Hindawi
Security and Communication Networks
Volume 2023, Article ID 9869657, 1 page
https://doi.org/10.1155/2023/9869657

Retraction

Retracted: A Novel Literary Translation Text Classification Method Based on Distributed Incremental Sequence Data Mining Algorithm

Security and Communication Networks

Received 11 July 2023; Accepted 11 July 2023; Published 12 July 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] J. Sun, "A Novel Literary Translation Text Classification Method Based on Distributed Incremental Sequence Data Mining Algorithm," *Security and Communication Networks*, vol. 2022, Article ID 1656151, 10 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1656151, 10 pages
https://doi.org/10.1155/2022/1656151

Research Article

A Novel Literary Translation Text Classification Method Based on Distributed Incremental Sequence Data Mining Algorithm

Ji Sun

Zhejiang International Studies University, Hangzhou 310023, China

Correspondence should be addressed to Ji Sun; sunji@zisu.edu.cn

Received 13 March 2022; Revised 6 April 2022; Accepted 15 April 2022; Published 9 May 2022

Academic Editor: Chin-Ling Chen

With the rapid development of electronic information technology and Internet technology, people's ability to generate and collect data is also increasing. With the rapid development of international exchanges, the number of literary translation texts has also increased dramatically. The information contained in the huge data of literary translation texts is huge, but these data are disorganized at this stage. Therefore, the classification of literary translation texts has become the key to efficient management of translated text information. Literary translation text classification is the process of classifying a given text as one or several of several predetermined text categories according to the content of the text. As one of the key steps in processing huge amounts of text data, text classification is generally regarded as the orderly organization of text sets, that is, grouping similar and related texts together. In this way, the problem of information clutter can be solved to a greater extent, and the efficiency of users' discovery, filtering, and analysis of text information resources can be effectively improved. At present, the basis of text classification is mainly based on the characteristics of words in the article, to analyze the correlation between words and categories. This approach ignores information such as word order and collocations in literary translation texts. To solve this problem, this paper introduces the distributed incremental sequence data mining algorithm into the classification of literary translation texts. This method can fully mine the features of syntactic order and other features based on considering the words and phrasing characteristics of articles in different application fields. The text classification effect is strengthened by discovering more effective information. The experimental results show that the method can improve the classification performance of literary translation texts.

1. Introduction

With the advancement of globalization, international exchanges and learning have become more common, and the dissemination of literary texts has grown in scope. As the world's primary language of communication today, English has gradually produced translated texts from various literature. Many literary translation texts were created over time. When confronted with a massive amount of literary translation text information, knowing how to efficiently filter useful information, accurately locate information categories, and deftly explore hidden information has become increasingly important. Based on this demand, text classification technology emerges as needed, having a significant impact on the process of knowledge discovery. The early text classification was done entirely by hand, and the work was arduous, clumsy, time consuming, labor intensive, and inefficient. Until the late 1950s, the IBM Corporation of the United States led the way in incorporating the concept of word frequency statistics into the text classification process, with positive results. This also marks the beginning of a new phase in text classification research. Many experts and scholars have conducted extensive and in-depth research on text classification methods since then, and many text classification methods have been proposed.

Because of Internet's rapid development, all types of information are flooding the network, and a keyword search in a search engine will return millions of results. The ability to use effective classification methods to filter useful category information has a wide range of applications in real life [1–3]. Many researchers from various fields have conducted extensive research on this topic, and various text

classification methods have emerged [4, 5]. Text classification based on similarity is a widely used method in the field of text classification [6–8]. Several methods for calculating vector similarity have been proposed, including Euclidean distance, Jaccard similarity, cosine similarity, Kullback–Leibler divergence, Canberra distance metric, hammering distance, dice coefficient, and pairwise adaptive, among others. Measures of similarity have been developed. It is commonly used in text classification [9, 10]. This different relationship, however, is a significant factor influencing classification performance. The use of bag-of-words is also very mature in the field of text classification [11–13]. A document is frequently represented as a vector, with each component containing the value corresponding to the corresponding feature. This feature value could be word frequency, associated word frequency, or something else. Many studies have begun to conduct in-depth research on text lexical rules in order to improve the performance of text classification [14]. Because of their simplicity, word-based methods are preferred by researchers in a variety of fields. At this point, many word-based feature selection methods have been proposed by various scholars [15–17]. However, word-based methods usually ignore the association between words, which will lead to loss of information about paragraphs and sentences in the text and will easily cause problems such as polysemy, synonyms, and noise interference [18].

Texts in various application fields, in general, have their own distinct writing characteristics, or literary habits. Some of these habits are reflected in the article's wording and phrasing, while others are reflected in word collocation, syntactic order, or other aspects. Because of the presence of these characteristics, we can easily identify texts in a variety of application fields. As a result, whether it is the initial manual recognition or machine recognition, the information used for text classification is likely to belong to the article characteristics of the application field contained in each document. Most current text classification methods primarily complete the task of text classification by identifying, extracting, and utilizing textual characteristics in various fields. The current problem is that the traditional classification method still focuses on the relevance of the article and only determines which feature words correspond to the field. However, words are relatively isolated from each other, and the characteristics of word collocation and syntactic order are not considered, so the information that can be extracted is relatively limited. Sequence pattern mining can dig out the pattern sequences hidden in the transaction set, and all the elements in these pattern sequences are frequent and relatively sequential. This order can be relative to time or relative to space. This shows that the sequential pattern mining method can fully and effectively utilize the sequential characteristics of transactions. In view of the shortcomings of the existing text classification methods, this paper attempts to introduce the idea of sequential pattern mining into the text classification method and uses the distributed incremental sequential data mining algorithm to propose a method of literary translation text based on the distributed incremental sequential data mining algorithm. The new method can fully mine text syntactic sequence features such as word order and word collocation based on considering the characteristics of words and phrasing of articles in different application fields. Thereby, more effective information can be discovered to strengthen and improve the effect of text classification.

