0.330±0.070
ECKM	5	0.372±0.022	0.383±0.035	0.401±0.014	0.382±0.015
	10	0.364±0.033	0.366±0.043	0.406±0.030	0.373±0.044
	15	0.368±0.042	0.362±0.042	0.405±0.040	0.372±0.051
	20	0.348±0.033	0.370±0.053	0.418±0.028	0.361±0.051
	25	0.333±0.040	0.372±0.050	0.412±0.030	0.367±0.046
	30	0.330±0.044	0.371±0.035	0.413±0.016	0.370±0.035

Table 2. Average F1 value and standard deviation (mean ± SD) of incremental clustering (IC) and ECIC on eight subsets of data

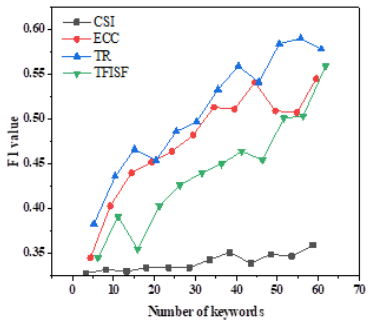
Clustering method	number	CSI	ECC	TextRank	TFISF
IC	5	0.218±0.007	0.252±0.023	0.322±0.050	0.274±0.044
	10	0.220±0.012	0.334±0.034	0.422±0.062	0.316±0.041
	15	0.230±0.006	0.382±0.041	0.436±0.050	0.326±0.027
	20	0.230±0.024	0.388±0.033	0.448±0.052	0.313±0.036
	25	0.234±0.040	0.428±0.042	0.462±0.044	0.357±0.014
	30	0.253±0.025	0.442±0.022	0.452±0.022	0.412±0.023
ECIC	5	0.218±0.007	0.251±0.021	0.331±0.053	0.244±0.084
	10	0.220±0.010	0.337±0.107	0.443±0.077	0.316±0.062
	15	0.237±0.017	0.414±0.064	0.441±0.053	0.320±0.052
	20	0.231±0.026	0.408±0.062	0.485±0.072	0.308±0.044
	25	0.240±0.037	0.441±0.044	0.506±0.051	0.380±0.044
	30	0.263±0.053	0.480±0.062	0.473±0.058	0422±0.058

values between the two methods, and ECKM performed significantly better than K-means. In the comparison between incremental clustering and ECIC, among the four keyword extraction methods, some incremental clustering had higher F1 values than ECIC, but this part was smaller, accounting for 25%, 25%, 17%, and 33% of the total results, respectively. In addition, the T-test results (t=3.413, P=0.000) indicate a significant difference in F1 values between the two, and the performance of ECIC is significantly higher than that of incremental clustering. These observations indicate that clustering ensemble is more suitable for automatic classification of academic text categories.

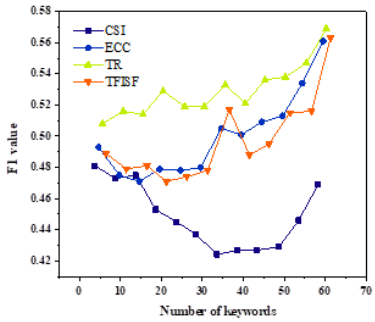
3.2. The impact of keyword count on the performance of text clustering ensemble

To analyse the influence of key word count on the performance of the research group, this part gives an in-depth study and comparative study on different kinds of key words. Figure 3 shows the average F1 score for different numbers of keywords. Figures 3a and 3b show the results of K-means and ECKM, respectively, while Figures 3c and 3d show the results of incremental clustering and

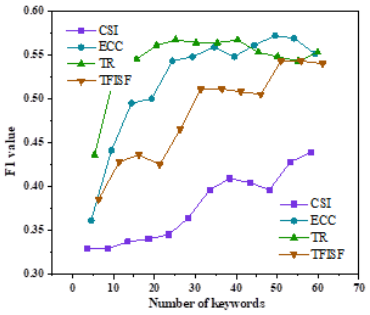
ECIC, respectively.



