Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6619088, 30 pages
https://doi.org/10.1155/2021/6619088

Research Article

Text Classification Using Novel Term Weighting Scheme-Based Improved TF-IDF for Internet Media Reports

Zhiying Jiang[1,2] **Bo Gao**[1,2] **Yanlin He**[1,2] **Yongming Han**[1,2] **Paul Doyle**[3] **and Qunxiong Zhu**[1,2]

[1]*College of Information Science & Technology, Beijing University of Chemical Technology, Beijing 100029, China*
[2]*Engineering Research Center of Intelligent PSE, Ministry of Education of China, Beijing 100029, China*
[3]*School of Computer Science within the College of Science and Health, Technological University Dublin, Dublin, Ireland*

Correspondence should be addressed to Qunxiong Zhu; zhuqx@mail.buct.edu.cn

Received 6 December 2020; Revised 18 January 2021; Accepted 29 January 2021; Published 5 March 2021

Academic Editor: Nianyin Zeng

With the rapid development of the internet technology, a large amount of internet text data can be obtained. The text classification (TC) technology plays a very important role in processing massive text data, but the accuracy of classification is directly affected by the performance of term weighting in TC. Due to the original design of information retrieval (IR), term frequency-inverse document frequency (TF-IDF) is not effective enough for TC, especially for processing text data with unbalanced distributions in internet media reports. Therefore, the variance between the DF value of a particular term and the average of all DFs ($\overline{\mathrm{DF}}$), namely, the document frequency variance (ADF), is proposed to enhance the ability in processing text data with unbalanced distribution. Then, the normal TF-IDF is modified by the proposed ADF for processing unbalanced text collection in four different ways, namely, TF-IADF, TF-IADF^{+}, TF-IADF$_{\mathrm{norm}}$, and TF-IADF$^{+}{}_{\mathrm{norm}}$. As a result, an effective model can be established for the TC task of internet media reports. A series of simulations have been carried out to evaluate the performance of the proposed methods. Compared with TF-IDF on state-of-the-art classification algorithms, the effectiveness and feasibility of the proposed methods are confirmed by simulation results.

1. Introduction

Due to the rapid development of internet technology and information infrastructure construction, the volume of text data which can be obtained online has increased dramatically. As the China Internet Network Information Center (CNNIC) stated, the number of netizens in China had increased to 828.51 million by the end of 2018 [1]. The internet has become the main channel for Chinese people to obtain information. The content of internet media is the most important data source, in which textual documents are the main one. And it is increasingly important to effectively analyze massive textual documents such as classification, indexing, and clustering. As a consequence, text classification (TC) is concerned by many researchers working in the field. Based on the previous studies, many applications based on TC technology have been developed, such as author identification [2, 3], spam e-mail filtering [4], medical documents' classification [5], management of customer relationship, and classification of web pages [6, 7]. Text classification (TC) is a task that assigns textual documents to predefined classes based on knowledge extracted from their content. The process of TC is as follows [8]:

(i) Given a set of k different discrete class label values $C = \{C_1, \ldots, C_k\}$ and training data and a set of documents $D = \{D_1, \ldots, D_n\}$, each document of which is labeled with a specific value from set C

(ii) Calculate text representations for documents in set D

(iii) Build a classification model based on the training data, which indicates the relationship between the

features in the underlying document and one of the classes

(iv) Predict class labels for class-unknown documents using the trained model

Calculating text representation, training classification models, and predicting class labels for class-unknown documents are the main steps of text classification. The entire steps, factors, and the way they organize in TC are shown in Figure 1.

As shown in Figure 1, before documents can be analyzed by a classification model, documents need to be preprocessed in a specific way such as be represented by vectors with numerical values. These values relate to predefined classes that the classification model can understand. This process is called text representation, and it is an essential prerequisite for TC tasks [9]. There are many methods proposed for text representation among which the vector space model (VSM) is the most commonly used one [10]. VSM is a feature vector that consists of numerical values which are also called term weights for representing a document. Components of this kind of model can be of different types such as words, sentences, and phrases [11]. These components are also called terms which are extracted from a document to form a bag of words (BOW) [12]. The abilities of these terms in distinguishing different documents are represented by numerical values (weights) related to the terms [13]. For example, a document can be represented as a vector of weighted features (or terms) $d_k = (t_1, t_2, \ldots, t_n)$ and a corresponding weight vector $w_k = (w_1, w_2, \ldots, w_n)$, where n is the number of selected features (terms) and $w_1, w_2, \ldots, w_n$ are the weights of $t_1, t_2, \ldots, t_n$. Then, a collection of documents (corpus) can be represented as shown in Figure 2, where element $w_{i,j}$ represents the weight of t_j from d_i.

As we can see in the matrix, each term in each document can only be assigned to one weight at the same time in VSM. It is obviously crucial to assign appropriate weights to terms for the performance of text classification. Therefore, many methods which are called term weighting scheme (TWS) are proposed to determine the weights for terms of documents. Different TWSs generate different vectors for the same document, thus attributing to the document with different representations. "Good" term weighting methods are of fundamental importance for guaranteeing good TC performance. So far, there are two main categories of TWSs in the literature: semantic-based TWSs and statistics-based TWSs [14].

The semantic-based TWSs focus on the semantic relationships between terms and documents which are hidden behind the extracted features (words) as well as focus on the meaning of words [15]. For example, Rao et al. proposed a new model based on a neural network which captures semantics of continuous text representations [16]. Based on the distributed hypothesis in which meanings of terms from documents with similar meanings will also be similar, a neural network was used to embed words into a continuous vector space (Word2Vec) for capturing the semantic information of words [17]. Doc2Vec, based on Word2Vec, was extended from the word level to the document level by fully using the information of the word sequence [18]. Because the above methods all have limitations when being used on their own, Kim et al. combined the BOW and Doc2Vec and proposed the bag-of-concepts method for overcoming the limitations [19], and Kim et al. also tried different methods, namely, TF-IDF, LDA, and Doc2Vec, for text representation [20]. More recently, Wu et al. proposed a novel phrase-based text representation called Phrase2Vec, which includes skip phrase, CBOP, and GloVeFP. They applied the novel method to text analysis research, and results show that Phrase2Vec can improve the performance of TC and clustering tasks [21]. Jaeyoung Kim et al. first applied capsule networks, which achieved success in image classification, in the TC task and demonstrated comparable performance to well-known schemes at the time [22].

For non-English languages, semantic analysis is also widely used for TC. Ye-wang Chen et al. proposed a novel method using the biggest open and free internet knowledge-based Baidu Baike to capture the semantic relationships of words to categories to enhance the performance of TC in Chinese text [23]. Ashraf Elnagar et al. introduced two new datasets for Arabic TC tasks, namely, SANAD and NADiA, both of which are freely available for research studies. In their experiments of extensive comparisons among several deep learning (DL) models for Arabic TC on SANAD and NADiA, their method outperformed others because of no requirement of a preprocessing stage and being completely based on deep learning models [24].

However, TWSs based on semantic analysis are more complex than statistical counterparts in analyzing and calculating the process. Furthermore, for semantic-based methods, performance cannot be significantly improved. Therefore, statistics-based TWSs are still major topics in the field of text classification [25]. Normally, most statistics-based TWSs all depend on the following philosophies:

(i) Terms with higher occurrences in a document relate to the document better, which is the basic idea of "term frequency" (TF)

(ii) Terms with occurrence in fewer documents relate to the documents where they occur better, which is the basic idea of "inverse document frequency" (IDF)

According to these principles, there are two main factors in a statistics-based TWS as it is shown in the following equation:

$$\text{TWS} = \text{term frequency factor} * \text{collection frequency factor}. \quad (1)$$

Many methods are proposed in the literature based on different implementations of equation (1). Some popular examples are shown in Table 1, where values of "NONE" indicate there is no corresponding method for the specific parameter. As the table shows, some methods focus on modifying the term frequency factor (i.e., LogTF-RF [11] and SQRT_TF-IGM [25]), while some focus on developing novel methods as the collection frequency factor (i.e., TF-IDF [26], TF-CHI2 [27], TF-IEF [14], and TF-IGM [25]). Nevertheless, TF-IDF is still one of the most preferred methods.

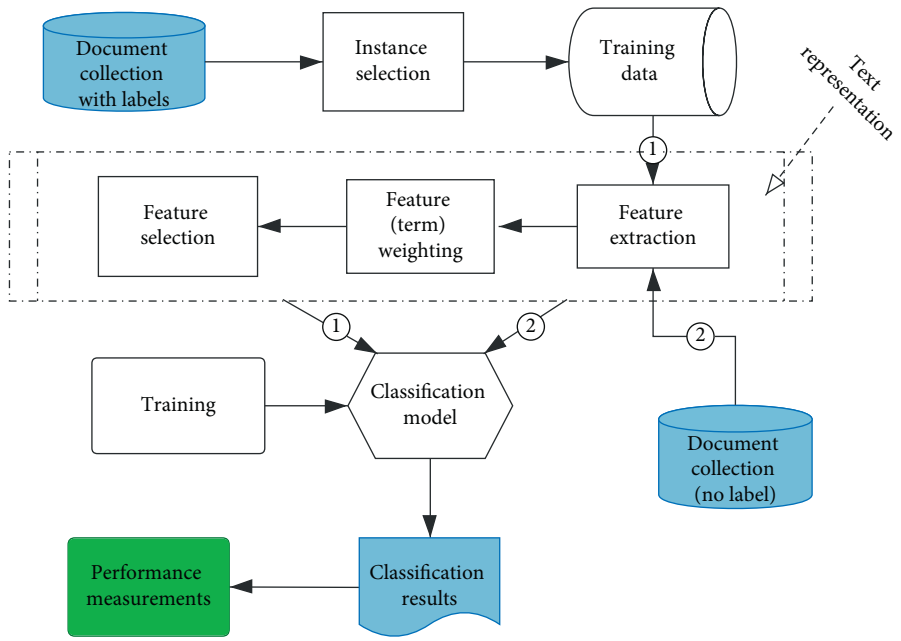


Figure 1: The entire steps, factors, and the way they organize in text classification.

$$\begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,j} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,j} \\ \vdots & \vdots & \vdots & \vdots \\ w_{i,1} & w_{i,2} & \cdots & w_{i,j} \end{bmatrix}$$

Figure 2: An example of the weight matrix for a collection of documents in the vector space model.