2. Knowledge Based on Sequence Patterns

2.1. Sequence Pattern Data Mining. The goal of sequential pattern mining [19] is to find all sequential patterns in a sequence database that have a frequency greater than a predefined threshold. Mining for sequence patterns is analogous to mining for association rules. However, the former is more concerned with the temporal or spatial dependencies of event elements than the latter, i.e., the former is more concerned with event sequence characteristics. This order can be both temporal and spatial in nature. The original goal of sequence mining was to uncover sequences of frequent buying patterns in a transaction database with transaction times. In this method, we can determine the purchase behavior trajectory of most clients over a certain time period. Later, with the deepening of algorithm research and the upgrading of technology, especially the optimization of mining algorithms and the use of high-performance computers, the application field of sequential pattern mining has become very broad. At present, it is widely used in professional fields such as customer purchasing behavior prediction, medical diagnosis, web page access pattern prediction, industrial control, and gene sequence analysis.

Sequential pattern mining is one of the important technologies in the field of data mining and has applications in many fields. At present, there are a lot of related research studies on sequential pattern mining algorithms, and the proposed algorithms are also very comprehensive. According to different mining strategies, sequential pattern mining algorithms can be divided into two categories, one is the algorithm based on the breadth-first search strategy, and the other is the algorithm based on the depth-first search strategy. Among them, the representative algorithms based on the breadth-first search strategy are the Apriori-All algorithm [20], the GSP algorithm [21], and the SPADE algorithm [22]. Typical depth-first search algorithms mainly include FreeSpan algorithm [23], PrefixSpan algorithm [24], and SPAM algorithm [25].

2.2. Text Classification Based on Sequential Patterns. As illustrated in Figure 1, text classification based on sequential patterns consists mostly of two stages. The first task is to create a library of classification patterns. The sequential pattern mining approach is used to mine the sequential patterns belonging to this category for each batch of training texts from a known category, resulting in the formation of the pattern sublibrary of this category. After each category's sublibraries are generated, they are aggregated into the final pattern library $P = \{P_1, \ldots, P_j, \ldots, P_k\}$. The next step is to examine the freshly entered text. The input text sequence is

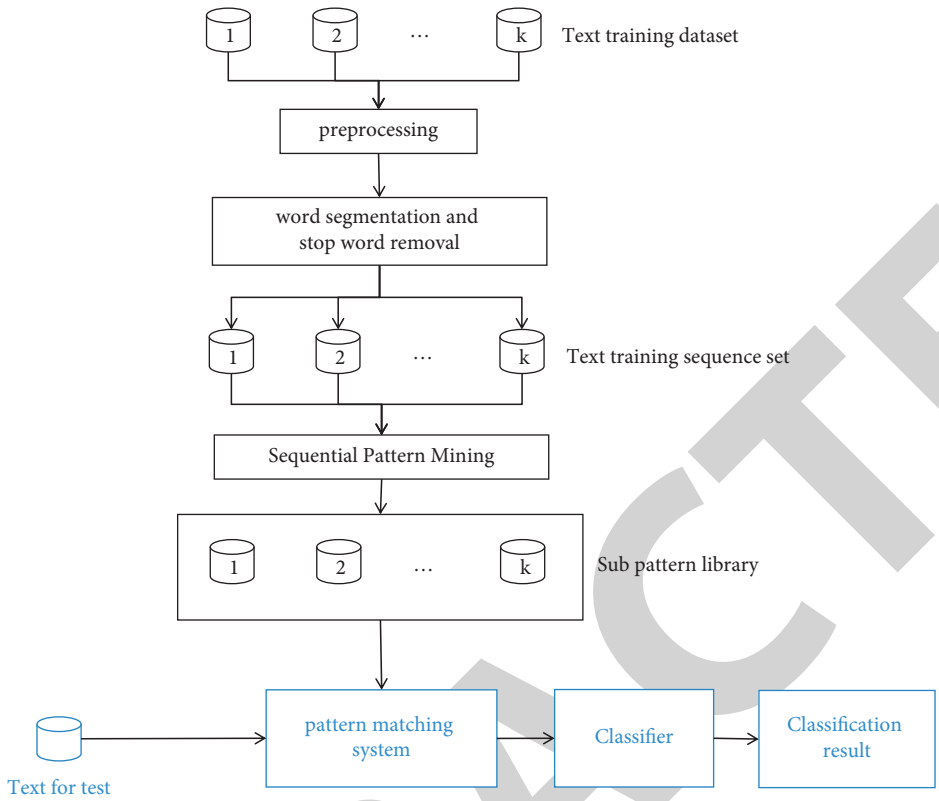

Figure 1: Architecture diagram of text classification based on sequence pattern.

generated by turning each text into a line of text characters using related approaches. Second, pattern matching is carried out. Finally, the text is classified in accordance with the classification principle. The classification of literary translation texts in this paper is based on the framework shown in Figure 1.

According to Figure 1, it can be concluded that the text classification process based on sequence patterns is as follows:

Text preparation is the first. Only discrete data can be analyzed with sequential pattern mining algorithms, and it is not always evident whether a dataset is discrete. If the data are continuous, it must first be discretized. To obtain a set of word sequences for each category, we first store all the texts in the training text set individually according to distinct categories and then execute word segmentation and stop word removal processing on them according to a unified standard.

Second, each category's frequent sequence patterns are mined to create a categorization pattern library. Each category's set of word sequences is turned into a standard set of transactions with timestamps and sequence IDs. The sequential pattern mining method is then utilized for the standard transaction sets of different categories to uncover the frequent patterns belonging to each category, and the subpattern library belonging to each category is obtained. The categorization pattern library is then obtained.