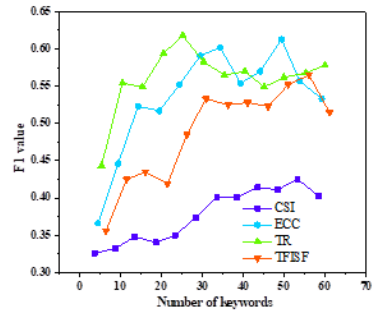
(a) Performance differences of K-means clustering method under different numbers of keywords



(b) Performance differences of ECKM clustering methods under different numbers of keywords



(c) Performance differences of incremental clustering methods under different numbers of keywords



(d) Performance differences of ECIC clustering methods under different numbers of keywords

Fig. 3. Average F1 score for different numbers of keywords

(1) The increase in the number of keywords promotes the improvement of clustering ensemble performance. From Figure 3, we can see that with K means and ECKM approaches, there is a tendency to improve with more key words. Moreover, when there are fewer key words, K means has a low F1, whereas ECKM has a big F1. When there are a large number of keywords, both K-means and ECKM have higher F1 values. Therefore, when the number of keywords is small, ECKM is more effective, and as the number of keywords increases, the performance difference between K-means and ECKM gradually narrows. In incremental clustering and ECIC methods, as the number of keywords increases, the performance of incremental clustering also shows an improvement trend.

(2) The ECIC method has a greater advantage when the number of keywords is small. According to Figure 3d, under the ECIC method, the performance is optimal when the number of keywords is 20-25, which significantly exceeds the performance of incremental clustering. When there are a large number of keywords, although the performance of ECIC is higher than that of incremental clustering, the difference between the two is relatively small. Therefore, the ECIC method has significant advantages when the number of keywords is small.

(3) ECKM has relatively stable performance across different numbers of keywords. In Figure 3a, in K-means, except for the low clustering performance under CSI, the average F1 value varies greatly with different numbers of keywords. There are differences between ECKM and K-means phenomena, as shown in Figure 3b, where the average F1 value changes slightly with the number of keywords. This indicates that under the ECKM clustering method, the performance of the average F1 value is

more stable under different numbers of keywords.

4. Conclusion

In the research on theme clustering and text mining of contemporary Chinese literary texts in the era of the internet, keyword based clustering ensemble methods have also demonstrated significant advantages. The author explores the application of thematic clustering in processing literary texts and analyzes the following three core issues: Firstly, can the clustering ensemble method improve the effectiveness of clustering based on Chinese modern and contemporary literary texts; Secondly, will the number of different keywords have an impact on the performance of clustering ensemble; Finally, how to choose the appropriate keyword extraction method to optimize the clustering results. In order to address these issues, the author conducted a comparative study using two clustering ensemble methods: One is a data based clustering ensemble based on K-means, and the other is an algorithm based ensemble based on incremental clustering. In terms of selecting keyword extraction methods, the author chose four classic unsupervised single text keyword extraction methods, TF-ISF, CSI, ECC, and TextRank, and conducted experiments by extracting 5, 10, ..., and 60 keywords from Chinese contemporary literature texts, respectively.

Experiments indicate that the clustering approach can greatly increase the efficiency of literature topic clustering, especially under different keyword extraction methods and keyword quantities, the ensemble method exhibits better stability. Clustering is also enhanced with more keyword count, but when there are fewer key words, the clustering ensemble method still maintains high performance. Based on the experimental results, we draw the following conclusions: Firstly, when clustering the themes of Chinese modern and contemporary literary texts, using clustering ensemble methods can effectively improve the accuracy of theme recognition; Secondly, the keyword extraction method and the number of keywords have a significant impact on the clustering effect of text topics. Therefore, when representing text, it is necessary to carefully choose the keyword extraction method and use more keywords under possible conditions. In addition, when the number of keywords is small, clustering ensemble methods can effectively compensate for the performance decline caused by insufficient keywords and demonstrate stronger robustness. Therefore, in text mining research, clustering ensemble method is an effective means to address the insufficient number of keywords and improve the effectiveness of topic recognition.