Table 1: Some popular examples of statistics-based term weighting schemes.

TWS	Term frequency factor	Collection frequency factor
TF	Term frequency (TF)	NONE
DF	NONE	Document frequency (DF)
TF-IDF	Term frequency (TF)	Inverse document frequency (IDF)
TF-CHI2	Term frequency (TF)	Globalized chi-square score (CHI2)
LogTF-RF	Logarithm of term frequency	Relevance frequency (RF)
TF-IGM	Term frequency (TF)	Inverse gravity moment (IGM)
SQRT_TF-IGM	Square root of term frequency	Inverse gravity moment (IGM)
TF-IEF	Term frequency (TF)	Inverse exponential frequency (IEF)

In this paper, enriching the collection frequency factor of statistics-based TWS is concerned, i.e., TF-IDF, for handling situations of imbalanced data distribution. A new formula is designed by using the variance between the DF of a specific term and the average value of all DFs ($\overline{DF}$) instead of original DF in TF-IDF. Based on the new formula, a novel method named TF-IADF and three other TWSs based on the same idea are proposed to enhance the TC performance in the imbalanced situation of internet media reports.

The remainder of this paper is organized as follows. An overview of background study on statistics-based term weighting schemes is given in Section 2. The main idea of our novel methods is described in Section 3. Section 4 briefly introduces the experimental settings and datasets, including data preprocessing, the classifiers, and the measurements. Experimental results and the analysis are presented in Section 5. The final conclusion is given in Section 6.

2. Background Study

When looking at the studies related to TWS in the literature, TF-IDF, originally designed for information retrieval (IR), may be at the top of the list. However, as Chen et al. stated, due to its original design, TF-IDF is not effective enough in the text classification domain [25]. Thus, they proposed a new statistics-based model named inverse gravity moment (IGM) to describe the inter-category distribution. Based on IGM, TF-IGM and sqrt_TF-IGM (RTF) are proposed. In their demonstration on popular classifiers, namely, SVM and kNN, the

proposed methods had better performance in measurements such as micro-F1 and macro-F1 than existing TWSs (TF, TF-IDF, TFIDF-ICSDF, TF-CHI, TF-PB, and TF-RF). However, Turgut Dogan et al. reviled that, for each case where the term document frequency changes, the term with the same weight is given by TF-IGM. This means terms with different distinguishing abilities obtain the same weights from the standard IGM method which is unreasonable [28]. In their studies, two novel TWSs, namely, SQRT_TF-IGM$_{imp}$ and TF-IGM$_{imp}$, are proposed deriving from IGM to overcome its limitations. In other aspects, Zhong Tang et al. described two deficiencies from which TF-IDF suffers, namely, collection frequency factor being undefined (division by zero) or being equal to zero in some special cases. They proposed a novel method, namely, term frequency-inverse exponential frequency (TF-IEF), to overcome these drawbacks [14]. The proposed methods replaced the IDF with a global weighting factor IEF, and a log-like method is used to characterize the collection frequency factor. It greatly reduced the influence caused by terms with high TF values, which helped in generating a more representative vector of terms. The experiments stated that the novel methods had an improved performance than compared schemes. The knowledge about Chinese language and Chinese culture provided by Baidu Baike is learned and organized by Chinese language-speaking people and professional employees of Baidu company. Therefore, Baidu Baike is used for optimizing TC on Chinese text a couple of times in the Chinese language aspect [23, 29]. However, both Baidu Baike-based methods are based on semantic analysis, and huge calculations are required for processing.

However, most of these methods are based on the assumption that the dataset is relatively balanced in distribution. In fact, the imbalanced distribution of the dataset occurs frequently in the TC domain [30]. Furthermore, the classification performance is heavily affected by the imbalanced distribution of the dataset in TC [31, 32]. Many studies have been proposed to address this problem, such as [33, 34]. In these proposed studies, two common ways are used to solve the problem of data imbalance, namely, the data-driven methods and the algorithm-driven methods. The data-driven method is to adjust the proportion of data categories by undersampling, oversampling, or a combination of undersampling and oversampling. The algorithm-driven method is to adjust the classification algorithm to achieve the effect of promoting learning without changing the dataset. The simulation results of these proposed methods show that the more unbalanced the proportion of categories is, the lower the overall performance of TC becomes. One of the main reasons is that some less-common terms in large-scale categories are weighted even higher than some more-common terms in small-scale categories due to their frequencies of occurrence.

Document classification of Chinese media reports on the internet which is also a TC problem with imbalanced dataset is researched in this paper. And a more representative model in cases of imbalance data is tried to create by modifying the term weighting method.

3. Novel Term Weighting Methods Based on Improved TF-IDF

TF-IDF is the most widely used TWS proposed by Karen Spärck Jones [26]. In this section, a new TWS based on TF-IDF, namely TF-IADF, and its variants proposed in this paper are described in specific.

3.1. Overview to TF-IDF. TF-IDF [35] is a combination of term frequency (TF) and inverse document frequency (IDF). Since the original value of term frequency in a document is used directly, the TF representation is one of the simplest TWSs. TF is based on the assumption that a term with a higher term frequency value is regarded to be more important than that with a lower term frequency value. It only depends on the number of occurrences of a specific term in a local document. Therefore, the capacity of TF for distinguishing all relevant documents from other irrelevant documents is very low due to its ignorance of collection frequency. To address this problem, the inverse document frequency (IDF) was proposed with a concern of collection frequency which enhanced the discriminative capacity of a term for text classification [36]. IDF extends from document frequency (DF) which means the number of documents where a term occurs. It is proposed based on the assumption that a term which occurs in fewer documents is regarded to be more important than that which occurs in more documents [11]. The IDF value of a specific term can be obtained as shown in the following:

$$\mathrm{IDF}(t,d,D) = \log\frac{|D|}{\mathrm{DF}(t,D)}. \tag{2}$$

In equation (2), $\mathrm{DF}(t,D)$ represents the DF value of term t in corpus D. The symbol in equation (2) represents the total number of documents in corpus D. To avoid infinity of some extreme cases, the formula is sometimes optimized as shown in the following:

$$\mathrm{IDF}(t,d,D) = \log\frac{|D|+1}{\mathrm{DF}(t,D)+1}. \tag{3}$$

After that, Jones extended the IDF method by adding the TF value into calculation [26]. The proposed combination with TF and IDF is the most well-known term weighting method, namely, TF-IDF. Similar with IDF, TF-IDF is also a global statistical measure. The classical structure of TF-IDF is shown as

$$\mathrm{TF}-\mathrm{IDF}(t,d,D) = \mathrm{TF}(t,d) * \mathrm{IDF}(t,d,D). \tag{4}$$

In equation (4), $\mathrm{TF}-\mathrm{IDF}(t,d,D)$ represents the weight of term t of document d in corpus D, while $\mathrm{TF}(t,d)$ represents the TF value of term t in document d.

As we introduced in Section 2, TF-IDF is not effective enough in the text classification domain due to its original

design. And many research studies have been deployed in optimizing term weighting methods based on TF-IDF from different perspectives. Some of them developed new methods replacing the term frequency factor or document frequency factor of TF-IDF, while some of them modified the existing method of TF-IDF. This paper focuses on modifying IDF to improve the TC performance, especially for Chinese internet media content.

3.2. Proposed Methods. When looking into the formula of calculating the value of IDF as shown in equations (2) and (3), we notice that when the corpus is not very balanced which means the size of different categories in a corpus varies from each other, terms from categories with larger size will be assigned smaller values than terms from other categories. This is obviously not in line with the real situation. Moreover, for some low document-frequency terms, the value of IDF is much higher than others even when those low document-frequency terms are meaningless, which is not in line with the true situation either. To address this kind of problems, we focus on the deviation of the DF value between a specific term and the overview of all terms in the whole corpus since when the deviation between the DF value of a specific term and the average of all DF values is large, its discriminative ability is weak. This factor should be considered in the term weighting process.

Definition 1 (average document frequency (ADF)). It is the variance between the DF value of a specific term and the average of all DF values in a corpus.