Finally, text classification is carried out. The new text is input to be categorized, and pattern matching is executed with all frequent patterns in the classification pattern library that has been created. We count the number of common patterns that match the new text in each subpattern library. Finally, documents having unidentified category attribute values are categorized using the definition classification principle.

3. Distributed Incremental Sequence Data Mining Algorithm

3.1. Distributed Computing Framework MapReduce. A sequential pattern mining method based on the distributed lexical sequence tree algorithm was proposed in reference [26]. Based on this research, this work employs a MapReduce-based distributed incremental sequential pattern mining approach. In a huge data setting, the approach can be utilized to overcome the incremental maintenance problem of sequential pattern mining. To deal with huge data challenges, MapReduce employs a distributed programming framework that employs a divide-and-conquer method. The system conceals the internal mechanics of data segmentation and distribution, task scheduling, intermachine communication, and fault tolerance from the programmer. It enables inexperienced programmers to efficiently manage the system's resources in a distributed system. The MapReduce approach not only simplifies distributed programming, but it also allows for more efficient processing of big data volumes.

Our method is divided into two MapReduce stages. The first stage consists primarily of reading the sequence's input split and determining the support count of the frequent 1

itemsets, as well as determining whether the frequent 1 itemsets belong to the incremental dataset using flag variables. The CMP data structure is created in the second stage. The candidate data are generated by using the CMP data structure and backward expansion. Furthermore, the prepruning attributes used in backward mining prevent erroneous candidate sequences from being generated in the input database, which speeds up the mining process.

3.2. Mining Frequent 1 Sequence. The first stage is used to mine frequent 1 sequence. Each Map1 identifies the input sequence dataset and determines whether item x belongs to the corresponding sequence's incremental dataset IDB. Item x stores 1 in distributed cache F if item belongs to IDB, otherwise 0. Then, Map outputs item x and its matching value in F as <key, value> and Reduce1<x, value> as input. The item count and flag variables are set to zero. The function of the flag variable is to indicate whether the item is included in the IDB. Reduce1 is used to determine how many values are related to each item. Reduce1 sets the flag variable to 1 after getting the item value 1, which aids in finding the incremental union in the second stage. At last, as the first stage's output, the frequent 1 itemsets, and their related counts and flags are used. Figure 2 depicts the first stage of this method's procedure.

3.3. Mining Frequent k Sequences. First, the frequent 1 itemsets, minimum support counts, and flags in the original dataset output from the first stage are used as the input of the second stage. Second, Map2 creates incremental unions and constructs CMP data structures, and uses prepruning attributes for backward mining. Finally, Reduce2 is used to mine frequent k sequences. The second stage process is shown in Figure 3.

3.3.1. Discovering the Incremental Union. The distributed cache file is read in the second stage, and the incremental union is found. For each item in the distributed cache file, we check if its flag is equal to 1. If it is equal to 1, we join the item to the incremental union, otherwise the 1 mode cannot be added to the incremental union.

3.3.2. Constructing the CMP Data Structure. This paper constructs the CMP data structure to optimize the speed of the algorithm. CMP(i) construction: By definition, CMP(i) is a mapping of an item and its preceding item list (co-occurrence list, CLST) relative to the extension of the itemset. The sequence scans from the last itemset to the first to match the CLST. If there is corresponding item to εi exists in CMP(i), then the CLST of εi is retrieved from CMP(i) of εi, and the co-occurrence item εj is checkedcby the retrieved CLST to find its count. If item j did not previously exist in CLST, it is included in CLST (εi). If the entry in CMP(i) corresponding to term i does not exist, then a CLST for term i is created by including the co-occurrence term εj. The CMP(i) of εi is updated with the corresponding CLST CMP(s) construction: By definition, a CMP(s) is an item mapping to its preceding list relative to the sequence expansion, and creating a CMP(s) is basically the same as creating a CMP(i).

3.3.3. Backward Mining Algorithm. The input of the backward mining algorithm is the sequence, the end projection (Epj) and H projection (Hpj) of the CMP(i), and CMP(s) sequences of the first item of the sequence, and the output is the updated data set frequent sequential patterns in UDB. The 1-mode end projection is the Epj from the backward extension. The sequence's backward extension is inferred by the CMP(i) and CMP(s) of the sequence. If the CMP(i) of the sequence does not exist, then itemset extension does not exist. The Epj of the generated sequences can be found in the Epj of the sequences, and the H projections of the sequences are scanned to find the H projections of the sequences extending from them. Support counts for these sequences are the sum of Epj's and Hpj's size. When the support counts of all extensions meet the lower threshold, they will be added into frequent sequences list and be called with recursion.

3.3.4. Generating Candidate Sequences. After creating the CMP data structure, a candidate set is generated. This paper designs two candidate generation rules based on backward mining to avoid generating misleading data so as to generate sequences of candidate efficiently. We assume the following definitions:

(1) For a given sequence pattern of length k $s = [\varepsilon_k \varepsilon_{k-1} \ldots \varepsilon_1]$, generate itemset candidates from a extensions that belong to CMP(i) (b) $C^{\varepsilon}_{k+1} = [\{a\varepsilon_k\}\varepsilon_{k-1} \ldots \varepsilon_1]$, where b is the first term in ε_k

(2) For a given sequence pattern of length k $s = [\varepsilon_k \varepsilon_{k-1} \ldots \varepsilon_1]$, generate sequence candidates by extending a that belong to CMP(s) (b) $C^{s}_{k+1} = [\{a\varepsilon_k\}\varepsilon_{k-1} \ldots \varepsilon_1]$, where b is the first term in ε_k