References

[1] L. M. Abualigah and A. T. Khader. Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. *The Journal of Supercomputing*, 73:4773–4795, 2017. https://doi.org/10.1007/s11227-017-2046-2.

[2] L. M. Q. Abualigah et al. *Feature selection and Enhanced Krill Herd Algorithm for Text Document Clustering*, volume 816. Springer, 2019. https://doi.org/10.1007/978-3-030-10674-4.

[3] C. C. Aggarwal and C. C. Aggarwal. *Machine Learning for Text: An Introduction*. Springer, 2018. https://doi.org/10.1007/978-3-319-73531-3_1.

[4] S. A. Curiskis, B. Drake, T. R. Osborn, and P. J. Kennedy. An evaluation of document clustering and topic modelling in two online social networks: twitter and reddit. *Information Processing & Management*, 57(2):102034, 2020. https://doi.org/10.1016/j.ipm.2019.04.002.

[5] C. Guangwei. *The "Historicization" of Contemporary Literature*. Taylor & Francis, 2024. https://doi.org/10.4324/9781003505716.

[6] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, and M. R. Yeganegi. Text mining in big data analytics. *Big Data and Cognitive Computing*, 4(1):1, 2020. https://doi.org/10.3390/bdcc4010001.

[7] K. Hladíková. In the name of stability: literary censorship and self-censorship in contemporary china. In *The Routledge Handbook of Chinese Studies*. Taylor & Francis, 2021. https://doi.org/10.4324/9780429059704-41.

[8] S. M. J. Jalali, H. W. Park, I. R. Vanani, and K.-H. Pho. Research trends on big data domain using text mining algorithms. *Digital Scholarship in the Humanities*, 36(2):361–370, 2021. https://doi.org/10.1093/llc/fqaa012.

[9] A. Karami, M. Lundy, F. Webb, and Y. K. Dwivedi. Twitter and research: a systematic literature review through text mining. *IEEE Access*, 8:67698–67717, 2020. https://doi.org/10.1109/ACCESS.2020.2983656.

[10] X. Kong, Y. Shi, S. Yu, J. Liu, and F. Xia. Academic social networks: modeling, analysis, mining and applications. *Journal of Network and Computer Applications*, 132:86–103, 2019. https://doi.org/10.1016/j.jnca.2019.01.029.

[11] S. Kumar, A. K. Kar, and P. V. Ilavarasan. Applications of text mining in services management: a systematic literature review. *International Journal of Information Management Data Insights*, 1(1):100008, 2021. https://doi.org/10.1016/j.jjimei.2021.100008.

[12] N.-C. Luo. Massive data mining algorithm for web text based on clustering algorithm. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 23(2):362–365, 2019. https://doi.org/10.20965/jaciii.2019.p0362.

[13] A. J. Obaid, T. Chatterjee, and A. Bhattacharya. Semantic web and web page clustering algorithms: a landscape view. *EAI Endorsed Transactions on Energy Web*, 8(33):e7–e7, 2021. https://doi.org/10.4108/eai.18-11-2020.167099.

[14] F. Rahma. Cyber literature: a reader–writer interactivity. *International Journal of Social Sciences & Educational Studies*, 3(4):156–164, 2017. https://doi.org/10.23918//ijsses.v3i4p156.

[15] S. Rettberg. *Electronic Literature*. John Wiley & Sons, 2018.

[16] S. A. Salloum, M. Al-Emran, A. A. Monem, and K. Shaalan. Using text mining techniques for extracting information from research articles. *Intelligent Natural Language Processing: Trends and Applications*:373–397, 2018. https://doi.org/10.1007/978-3-319-67056-0_18.

[17] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, and J. Zhao. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88:22–31, 2017. https://doi.org/10.1016/j.neunet.2016.12.008.