We modified the collection frequency factor by adding ADF into calculation to address the problems mentioned above. In this study, the average of all DF values in the corpus is represented as $\overline{\mathrm{DF}}$, while the ADF value of term t in document D is represented as $A_{\mathrm{DF}}(t, D)$. Equations (5) and (6) show how they are calculated, where n is the number of terms:

$$\overline{\mathrm{DF}} = \frac{\sum \mathrm{DF}(t, D)}{n}, \tag{5}$$

$$A_{\mathrm{DF}}(t, D) = \frac{(\mathrm{DF}(t, D) - \overline{\mathrm{DF}})^2}{n}. \tag{6}$$

As ADF is extended from DF, the simplest way of optimizing IDF is to replace DF by ADF in the formula. Then, we get a novel formula of collection frequency which is shown as follows:

$$\mathrm{IADF}(t, D) = \log \frac{|D| + 1}{A_{\mathrm{DF}}(t, D) + 1}. \tag{7}$$

In fact, the IDF method is successful enough in most cases; what we need to do is just to modify it for some extreme cases. Then, we get another novel formula as shown in equation (8), where ADF is used to reduce the weight of the terms with extremely high or extremely low DF value.

$$\mathrm{IADF}^{+}(t, D) = \log \frac{|D| + 1}{\mathrm{DF}(t, D) + 1} * \frac{1}{\log\left(A_{\mathrm{DF}}(t, D) + 1\right) + 1}. \tag{8}$$

The two ADF-based methods can improve the TC performance in some cases we mentioned. However, there are still limitations due to the variance itself that when the size is too large or too small, the variance will be relatively too small or too large. Extreme values for terms will obviously impact the TC performance. Therefore, we further optimized the formula by normalizing the ADF to reduce the effect caused by the extreme value of terms. First, $A_{\mathrm{DF}}(t, D)$ is modified as shown in equation (9) and then using the normalization formula as shown in equation (10):

$$A'_{\mathrm{DF}}(t, D) = \log \frac{1}{\left(A_{\mathrm{DF}}(t, D) + 1\right)} + 1, \tag{9}$$

$$A'_{\mathrm{DF}}(t, D) = \frac{A_{\mathrm{DF}}{}'(t, D) - \min\left(A_{\mathrm{DF}}{}'(t, D)\right)}{\max\left(A_{\mathrm{DF}}{}'(t, D)\right) - \min\left(A_{\mathrm{DF}}{}'(t, D)\right)}. \tag{10}$$

Based on $A_{\mathrm{DF}}{}''$, another two novel formulas are designed as shown in equations (11) and (12), where α (default value is 1) is used as an optional weight proportion to adjust the importance of $A_{\mathrm{DF}}{}''$ in different cases.

$$\mathrm{IADF}_{\mathrm{norm}}(t, D) = \log \frac{|D| + 1}{A_{\mathrm{DF}}{}''(t, D) + 1}, \tag{11}$$

$$\mathrm{IADF}^{+}_{\mathrm{norm}}(t, D) = \log \frac{|D| + 1}{\mathrm{DF}(t, D) + 1} * \left(A_{\mathrm{DF}}{}''(t, D) * \alpha\right). \tag{12}$$

Based on the above four proposed formulas of collection frequency based on IDF, we get four novel term weighting methods which are shown in equations (13)–(16):

$$\mathrm{TF\text{–}IADF}(t, d, D) = \mathrm{TF}(t, d) * \mathrm{IADF}(t, D), \tag{13}$$

$$\mathrm{TF\text{–}IADF}^{+}(t, d, D) = \mathrm{TF}(t, d) * \mathrm{IADF}^{+}(t, D), \tag{14}$$

$$\mathrm{TF\text{–}IADF}_{\mathrm{norm}}(t, d, D) = \mathrm{TF}(t, d) * \mathrm{IADF}_{\mathrm{norm}}(t, D), \tag{15}$$

$$\mathrm{TF\text{–}IADF}^{+}_{\mathrm{norm}}(t, d, D) = \mathrm{TF}(t, d) * \mathrm{IADF}^{+}_{\mathrm{norm}}(t, D). \tag{16}$$

As a result, the optimized text representation model of processing internet media reports is shown in Figure 3. Four new calculation formulas are used to replace IDF of TF-IDF. And four novel term weighting methods are obtained to enhance the performance of processing unbalanced text collection.

4. Case Study

To evaluate our proposed TWSs, experiments are carried out by using proposed methods in state-of-the-art classification algorithms on both Chinese and English corpuses. In this section, datasets used in experiments are briefly described. Then, algorithms utilized for the classification process and the measurements used for performance evaluation in this

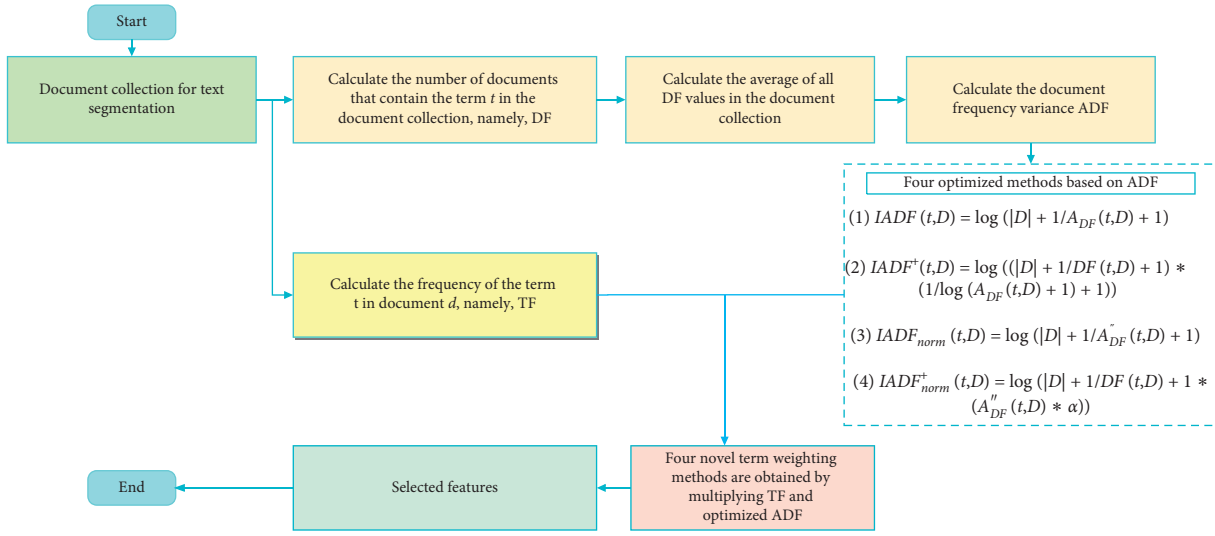


Figure 3: The flowchart of the text classification method based on the improved TF-IDF.

study are introduced. Finally, the experiment settings are also presented.

4.1. The Data Source. This study carried out experiments on three different datasets, i.e., standard dataset of English text, namely, Reuters-21578 corpus, classic dataset of Chinese text, namely, Fudan corpus, and a collection of Chinese internet media reports named Internet corpus, which were crawled off web and transformed into forms of Chinese textual document.

4.1.1. Reuters-21578 Corpus. The Reuters-21578 corpus contains top-10 categories of Reuters-ModApte separately split which is most preferred in the TC domain [37]. In this study, multilabeled samples are removed since single-label-classification is focused. So, only 8 categories of 5607 training samples and 2270 test samples in Reuters-21578 were used in our experiments. The detail of data distribution of this corpus is shown in Figure 4 and Table 2.

4.1.2. Fudan Corpus. The Fudan University TC corpus is from the Chinese NLP group in Department of Computer Information and Technology, Fudan University of China. There are 20 categories of which the data distributions are shown in Figure 5 and Table 3. Similar to Reuters-21578, Fudan corpus is also an unbalanced dataset, but in Chinese language.

4.1.3. Internet Corpus. To test the performance of our proposed methods on Chinese internet media reports, some reports from the web are crawled, and this corpus is formed. There are six categories, namely, sport, education, tourism, traffic, tech, finance, and food. The data format is shown in Figure 6. There are three parts in each instance of the test data which are the category index, the article content, and

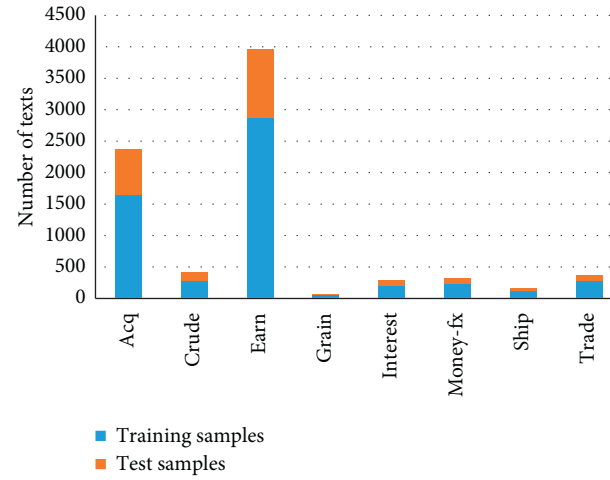


Figure 4: Data distribution of Reuters-21578 (unbalanced).

Table 2: Reuters-21578 corpus.

No.	Class label	Training samples	Testing samples
1	Acq	1634	728
2	Crude	277	131
3	Earn	2859	1086
4	Grain	41	10
5	Interest	198	87
6	Money-fx	214	93
7	Ship	115	43
8	Trade	269	92

the total number of words. In this study, both balanced and unbalanced data distributions of this corpus are tried.

4.2. Classification Algorithms Used for Experiments and Measurements. Three popular classification algorithms, namely, naïve Bayes (NB), support vector machine (SVM), and random forests (RF), are utilized using our proposed methods and existing methods for a brief comparison.

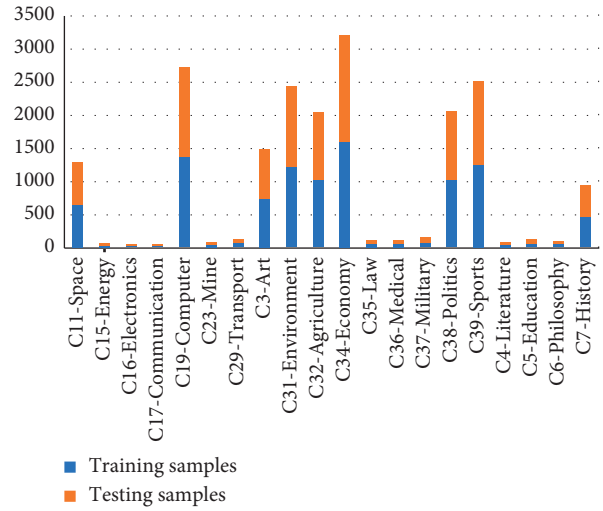


Figure 5: Data distribution of Fudan corpus (unbalanced).

Table 3: Fudan corpus.