3.3.5. Generating Prepruning Sequences. Computing the support counts is quite slow. Mining speed can be improved if the generated candidates are pruned before computing support counts. This paper defines prepruning properties based on the CMP data structure. Sequences are trimmed early if their length exceeds a certain value, according to their properties. As shown in Figure 4, the following definitions are assumed:

3.3.6. Algorithm Implementation. The resources used are initialized through the SETUP function in the second phase of the MapReduce platform in this paper. After getting the input, Mappers treats infrequent items with pruning, then creates CMP(i) and CMP(s), and finds Projection and end projection of each 1-mode a. Each 1-mode is checked to see if it belongs to an incremental union to get the 1-modes of stable and expand them to seek the frequent sequences. If 1 mode is unstable, the function of backward mining will be

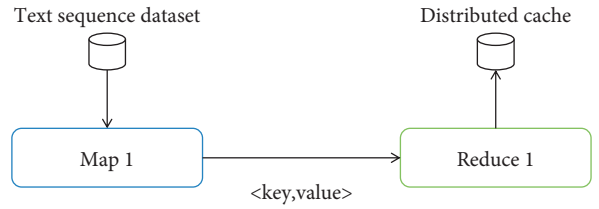

Figure 2: Flowchart of the first stage.

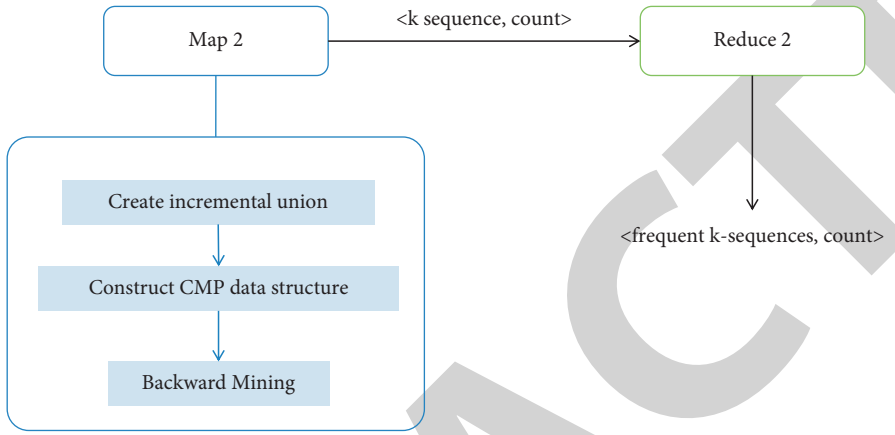

Figure 3: Flowchart of the second stage.

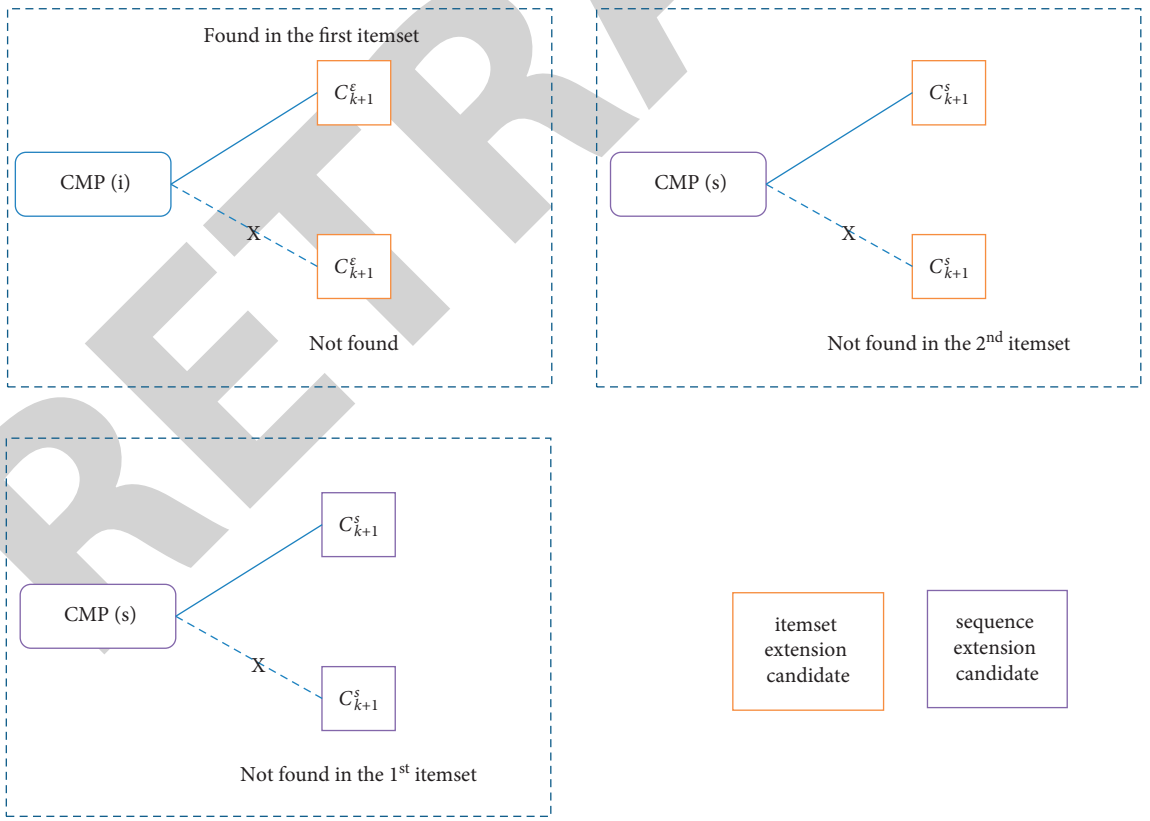

Figure 4: Definitions of prepruning.

called for each unstable ones, and 5 parameters are passed to the function when calling: unstable 1-mode a, the extended sequence of a, the itemset extended sequence of a, the end projection of a, and the H projection of a, where the H projection of a is calculated from the difference between the projection of a and the end projection of a.

Reduce2 takes k and value v as input and initializes the item count to 0. Reduce2 computes the items' values, after receiving the value about the item, and increments the item count. If the item count is greater than the product of the min support count and $|UD|$, we output k and item counts and then write them to files in the Hadoop distributed file system HDFS.

4. Experiment

4.1. Experimental Design. The hardware environment used in the experiment in this paper is CPU frequency 3.40 GHz; memory 4G. Software environment: Operating system: Windows10; IDE : MyEclipse9.0; Development environment: JDK 1.6; Development language: Python 2.7; Database: MySql.

The evaluation metrics used in the experiment are Recall (R), Precision (P) and F1. Recall refers to the ratio of the number of samples correctly judged by the classifier to the total number of samples belonging to this class; precision refers to the proportion of samples that truly belong to this class among the samples judged by the classifier as this class. We assume that when retrieving documents from a large-scale document collection, the documents can be divided into four groups: A represents relevant documents retrieved by the system; B represents irrelevant documents retrieved by the system; C represents relevant but not retrieved by the system documents; D stands for irrelevant documents that have not been retrieved by the system. Then, the text classification recall and precision can be defined as

$$\begin{aligned} \text{Recall} &= \frac{A}{A+C} \times 100\%, \\ \text{Precision} &= \frac{A}{A+B} \times 100\%, \\ F1 &= 2 \times \text{Recall} \times \frac{\text{Precision}}{(\text{Recall} + \text{Precision})}. \end{aligned} \tag{1}$$

In general, the text classification process can be divided into two parts: training the model and classifying the text. The main goal of training is to construct a classification model using the special relationship between text features and text categories through a set of training texts of known categories. The generalized model training process includes five steps: acquisition of training text sets, text preprocessing, text feature extraction, text representation, and selection of classification algorithms to construct classifiers. Categorizing text is to use the classifier obtained from the above training results to classify new and unknown text into categories, and the process of "sticking" category labels, mainly including text preprocessing, text representation, text classification, and classification performance evaluation. The main feature selection methods used in this step are mutual information (MI) and information gain (IG). In order to compare the performance of text algorithms, the comparison classification algorithms used include classic support vector machines (SVM) and random forests (RF).

4.2. Experimental Data. The datasets used in the experiments are two public corpora: Reuters-21578 and 20-Newsgroups.

4.2.1. Reuters-21578. The Reuters-21578 corpus is widely used for text mining. The data, originally collected by Carnegie Corporation and Reuters newsgroups, contained 21,578 documents on 135 different topics. The experiment uses a total of 9,980 documents from the top 10 topics. Because CORN and WHEAT are closely related to GRAIN, they are classified into GRAIN, so they are also called R8. This paper uses the 10-fold cross-validation method to divide the training dataset and the test dataset. We choose 80% as the training set and 20% as the test set. Table 1 describes the categories of documents in the dataset Reuters-21578.

The 20-Newsgroups dataset contains 4 major categories: COMP, REC, SCI, and TALK; each category contains 4 subcategories, with a total of 15,033 documents. Table 2 describes the document descriptions for each category in the dataset 20-Newsgroups. We choose 80% as the training set and 20% as the test set.

4.3. Experimental Results. The experimental results obtained by each classification algorithm on the Reuters-21578 dataset are shown inTable 3–5.

From the experimental data in Table 3–5 and the comparison chart of the classification results of each method shown in Figure 5, the following experimental conclusions can be drawn:

(1) Regardless of the classification method, the experimental results obtained by the IG-based feature extraction method are generally better than those obtained by the MI-based feature extraction method. This is because the IG-based feature extraction method can perform global feature extraction, and the extracted features are often valid for all classes.

(2) When the IG feature extraction method is used, the classification performance of the method proposed in this paper is better than other methods in most cases. The classification performance of RF is not much different from that of the method used in this paper, especially in the text classification of the first 6 classes. When using the MI feature classification method, comparing the experimental results obtained by different classifiers, in the first four categories of text classification, the method used in this paper is close to the classification performance of RF and the performance in the latter four categories. In this paper, the method used has distinct advantages. From the beginning to the end, the classification performance shown by the SVM method is not very good.

(3) Based on the above analysis, the method proposed in this paper can obtain better results than other classifiers no matter which feature extraction method is used. This fully shows that the method in this paper has certain advantages in the similar methods. In addition, according to the size of the specific data in each table, the text classification accuracy obtained by this method

Table 1: Dataset Reuters-21578 details.

Category	Total of each category	Training number	Testing number
ACQ	2369	1895	474
CRUD	578	462	116
EARN	3964	3171	793
GRAIN	582	466	116
INTEREST	478	382	96
MONEY	717	574	143
SHIP	286	229	57
TRADE	486	389	97

Table 2: Dataset 20-Newsgroups details.

Category	Total of each category	Training number	Testing number
COMP	3870	3096	774
REC	3968	3174	794
SCI	3945	3156	789
TALK	3250	2600	650

Table 3: Recall values on the Reuters-21578 dataset.

Method	Feature extraction	ACQ	CRUD	EARN	GRAIN	INTEREST	MONEY	SHIP	TRADE
SVM	IG	0.4835	0.4909	0.6536	0.5986	0.5069	0.5985	0.4943	0.4665
	MI	0.4328	0.4251	0.5902	0.5265	0.4401	0.5336	0.4438	0.4240
RF	IG	0.5002	0.8418	0.7420	0.8910	0.7084	0.6894	0.8165	0.5900
	MI	0.4301	0.7577	0.6436	0.7970	0.6262	0.5988	0.6815	0.5172
Proposed	IG	0.5976	0.8417	0.8574	0.8884	0.9608	0.824	0.9061	0.8217
	MI	0.5395	0.7306	0.6960	0.7303	0.8612	0.802	0.8105	0.6827

Table 4: Precision values on the Reuters-21578 dataset.