No.	Class label	Training samples	Testing samples
1	C11-Space	640	642
2	C15-Energy	32	33
3	C16-Electronics	27	28
4	C17-Communication	25	27
5	C19-Computer	1357	1358
6	C23-Mine	33	34
7	C29-Transport	57	59
8	C3-Art	740	742
9	C31-Enviornment	1217	1218
10	C32-Agriculture	1021	1022
11	C34-Economy	1600	1601
12	C35-Law	51	52
13	C36-Medical	51	53
14	C37-Military	74	76
15	C38-Politics	1024	1026
16	C39-Sports	1253	1254
17	C4-Literature	33	34
18	C5-Education	59	61
19	C6-Philosophy	44	45
20	C7-History	466	468

,分类,文章,字数
0,娱乐,《青蛇》造型师默认新《红楼梦》额妆抄袭〔图〕凡是看过电影《青蛇》的人，都不会忘记青白二蛇的经典造型:飘逸的身材，妩媚的妆容，而最独特的就是取材于京剧女角头型的精华之处"打片子"(也被称之为额妆)……由于吴宝玲在香港出道很早，所以无论是对造型设计本身还是对文化的钻研，都有独到之处，而且往往能把两者结合得很好，她甚至感觉新版《红楼梦》的设计缺乏文学性，"《红楼梦》本身就很让人很期待，假如我来设计造型的话，因为片中将出现很多生活场景，我可能会尽量做得不这么沉重，而且人物角色的感觉应该有所区分，薛宝钗就是薛宝钗，而不会分不清楚她和林黛玉。"最后吴宝玲表示，她还会观察一段时间，看后期的设计是否真的完全会对《青蛇》有所模仿，必要的时候她可能向法律寻找帮助。昨晚，记者致电新《红楼梦》导演李少红，但她的手机一直无人接听。(责任编辑: CL) .1519

Figure 6: Style of test data.

Introductions about these algorithms and measurements for evaluation are given as follows.

4.2.1. Naïve Bayes. Naïve Bayes algorithm [38] is a well-known TC classifier based on Bayes' assumption that the features are regarded to be independent from each other. In the TC process, document d_k can be represented as a vector of terms $(t_1, t_2, \ldots, t_n)$. The probability that d_k belongs to a specific category c_i can be calculated using equation (17). More details about the NB classifier can be accessed in [39]. In this study, a NB classifier is used for evaluating the text weighting performance.

$$P(c_i|d_k) = P(c_i)\frac{\prod_{j=1}^{n} P(t_{jk}|c_i)}{P(d_k).} \tag{17}$$

4.2.2. Support Vector Machine. SVM [40, 41] is one of the most preferred algorithms for TC and many other patter recognition problems. Since it is a learning algorithm, it can handle problems with high dimensions well. The main principle of the SVM is to create linear or nonlinear hyperplanes to separate positive and negative samples. SVM uses some samples in the training set (called support vectors) to create hyperplanes at locations maximizing margins between negative and positive samples. In this study, a classic SVM classifier is used for evaluating the text weighting performance.

4.2.3. Random Forests. The random forest (RF) algorithm [42] is a parallelizable integration method which is one of the most preferred classifiers in the field of TC [43]. RF is composed of multiple decision trees. It is used to build a forest in a random way, which consists of many decision trees (DT). There is no correlation between each decision tree in the RF. After the RF is obtained, for each sample input, each decision tree in the forest is judged to see which category this sample belongs to, which category ultimately gets the most results, and which type of input prediction is. The architecture of RF is shown in Figure 7, where DT refers to the decision tree. In this study, a RF classifier is used for evaluating the text weighting performance.

4.2.4. Measurements for TC Performance. To evaluate the classification performance on the aforementioned datasets, accuracy, precision, recall, and F_1 score are calculated for validation according to equations (18)–(21), where c_i refers to a predefined category, while $\mathrm{TP}(c_i)$ refers to the number of documents which belong to c_i resulting in c_i, $\mathrm{FN}(c_i)$ refers to the number of documents which do not belong to c_i resulting in c_i, $\mathrm{FP}(c_i)$ refers to the number of documents which belong to c_i not resulting in c_i, and $\mathrm{TN}(c_i)$ refers to the number of documents which do not belong to c_i not resulting in c_i. The relationship between them and the classification result are shown in Table 4.

Table 4: Meanings of TP, TN, FP, and FN.

An instance of documents in the corpus (d_k)		
Sample d_k	Result in c_i	Not a result in c_i
Belongs to c_i	TP (c_i)	FP (c_i)
Does not belong to c_i	FN (c_i)	TN (c_i)

$$\mathrm{Accuracy}(c_i) = \frac{\mathrm{TP}(c_i) + \mathrm{TN}(c_i)}{\mathrm{TP}(c_i) + \mathrm{TN}(c_i) + \mathrm{FP}(c_i) + \mathrm{FN}(c_i)}, \tag{18}$$

$$\mathrm{precision}(c_i) = \frac{\mathrm{TP}(c_i)}{\mathrm{TP}(c_i) + \mathrm{FP}(c_i)}, \tag{19}$$

$$\mathrm{recall}(c_i) = \frac{\mathrm{TP}(c_i)}{\mathrm{TP}(c_i) + \mathrm{FN}(c_i)}, \tag{20}$$

$$F_1(c_i) = \frac{2 * \mathrm{precision}(c_i) * \mathrm{recall}(c_i)}{\mathrm{precision}(c_i) + \mathrm{recall}(c_i)}. \tag{21}$$

In multiclass classification problems, the overall performance can be measured by averaging the evaluation methods. Microaverage and macroaverage are used widely for this purpose. In this study, the microaveraged $F1$ (micro-$F1$) and macroaveraged $F1$ (macro-$F1$) measurements are also calculated to evaluate the experimental methods. The definition of macro-$F1$ and micro-$F1$ is as shown in equations (22) and (23):

$$\mathrm{macro} - F1 = \frac{1}{m}\sum_{i=1}^{m} F_1(c_i), \tag{22}$$

$$\mathrm{micro} - F1 = \frac{2 * \sum_{i=1}^{m} \mathrm{TP}(c_i)}{2 * \sum_{i=1}^{m} \mathrm{TP}(c_i) + \sum_{i=1}^{m} \mathrm{FP}(c_i) + \sum_{i=1}^{m} \mathrm{FN}(c_i)}. \tag{23}$$

In cases of unbalanced distribution, it is better to use micro-$F1$ than macro-$F1$ since the data size of categories is not considered in micro-$F1$ score calculation.

4.3. Experiment Settings. In this study, we carried out three experiments on the aforementioned datasets. All experiments were implemented on a 64 bit Windows 10 computer with 8 GB internal storage. The experimental code was written in Python language using Scikit-learn (sk-learn). sk-learn is a commonly used third-party module in machine learning which encapsulates many commonly used machine learning algorithms such as regression, dimension reduction, clustering [44], and classification. For each dataset, in preprocessing, term weighting, and term extraction, term representation and classification were utilized. In preprocessing, all documents were segmented into words by the open-source tool Jieba, and stop words were removed in this process. After that, a vector space model (VSM) was used for term representation using words as terms. Term weighting methods including TF-IDF and proposed methods, i.e., TF-IADF, TF-IADF$_{\mathrm{norm}}$, TF-IADF^{+}, and TF-IADF$^{+}_{\mathrm{norm}}$, were used here to form a final representation for each document. Finally, NB classifier, SVM classifier, and RF classifier were utilized for TC purpose. Combinations of different term

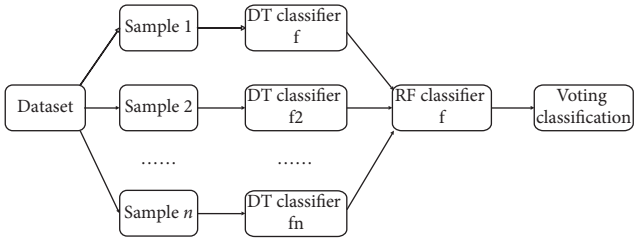


Figure 7: The architecture of random forests.

weighting methods and different classification algorithms on different datasets were compared for a brief analysis.

5. Results' Analysis and Discussion

5.1. Results on the Internet Corpus. In this part, the experiments on both a balanced dataset and an unbalanced dataset of internet media reports in the Chinese language are carried out.

5.1.1. Balanced Dataset. In this dataset, there are 1000 training samples and 1000 test samples in each category. This is an exactly balanced dataset. SVM, NB, and RF classifiers are utilized as TWS for a comparison with TF-IDF. The overall performance of the proposed TF-IADF outperformed all other methods in SVM and RF classifiers as shown in Figure 8. The details of experimental results are shown in Table 5 in that the proposed TF-IADF$^{+}_{\text{norm}}$ demonstrates better performance than TF-IDF in all cases. Furthermore, all proposed methods outperformed the TF-IDF, in some cases, respectively.

For the SVM classifier, the overall classification effect is better than the other two classifiers, and the micro-$F1$ value is over 94%. TF-IADF has achieved the best effect of 94.45% (increased by 0.31% than TF-IDF). For the RF classifier, TF-IADF achieves the best effect. In addition, TF-IADF$_{\text{norm}}$ and TF-IADF$^{+}_{\text{norm}}$ also come with some improvement. The micro-$F1$ value of TF-IDF is 84.51%, while that of TF-IADF$^{+}_{\text{norm}}$ is 85.52%, and that of TF-IADF$_{\text{norm}}$ is 85.26%. The micro-$F1$ value of TF-IADF reaches to 86.37%, which is the best among all methods, an improvement of 1.86% from TF-IDF. For the NB classifier, both TF-IADF^{+} and TF-IADF$^{+}_{\text{norm}}$ show improvements for the micro-$F1$ value of TF-IADF^{+} and TF-IADF$^{+}_{\text{norm}}$ reaching 92.26% and 92.33%, while that of TF-IDF is only 92.13%. TF-IADF$^{+}_{\text{norm}}$ achieves the best effect with an increase of 1.2% over TF-IDF. In the classification results, we notice that the precision rate and recall rate of finance are relatively low, at only 80% and 85%, respectively. The reason may be that the terms of this category are not obvious enough, and the scope involved is relatively wide, which may cover some contents from tourism and traffic categories, resulting in the poor classification effect of the whole category.

The results show that, in the case of balanced dataset which is not focused by our design, the proposed methods can improve the classifiers to a certain extent even though the improvement range is not so obvious.

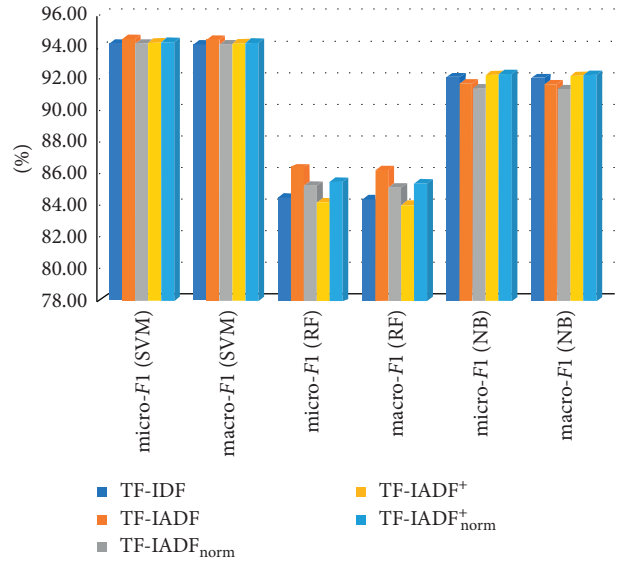


Figure 8: Data distribution of the Fudan corpus (unbalanced).

5.1.2. Unbalanced Dataset. In this section, we investigate how TC performance is impacted by unbalanced datasets. Dataset here is specially designed. The size of food is increased to 5000, and the sizes of other categories are kept the same as in the balanced dataset (1000) except sport. For sport, the size is gradually reduced from 400 to 50, decreasing 50 at a time.

Definition 2 (balance ratio). It is the proportion between the sizes of the category with the smallest size and the category with the largest size.

In this section, we call the proportion between the size of sport and that of food as balance ratio which drops from 8% to 1% with the decrease of sport's size. Experiments by using the SVM classifier with TF-IDF and our proposed methods are carried out. The experimental results show that performances on categories other than sport and food are basically the same with the balance ratio changing. However, the recall and $F1$ score of sport and the precision of food are impacted heavily by decreasing the balance ratio. The details are shown in Tables 6–8. As shown, the balance ratio is decreasing, with the recall and $F1$ score of sport also decreasing in all methods. However, the proposed TF-IADF and TF-IADF$_{\text{norm}}$ outperformed TF-IDF in all cases. Especially, when the balance ratio decreased to an extreme value (1%), TF-IADF$_{\text{norm}}$ came with a recall of 0.7168 which is almost 170% of that of TF-IDF. Due to the balance ratio changes, precision of the category with a relatively large size (food) was also impacted which can be seen from Table 8.

Definition 3 (decline ratio). It is the ratio of the current value to the initial value in a declining trend.

To make the relationship between the balance ratio and the performance clear, the decline ratio is calculated. It is the ratio of the value for each balance ratio to its initial value (value at 8%). This can be seen in Figure 9, where the ordinate refers to the decline ratio of the corresponding

Table 5: Experiments on the balanced dataset of the internet corpus.

Balanced dataset	TF-IDF (%)	TF-IADF (%)	TF-IADF$_{norm}$ (%)	TF-IADF^{+} (%)	TF-IADF$^{+}{}_{norm}$ (%)
micro-*F*1 (SVM)	94.23	**94.54**	94.23	94.32	94.35
macro-*F*1 (SVM)	94.16	**94.48**	94.18	94.25	94.28
micro-*F*1 (RF)	84.51	**86.37**	85.26	84.23	85.52
macro-*F*1 (RF)	84.41	**86.26**	85.16	84.08	85.41
micro-*F*1 (NB)	92.13	91.73	91.42	92.26	**92.33**
macro-*F*1 (NB)	92.05	91.66	91.36	92.18	**92.24**

Table 6: Recall (sport).

Balance ratio (%)	TF-IDF	TF-IADF	TF-IADF$_{norm}$	TF-IADF^{+}	TF-IADF$^{+}{}_{norm}$
8	0.9479	**0.9581**	**0.9581**	0.9315	0.9162
7	0.9376	0.9458	**0.9479**	0.8988	0.8886
6	0.9182	0.9397	**0.9417**	0.8824	0.8691
5	0.9018	0.9346	**0.9407**	0.8650	0.8507
4	0.8763	0.9243	**0.9315**	0.8262	0.8129
3	0.8272	0.8937	**0.9141**	0.7464	0.7301
2	0.7423	0.8252	**0.8507**	0.6012	0.5910
1	*0.4274*	0.6626	***0.7168***	0.2567	0.2413

Table 7: *F*1 score (sport).

Balance ratio (%)	TF-IDF	TF-IADF	TF-IADF$_{norm}$	TF-IADF^{+}	TF-IADF$^{+}{}_{norm}$
8	0.9732	**0.9786**	0.9776	0.9645	0.9562
7	0.9678	0.9722	**0.9732**	0.9467	0.9410
6	0.9574	0.9689	**0.9700**	0.9375	0.9300
5	0.9484	0.9662	**0.9694**	0.9276	0.9193
4	0.9341	0.9607	**0.9645**	0.9048	0.8968
3	0.9054	0.9438	**0.9551**	0.8548	0.8440
2	0.8521	0.9042	**0.9193**	0.7510	0.7429
1	0.5989	0.7971	**0.8350**	0.4085	0.3888

Table 8: Precision (food).

Balance ratio (%)	TF-IDF	TF-IADF	TF-IADF$_{norm}$	TF-IADF^{+}	TF-IADF$^{+}{}_{norm}$
8	0.8726	0.9185	**0.9305**	0.8310	0.7788
7	0.8711	0.9160	**0.9288**	0.8241	0.7716
6	0.8688	0.9160	**0.9296**	0.8213	0.7668
5	0.8650	0.9143	**0.9296**	0.8146	0.7580
4	0.8583	0.9126	**0.9279**	0.8041	0.7444
3	0.8480	0.9101	**0.9253**	0.7875	0.7170
2	0.8275	0.8849	**0.9092**	0.7517	0.6728
1	0.7813	0.8725	**0.8976**	0.6728	0.5691

performance factor. As it is shown, with the balance ratio decreasing, the growth rate of decline ratio becomes faster and faster. It is obvious that, for datasets with extreme categories such as food (relatively too large) and sport (relatively too small), performances of TF-IADF and TF-IADF$_{norm}$ are more stable than TF-IDF.

The overall performance on this dataset is impacted by the balance ratio also. The details of micro-*F*1 and macro-*F*1 are shown in Tables 9 and 10, respectively. The proposed TF-IADF and TF-IADF$_{norm}$ outperformed TF-IDF in all cases. To make it clear how balance ratio impacts the performance, we calculated the decline ratio which is shown in Figure 10. As it is shown, with the decrease of balance ratio, both micro-*F*1 and macro-*F*1 decrease gradually. For example, when looking at micro-*F*1, TF-IADF$_{norm}$ was with a decrease of just 3.65%, while TF-IDF was with a decrease of 8.06% which is more than twice of that of the proposed TF-IADF$_{norm}$, meaning the performance of TF-IADF$_{norm}$ is much more stable than that of TF-IDF in this dataset.

Even though TF-IADF^{+} and its variance do not achieve improvement in this experiment, TF-IADF and its variance outperformed TF-IDF significantly. Furthermore, it can be seen from Figure 10 that our proposed TF-IADF and TF-IADF$_{norm}$ are not only numerically better but also more stable than TF-IDF. Therefore, considering ADF in the term weighting method can actually improve the performance of text classification considerably in unbalanced cases.