Method	Feature extraction	ACQ	CRUD	EARN	GRAIN	INTEREST	MONEY	SHIP	TRADE
SVM	IG	0.5313	0.5103	0.6611	0.6224	0.5574	0.6072	0.4982	0.5290
	MI	0.4568	0.4294	0.6148	0.5315	0.4473	0.5997	0.4972	0.4470
RF	IG	0.5109	0.8771	0.7785	0.9149	0.7149	0.7164	0.8648	0.6454
	MI	0.4471	0.7588	0.6517	0.8007	0.6884	0.6223	0.7236	0.5617
Proposed	IG	0.6692	0.9135	0.9370	0.9288	0.8932	0.8821	0.9645	0.8534
	MI	0.5452	0.7463	0.7226	0.7912	0.8539	0.810	0.8671	0.6876

Table 5: F1 values on the Reuters-21578 dataset.

Method	Feature extraction	ACQ	CRUD	EARN	GRAIN	INTEREST	MONEY	SHIP	TRADE
SVM	IG	0.5063	0.5004	0.6573	0.6103	0.5310	0.6028	0.4962	0.4958
	MI	0.4445	0.4272	0.6022	0.5290	0.4437	0.5647	0.4690	0.4352
RF	IG	0.5055	0.8591	0.7598	0.9028	0.7116	0.7026	0.8400	0.6165
	MI	0.4384	0.7582	0.6476	0.7988	0.6558	0.6103	0.7019	0.5385
Proposed	IG	0.6314	0.8761	0.8954	0.9082	0.8555	0.8512	0.9344	0.8373
	MI	0.5423	0.7384	0.7091	0.7595	0.8320	0.7680	0.8378	0.6851

generally exceeds 0.8, which shows that the text classification based on this method has certain practicability.

The experimental results obtained by each classification algorithm on the 20-Newsgroups dataset are shown in Table 6:

From the experimental data in Table 6 and the comparison chart of the classification results of each method shown in Figure 6, the following experimental conclusions can be drawn:

(1) The 20-Newsgroups dataset has 4 categories, and the experimental data of the first 3 categories shown in

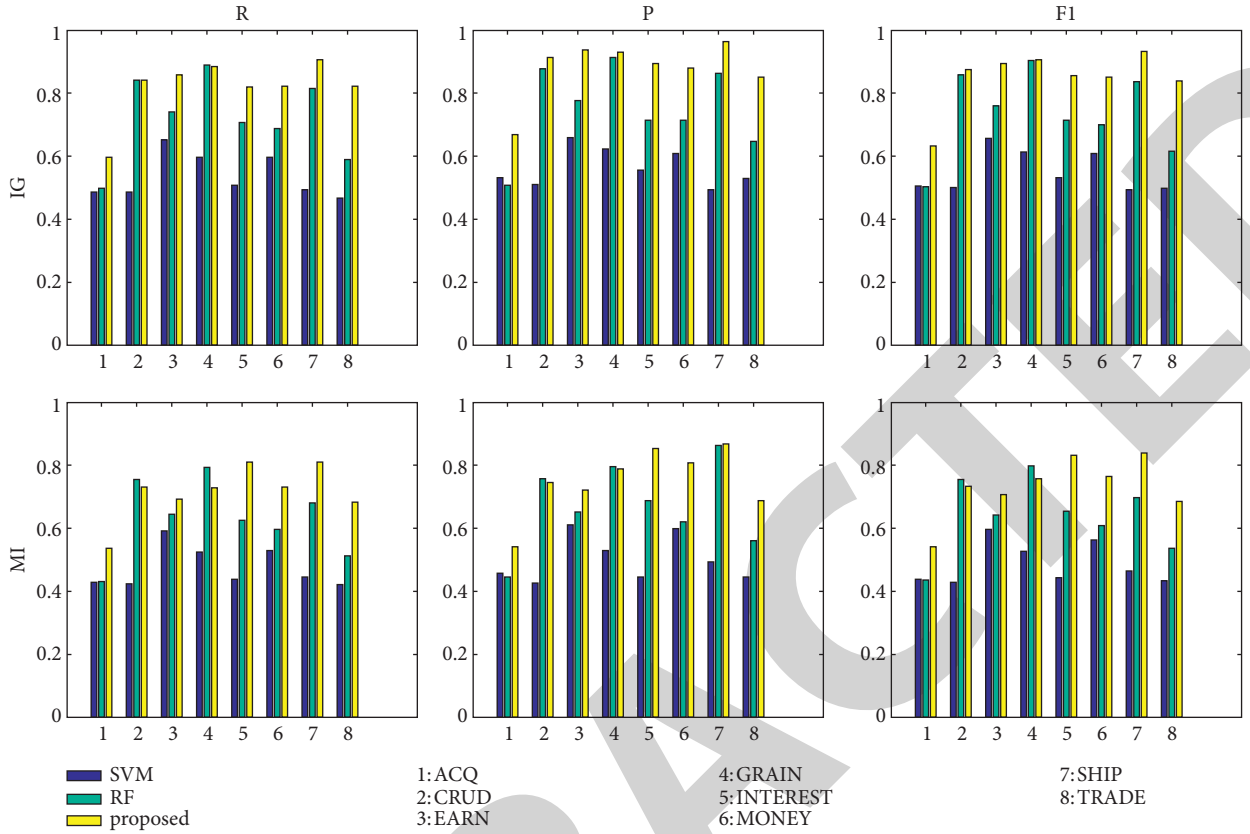

Figure 5: Comparison of experimental results on the Reuters-21578 dataset.