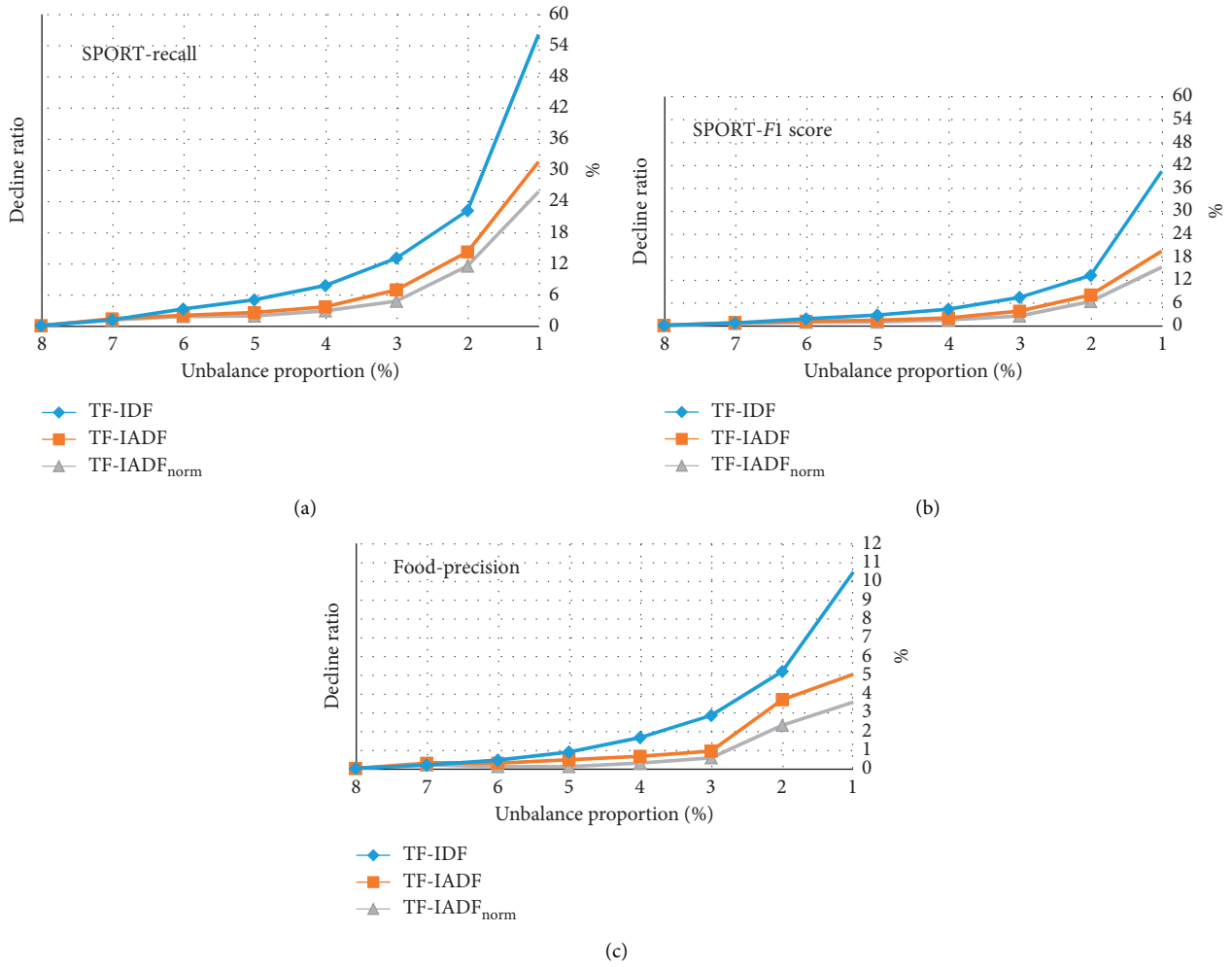


Figure 9: The decline ratio of the performance with balance ratio changing.

Table 9: Micro-$F1$.

Balance ratio (%)	TF-IDF	TF-IADF	TF-IADF$_{norm}$	TF-IADF^{+}	TF-IADF$^{+}{}_{norm}$
8	0.9292	**0.9372**	**0.9372**	0.9226	0.9140
7	0.9278	0.9354	**0.9360**	0.9179	0.9094
6	0.9250	0.9345	**0.9352**	0.9154	0.9063
5	0.9226	0.9338	**0.9351**	0.9129	0.9032
4	0.9189	0.9323	**0.9338**	0.9073	0.8977
3	0.9119	0.9279	**0.9313**	0.8960	0.8859
2	0.8994	0.9179	**0.9222**	0.8753	0.8658
1	0.8542	0.8946	**0.9029**	0.8258	0.8156

Table 10: Macro-$F1$.

Balance ratio (%)	TF-IDF	TF-IADF	TF-IADF$_{norm}$	TF-IADF^{+}	TF-IADF$^{+}{}_{norm}$
8	0.9287	**0.9366**	**0.9366**	0.9225	0.9147
7	0.9273	0.9349	**0.9355**	0.9179	0.9102
6	0.9246	0.9341	**0.9348**	0.9154	0.9072
5	0.9223	0.9333	**0.9346**	0.9130	0.9042
4	0.9186	0.9319	**0.9333**	0.9074	0.8987
3	0.9115	0.9275	**0.9309**	0.8956	0.8868
2	0.8985	0.9174	**0.9217**	0.8726	0.8649
1	0.8443	0.8922	**0.9014**	0.8055	**0.7987**

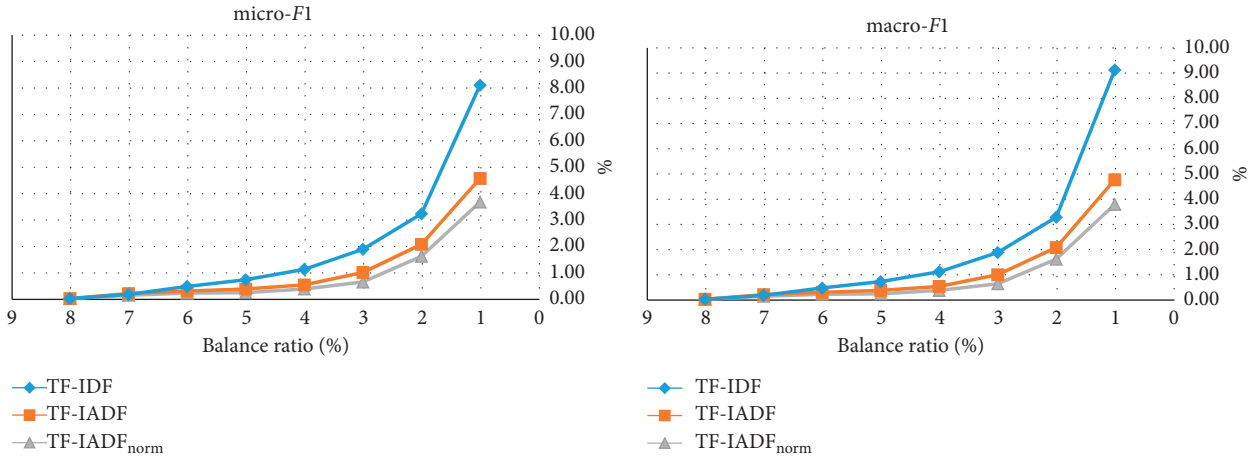


Figure 10: The decline ratio of the overall performance with balance ratio changing.

5.1.3. Analysis. For this corpus, the TF-IADF^{+} and TF-IADF$^{+}_{norm}$ methods are more suitable for the NB classifier, and TF-IADF$^{+}_{norm}$ is the best. For the RF and SVM classifiers, TF-IADF$^{+}_{norm}$ and TF-IADF are more suitable. In addition, TF-IADF is better in case of a balanced dataset, while TF-IADF$_{norm}$ is better in case of an unbalanced dataset. This means that the processed TF-IADF$_{norm}$ method is more sensitive and stable in the case of unbalanced datasets, while TF-IADF improves the classification effect more when the datasets are relatively balanced. For several mathematical models proposed by different algorithms, it can be concluded that the formula suitable for different algorithms may be different, and for the improvement of the corresponding algorithm effect, it can also be concluded that the ADF index has improved the effect of text classification, which confirms the conjecture; especially, when the dataset is not evenly distributed, the effect of text classification is more stable.

5.2. Results on the Fudan Corpus

5.2.1. Results' Analysis. Table 11 and Figure 11 show the micro-*F*1 and macro-*F*1 scores obtained on Fudan corpus using SVM, RF, and NB algorithms with different TWSs, while Tables 12–14 show the detailed results.

For the NB classifier, TF-IADF$^{+}_{norm}$ has the best performance, which is 89.38%, an increase of 0.39% compared to TF-IDF. Meanwhile, TF-IADF^{+} also gets an increase of 0.32%. This means for Chinese text datasets, these two models are more suitable for a NB classifier and can achieve an improvement on the overall classification effect. When comparing the specific measurements which are shown in Figure 12 and more details can be seen in Table 14, TF-IADF$^{+}_{norm}$ has achieved a lot of the highest performance-score items. Furthermore, the difference between performance scores of TF-IDF and those of TF-IADF$^{+}_{norm}$ is relatively small in categories, where TF- IADF$^{+}_{norm}$ is not as good as TF-IDF. For example, in C32 and C35, where TF-IDF achieves the best precision score, the difference between the precision score of TF-IDF and that of TF-IADF$^{+}_{norm}$ is less than 1%. However, in categories such as C17, where TF-IADF$^{+}_{norm}$ achieves the highest precision score, the difference between that and precision score of TF-IDF is more than 6% which is six times of the difference occurring in categories where TF-IDF achieves the higher score. In fact, in C17, when comparing TF-IADF$^{+}_{norm}$ to TF-IDF, the precision is improved by 6.62%, and the recall score of C17 is also increased by 7.41%, which is obvious. In an overall view of the *F*1 scores, it can be noticed that, in all the 20 categories except C35 and C5, the scores of TF-IADF$^{+}_{norm}$ are not lower than those of TF-IDF. Especially, in cases of TF-IADF$^{+}_{norm}$ with higher elevation such as C16, the *F*1 score is increased by nearly 10%.

For the RF classifier, the micro-*F*1 score obtained by TF-IADF is 81.95%, which is 1.63% higher than that obtained by TF-IDF. Meanwhile, TF-IADF^{+} and TF-IADF$_{norm}$ also have achieved improvements in different extents. It is known that RF algorithm has certain randomness, but our proposed TF-IADF method is more stable and has achieved better results which can be seen in Figure 13. When looking into detailed results as shown in Table 13, in some categories, such as C29 and C5, the precision score has been improved by 25.98% and 30.16%, respectively, and the recall score has remained basically unchanged. For the performance of *F*1 score, in some categories, such as C16 and C36, the score obtained by TF-IADF is about 10% higher than that obtained by TF-IDF. Furthermore, there are only three categories where *F*1 of TF-IADF is not as good as TF-IDF.

For the SVM classifier, it is still the case that TF-IADF and TF-IADF$_{norm}$ have achieved improved performances. When comparing with TF-IDF in the micro-*F*1 score, TF-IADF has increased by 0.73%, while TF-IADF$_{norm}$ has increased by 0.63%. It can be seen in Figure 14 that, for the SVM classifier, these two improved methods are more stable and have achieved some improvements. As shown in Table 12, there are many categories where the precision scores are very high, even up to 1. It can easily be seen in Figure 14

Table 11: Overall performances on the Fudan corpus.

Fudan corpus	TF-IDF	TF-IADF	TF-IADF$_{norm}$	TF-IADF^{+} (%)	TF-IADF$^{+}_{norm}$ (%)
micro-*F*1 (SVM)	50.02	**54.57**	54.93	48.85	47.90
macro-*F*1 (SVM)	90.31	**91.04**	90.94	90.24	89.82
micro-*F*1 (RF)	48.57	**51.43**	48.80	49.78	51.64
macro-*F*1 (RF)	80.32	**81.95**	80.83	81.20	79.85
micro-*F*1 (NB)	65.44	62.28	59.94	66.39	**66.97**
macro-*F*1 (NB)	88.99	88.20	87.55	89.31	**89.38**

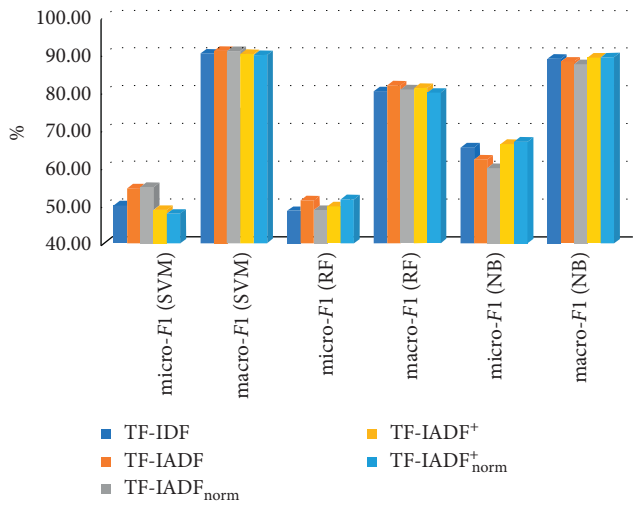


Figure 11: Overall performances of classification on the Fudan corpus.

that the size (data amount) of those categories is very small. For example, there are only 32 training samples in C15. The reason is that due to the small eigenvalues, no other categories being considered as this category are responsible for the high precision score. However, it can also be concluded from the detailed results that, for most of those categories with a high precision but a small size, the recall score is significantly reduced; that is to say, many documents are assigned to the wrong categories, which may be due to the impact caused by a small number of features. For example, the recall score of C15 is only 0.09. TF-IADF and TF-IADF$_{norm}$ perform better in most of the categories, especially in C29, where the size is small. Although the precision scores obtained by the two methods are decreased by about 4%, the recall scores are improved by about 30% which is significant. In addition, there are only two categories where *F*1 scores of TF-IADF are lower than those of TF-IDF, while for most categories, TF-IADF is better.

5.2.2. Further Discussion. Due to the unbalanced distribution of this corpus, all categories can be roughly divided into three groups by their size for specific analysis. The first group refers to categories with size in the range of 0 to 100, the second refers to categories with size in the range of 101 to 1000, and the third group refers to categories with size over 1000. First of all, for the first group, due to the insufficient size of training data, the same property is reflected in all term weighting methods, i.e., high precision and low recall scores, such as C16. Taking SVM combined with TF-IDF as an example, the precision score reaches 100%, while the recall score is only 3.57%, resulting in poor classification effect. In short, it assigns a few samples correctly, while a large number of test samples are assigned to the wrong categories. And the performance is similar in the RF and NB classifiers. This is the impact caused by the unbalanced distribution and the small size of the training data. For the second group, the number of training sets has been improved to a higher level. For example, in C11 when using the SVM classifier with TF-IDF, the precision score and recall score are 96.10% and 92.06%, respectively, which is a significant improvement compared to the first group. For the third category, such as C19, the precision score and recall score are 95.64% and 98.53%, respectively. It can be concluded that the precision score in categories with large training set size is relatively low, while the recall rate is relatively high. In categories with a small training set size, it will have better precision but very low recall score. In these kinds of conditions, our proposed methods will demonstrate a similar but more stable performance compared to TF-IDF. Taking C29 as an example, there are only 57 training samples. Comparing TF-IDF with TF-IADF using the SVM classifier, the precision scores are 100% to 96.15%, while the recall scores are 8.48% to 42.37%. It can be seen that although the accuracy rate of our algorithm is slightly reduced, the recall rate is greatly increased by nearly 500%, and the *F*1 score is also greatly improved, from 15.63% to 58.82%, showing a very obvious improvement.

In the Fudan corpus, a similar phenomenon with the internet corpus can be seen. For example, the TF-IADF$^{+}_{norm}$ method is the best one in the NB classifier, while TF-IADF^{+} is also better than TF-IDF. And with RF and SVM classifiers, both TF-IADF and TF-IADF$_{norm}$ achieve a relatively stable performance. The difference is that TF-IADF achieves the best effect in the Fudan unbalanced dataset. It is also noticed that although the micro-*F*1 score of the SVM classifier is the best, the macro-*F*1 score, which is about 50%, is not as good as that of the NB classifier which is over 65%. That is to say, although the overall accuracy of the SVM classifier is high, the effect of the NB classifier is better when each category is regarded as equally important.

5.3. Results on Reuters-21578. The overall performance on Reuters-21758 of all methods is shown in Table 15, while Figure 15 shows a brief comparison between all proposed

Table 12: Performance on the Fudan corpus using the SVM classifier.

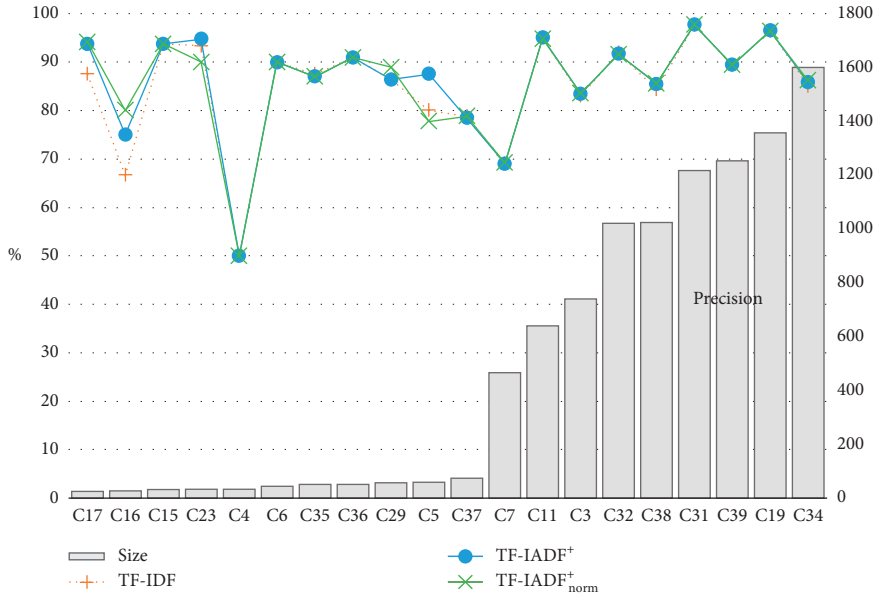
SVM	Precision					*F*1 score					Recall				
	TF-IDF	TF-IADF	TF-$IADF_{norm}$	TF-$IADF^{+}$	TF-$IADF^{+}{}_{norm}$	TF-IDF	TF-IADF	TF-$IADF_{norm}$	TF-$IADF^{+}$	TF-$IADF^{+}{}_{norm}$	TF-IDF	TF-IADF	TF-$IADF_{norm}$	TF-$IADF^{+}$	TF-$IADF^{+}{}_{norm}$
C11	96.10	95.98	96.13	**96.24%**	**96.24%**	92.06	**92.99**	92.84	91.75%	91.75%	94.03	**94.46**	94.45	93.94%	93.94%
C15	**100.00**	**100.00**	**100.00**	**100.00%**	**100.00%**	**9.09**	**9.09**	**9.09**	**9.09%**	**9.09%**	**16.67**	**16.67**	**16.67**	**16.67%**	**16.67%**
C16	**100.00**	**100.00**	**100.00**	**100.00%**	**100.00%**	**3.57**	**3.57**	**3.57**	**3.57%**	**3.57%**	**6.90**	**6.90**	**6.90**	**6.90%**	**6.90%**
C17	**100.00**	**100.00**	**100.00**	**100.00%**	**100.00%**	18.52	**22.22**	**22.22**	18.52%	18.52%	31.25	**36.36**	**36.36**	31.25%	31.25%
C19	95.64	**95.93**	95.53	95.50%	95.56%	98.53	98.97	**99.12**	98.53%	98.31%	97.06	**97.43**	97.29	96.99%	96.92%
C23	**100.00**	**100.00**	**100.00**	**100.00%**	**100.00%**	**5.88**	**5.88**	**5.88**	**5.88%**	**5.88%**	**11.11**	**11.11**	**11.11**	**11.11%**	**11.11%**
C29	**100.00**	96.15	95.65	**100.00%**	**100.00%**	8.48	**42.37**	37.29	5.09%	3.39%	15.63	**58.82**	53.66	9.68%	6.56%
C3	87.10	88.85	**89.17**	87.18%	86.90%	**95.55**	**95.55**	95.42	95.28%	94.74%	91.13	92.08	**92.19**	91.05%	90.65%
C31	95.61	96.97	**97.77**	95.80%	95.86%	96.63	97.21	96.96	**97.29%**	97.04%	96.12	97.09	**97.36**	96.54%	96.45%
C32	94.17	94.10	93.47	**94.43%**	94.37%	96.38	**96.67**	96.58	96.18%	95.21%	95.26	**95.37**	95.00	95.30%	94.79%
C34	90.20	90.18	**90.23**	89.67%	87.03%	94.88	**95.19**	94.63	94.88%	94.75%	92.48	**92.62**	92.38	92.20%	90.73%
C35	**100.00**	95.24	92.31	**100.00%**	**100.00%**	23.08	38.46	**46.15**	11.54%	7.69%	37.50	54.80	**61.54**	20.69%	14.29%
C36	**100.00**	**100.00**	**100.00**	**100.00%**	**100.00%**	1.89	13.21	**15.09**	1.89%	1.89%	3.70	23.33	**26.23**	3.70%	3.70%
C37	**100.00**	**100.00**	**100.00**	**100.00%**	**100.00%**	**5.26**	**5.26**	**5.26**	**5.26%**	**5.26%**	**10.00**	**10.00**	**10.00**	**10.00%**	**10.00%**
C38	82.65	82.36	82.78	**83.19%**	82.66%	**95.61**	94.64	94.64	95.52%	95.22%	88.66	88.07	88.31	**88.93%**	88.50%
C39	85.36	**88.69**	87.68	84.86%	85.93%	96.73	**96.97**	96.49	**96.97%**	96.89%	90.69	**92.65**	91.88	90.51%	91.08%
C4	**100.00**	**100.00**	**100.00**	**100.00%**	**100.00%**	**5.88**	**5.88**	**5.88**	**5.88%**	2.94%	**11.11**	**11.11**	**11.11**	**11.11%**	5.71%
C5	**66.67**	**66.67**	**66.67**	**66.67%**	**66.67%**	**3.28**	**3.28**	**3.28**	**3.28%**	**3.28%**	**6.25**	**6.25**	**6.25**	**6.25%**	**6.25%**
C6	88.89	88.89	**90.00**	88.89%	88.89%	17.78	17.78	**20.00**	17.78%	17.78%	29.63	29.63	**32.73**	29.63%	29.63%
C7	**82.78**	81.30	80.99	82.43%	81.70%	68.80	72.44	**73.72**	68.16%	65.81%	75.15	76.61	**77.18**	74.62%	72.90%