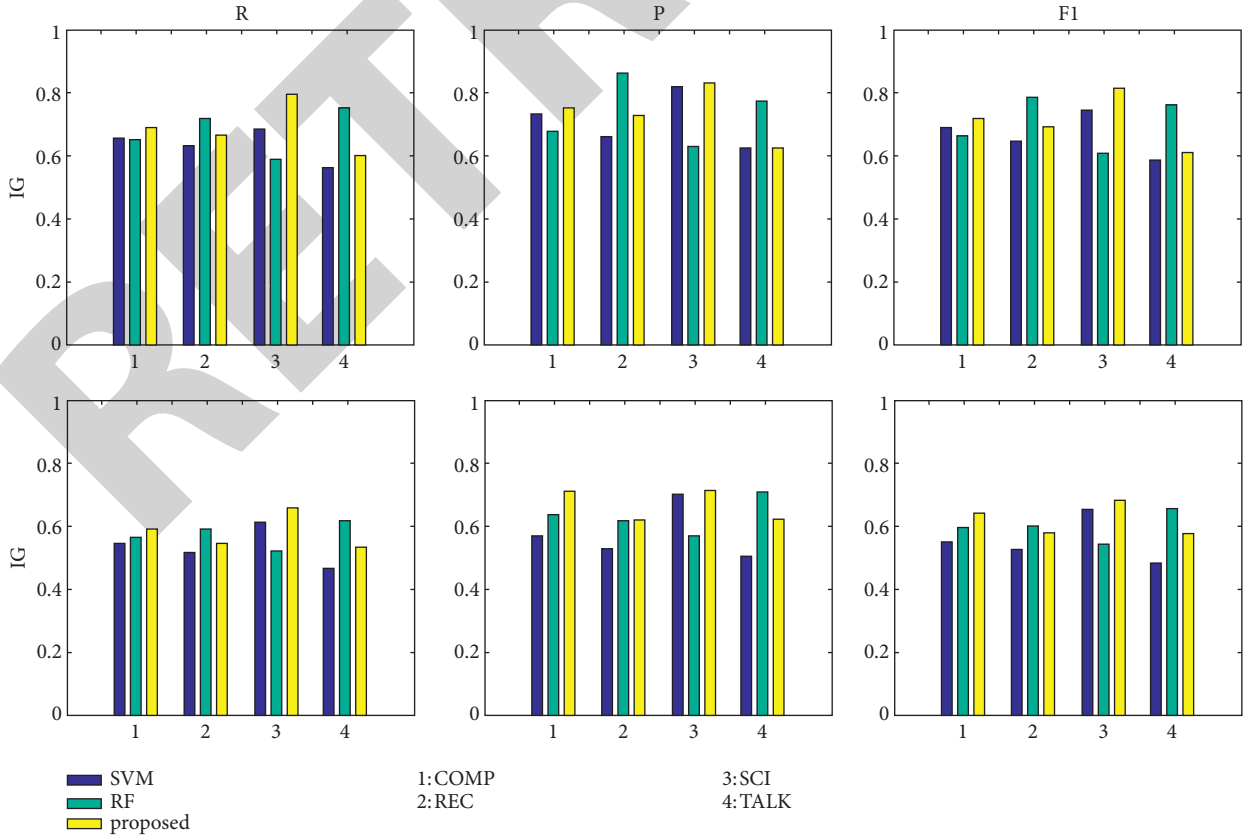

Figure 6: Comparison of experimental results on the 20-Newsgroups dataset.

Table 6: Experimental results on the 20-Newsgroups dataset.

Method	Category	Feature extraction	Recall	Precision	F1
SVM	COMP	IG	0.6555	0.7349	0.6929
		MI	0.5445	0.5700	0.5569
	REC	IG	0.6294	0.6624	0.6455
		MI	0.5153	0.5284	0.5217
	SCI	IG	0.6868	0.8192	0.7472
		MI	0.6093	0.7016	0.6522
	TALK	IG	0.5603	0.6284	0.5924
		MI	0.4654	0.4992	0.4817
RF	COMP	IG	0.6534	0.6816	0.6672
		MI	0.5609	0.6363	0.5962
	REC	IG	0.7180	0.8670	0.7855
		MI	0.5924	0.6186	0.6052
	SCI	IG	0.5904	0.6268	0.6081
		MI	0.5244	0.5713	0.5469
	TALK	IG	0.7536	0.7723	0.7628
		MI	0.6180	0.7087	0.6602
Proposed	COMP	IG	0.6923	0.7550	0.7223
		MI	0.5888	0.7105	0.6440
	REC	IG	0.6668	0.7271	0.6957
		MI	0.5471	0.6205	0.5815
	SCI	IG	0.7989	0.8315	0.8149
		MI	0.6561	0.7127	0.6832
	TALK	IG	0.6012	0.6266	0.6136
		MI	0.5341	0.6204	0.5740

Table 6 show that the method used in this paper has the best classification performance, regardless of whether it is based on IG or MI. The classification performance of RF is not stable. The performance of the SVM algorithm is stable, but the overall classification performance is lower than the method proposed in this paper.

(2) On the 20-Newsgroups dataset, the difference between the classification results obtained by the two feature extraction methods becomes smaller. However, the experimental results obtained by the IG-based feature extraction method are still better than the MI-based method.

(3) Compared with the classification results of the Reuters-21578 dataset, the classification results obtained on the 20-Newsgroups dataset are slightly worse. This shows that the experimental results obtained by the same classification method on different text datasets are not the same.

5. Conclusion

With the development of internationalization, the demand for literary translation has increased dramatically. In order to make better use of these literary translation texts, this paper studies the classification of literary translation texts. Because the traditional classification method only pays attention to the words and phrases of the composition, and only knows which words correspond to which application fields, words are relatively isolated from each other and do not consider other characteristics such as word collocation, style, and typesetting. Therefore, this paper proposes a literary translation text classification method based on the distributed incremental sequence data mining algorithm. The method can fully mine the features of syntactic order and other features based on considering the characteristics of composition words and phrasing in different application fields, to discover more effective information, so it can strengthen the text classification effect. The experimental results also support this. This method also has some shortcomings, such as the classification accuracy is still far from being used in real production scenarios, and the classification accuracy needs to be further improved. The future research will be mainly carried out from the following two aspects: First, we will continue to deeply study the characteristics of composition words and wording in different application fields and try to compile a relatively high-quality stop word list, to improve the efficiency of the classification of literary translation texts by this method. The second is to introduce more feature extraction methods to improve the classification performance. Because a good feature extraction method can greatly improve the text classification performance, which is also shown in the experiments.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the Zhejiang International Studies University.

References

[1] G. Fumera, I. Pillai, and F. Roli, "Spam filtering based on the analysis of text information embedded into images," *Journal of Machine Learning Research*, vol. 7, pp. 2699–2720, 2006.

[2] X. Qi and B. D. Davison, "Web page classification," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1–31, 2009.

[3] I. Anotonellis, C. Bouras, and V. Poulopoulos, "Personalized news categorization through scalable text classification," *Front. WWW Res. Dev-APWEB Lect. Notes. Comput. Sci.,* vol. 3841, pp. 391–401, 2006.

[4] B. Tang, H. He, P. M. Baggenstoss, and S. Kay, "A bayesian classification approach using class-specific features for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1602–1606, 2016.

[5] Y. Gao, Y. Xu, and Y. Li, "Pattern-based topics for document modelling in information filtering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 6, pp. 1629–1642, 2015.

[6] J. D'hondt, J. Vertommen, P.-A. Verhaegen, and D. J. R. Cattrysse, "Pairwise-adaptive dissimilarity measure for document clustering," *Information Sciences*, vol. 180, no. 12, pp. 2341–2358, 2010.

[7] Y. Ko, J. Park, and J. Seo, "Improving text categorization using the importance of sentences," *Information Processing & Management*, vol. 40, no. 1, pp. 65–79, 2004.

[8] G. Erkan and D. R. Radev, "LexRank: graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, no. 7, pp. 457–479, 2004.

[9] P. A. V. Hall and G. R. Dowling, "Approximate string matching," *ACM Computing Surveys*, vol. 12, no. 4, pp. 381–402, 1980.

[10] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[11] T. Joachims, "A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization," in *Proceedings of the 14th Int. Conf. Mach. Learn*, pp. 143–151, San Francisco, CA USA, July 1997.

[12] K. R. Kim, J. L. Kwon, J. S. Kim, Z. Kim, and H. G. Cheon, *European Journal of Pharmacology*, vol. 528, no. 1–3, pp. 37–42, 2005.

[13] M. G. Michie, "Use of the bray-curtis similarity measure in cluster analysis of foraminiferal data," *Journal of the International Association for Mathematical Geology*, vol. 14, no. 6, pp. 661–667, 1982.

[14] Z. Zhao, X. He, D. Cai, L. Zhang, W. Ng, and Y. Zhuang, "Graph regularized feature selection with data reconstruction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 689–700, 2016.

[15] Q. Qinbao Song, J. Jingjie Ni, and G. Guangtao Wang, "A fast clustering-based feature subset selection algorithm for high-dimensional data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 1–14, 2013.

[16] Y. F. Yuefeng Li and N. Ning Zhong, "Mining ontology for automatically acquiring Web user information needs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 554–568, 2006.

[17] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[18] A. Algarni and Y. Li, "Mining specific features for acquiring user information needs," *Advances in Knowledge Discovery and Data Mining*, vol. 7818, pp. 532–543, 2013.

[19] R. Agrawal, R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*, pp. 3–14, IEEE Xplore, Taipei, Taiwan, March 1995.

[20] S. Raj, D. Ramesh, and K. K. Sethi, "A Spark-based Apriori algorithm with reduced shuffle overhead," *The Journal of Supercomputing*, vol. 77, no. 1, pp. 133–151, 2020.

[21] T. Branco, D. J. Moura, I. A. Nääs, S. R. M. Oliveira, and S. R. M. Oliveira, "Detection of broiler heat stress by using the generalised sequential pattern algorithm," *Biosystems Engineering*, vol. 199, pp. 121–126, 2020.

[22] S. C. Pedduri, S. R. R. Ginnavaram, M. B. Myneni, and B. Padmaja, "Photorealistic image synthesis using spade algorithm," *Lecture Notes in Electrical Engineering*, vol. 664, pp. 377–389, 2020.

[23] P. Jian and H. Jiawei, B. Mortazavi-Asl, H. Pinto, and C. Qiming, "Mining sequential patterns by pattern-growth: the PrefixSpan approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1424–1440, 2004.

[24] S. Roy, M. Choudhury, R. Puri, and D. Z. Pan, "Towards optimal performance-area trade-off in adders by synthesis of parallel prefix structures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 10, pp. 1517–1530, 2014.

[25] A. Sinhmar, V. Malhotra, R. K. Yadav, and M. Kumar, "Spam detection using genetic algorithm optimized LSTM model," *Computer Networks and Inventive Communication Technologies*, vol. 75, pp. 59–72, 2022.

[26] C.-C. Chen, H.-H. Shuai, and M.-S. Chen, "Distributed and scalable sequential pattern mining through stream processing," *Knowledge and Information Systems*, vol. 53, no. 2, pp. 365–390, 2017.