Table 13: Performance on the Fudan corpus using the RF classifier.

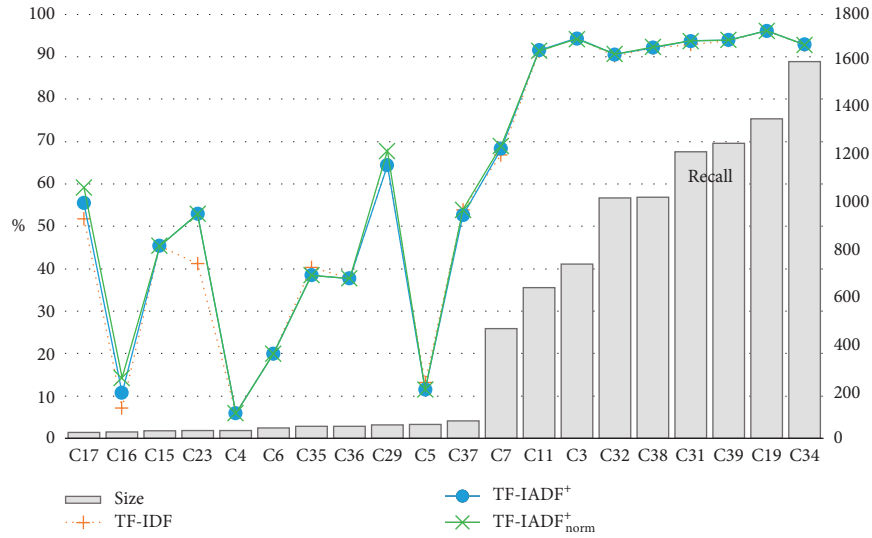
RF	TF-IDF (%)	Precision				Recall					*F*1 score				
		TF-IADF (%)	TF-$IADF_{norm}$ (%)	TF-$IADF^{+}$ (%)	TF-$IADF^{+}{}_{norm}$ (%)	TF-IDF (%)	TF-IADF (%)	TF-$IADF_{norm}$ (%)	TF-$IADF^{+}$ (%)	TF-$^{+}{}_{norm}$ (%)	TF-IDF (%)	TF-IADF (%)	TF-$IADF_{norm}$ (%)	TF-$IADF^{+}$ (%)	TF-$IADF^{+}{}_{norm}$ (%)
C11	82.94	**90.85**	85.52	87.61	86.50	82.56	**85.05**	79.13	78.19	83.80	82.75	**87.85**	82.20	82.63	85.13
C15	33.33	80.00	44.44	66.67	**83.33**	**15.15**	12.12	12.12	12.12	**15.15**	20.83	21.05	19.05	20.51	**25.64**
C16	40.00	50.00	**57.14**	30.77	**57.14**	7.14	**17.86**	14.29	14.29	14.29	12.12	**26.32**	22.86	19.51	22.86
C17	55.56	50.00	38.71	58.33	**61.91**	37.04	37.04	44.44	25.93	**48.15**	44.44	42.55	41.38	35.90	**54.17**
C19	89.90	**90.55**	87.61	86.31	88.24	96.32	96.61	96.32	**97.05**	96.69	93.00	**93.48**	91.76	91.37	92.27
C23	43.75	38.46	33.33	34.78	**50.00**	20.59	14.71	8.82	**23.53**	14.71	28.00	21.28	13.95	**28.07**	22.73
C29	43.59	**69.57**	50.00	67.74	60.87	28.81	27.12	22.03	35.59	**47.46**	34.69	39.02	30.59	46.67	**53.33**
C3	**75.77**	74.89	73.41	75.14	71.73	86.39	**90.43**	**90.43**	89.62	87.87	80.73	**81.93**	81.04	81.75	78.98
C31	92.04	93.17	91.77	92.86	**93.49**	90.23	**91.79**	91.54	90.72	88.42	91.13	**92.47**	91.66	91.78	90.89
C32	84.10	87.01	86.30	**88.08**	81.96	84.34	86.50	83.86	**86.79**	80.92	84.22	86.75	85.06	**87.43**	81.44
C34	74.49	**75.85**	73.59	75.03	72.61	88.26	**90.26**	89.82	89.69	88.07	80.79	**82.43**	80.90	81.71	79.59
C35	66.67	**77.78**	56.52	68.75	60.00	7.69	13.46	**25.00**	21.15	17.31	13.79	22.95	**34.67**	32.35	26.87
C36	57.14	66.67	51.85	**75.00**	51.52	22.64	30.19	26.42	28.30	**32.08**	32.43	**41.56**	35.00	41.10	39.54
C37	42.86	42.86	45.46	13.64	**66.67**	11.84	**15.79**	6.58	3.95	10.53	18.56	**23.08**	11.49	6.12	18.18
C38	65.66	66.45	**70.00**	66.86	66.81	76.22	77.97	**80.51**	80.02	76.51	70.55	71.75	**74.89**	72.85	71.33
C39	84.79	86.29	87.54	**87.65**	86.67	84.45	**85.33**	84.05	82.62	81.42	84.62	**85.81**	85.76	85.06	83.96
C4	**100.00**	75.00	**100.00**	**100.00**	**100.00**	5.88	**8.82**	5.88	5.88	**8.82**	11.11	15.79	11.11	11.11	**16.22**
C5	55.56	**85.71**	50.00	62.50	77.78	8.20	9.84	**11.48**	8.20	**11.48**	14.29	17.65	18.67	14.49	**20.00**
C6	**90.91**	84.62	83.33	63.64	90.00	22.22	**24.44**	22.22	15.56	20.00	35.71	**37.93**	35.09	25.00	32.73
C7	69.54	70.48	73.04	**77.91**	65.05	25.86	25.00	17.95	**27.14**	25.86	37.70	36.91	28.82	**40.25**	37.00

Table 14: Performance on the Fudan corpus using the NB classifier.

NB	Precision					Recall					*F*1 score				
	TF-IDF (%)	TF-IADF (%)	TF-$IADF_{norm}$ (%)	TF-$IADF^{+}$	TF-$IADF^{+}{}_{norm}$ (%)	TF-IDF (%)	TF-IADF (%)	TF-$IADF_{norm}$ (%)	TF-$IADF^{+}$ (%)	TF-$IADF^{+}{}_{norm}$ (%)	TF-IDF (%)	TF-IADF (%)	TF-$IADF_{norm}$	TF-$IADF^{+}$ (%)	TF-$IADF^{+}{}_{norm}$ (%)
C11	94.98	94.57	94.53	**94.99**	94.83	91.28	89.56	88.79	**91.59**	91.43	93.09	92.00	91.57	**93.26**	93.10
C15	93.75	**100.00**	**100.00**	93.75	93.75	**45.46**	30.30	27.27	**45.46**	**45.46**	**61.22**	46.51	42.86	**61.22**	**61.22**
C16	66.67	**100.00**	**100.00**	75.00	80.00	7.14	3.57	3.57	10.71	**14.29**	12.90	6.90	6.90	18.75	**24.24**
C17	87.50	86.67	83.33	93.75	**94.12**	51.85	48.15	37.04	55.56	**59.26**	65.12	61.91	51.28	69.77	**72.73**
C19	96.38	96.02	95.87	**96.46**	96.38	**96.17**	96.02	95.80	**96.17**	**96.17**	96.28	96.02	95.84	**96.31**	96.28
C23	93.33	92.31	**100.00**	94.74	90.00	41.18	35.29	20.59	**52.94**	**52.94**	57.14	51.06	34.15	**67.93**	66.67
C29	86.36	87.88	87.10	86.36	**88.89**	64.41	49.15	45.76	64.41	**67.80**	73.79	63.04	60.00	73.79	**76.92**
C3	82.94	81.81	81.19	83.43	**83.51**	94.34	95.15	**95.42**	94.34	94.21	88.27	87.98	87.73	**88.55**	88.54
C31	97.59	**97.89**	97.78	97.69	97.69	92.94	91.38	90.48	**93.76**	**93.76**	95.21	94.52	93.99	**95.69**	**95.69**
C32	**91.92**	90.70	90.35	91.68	91.60	90.22	88.75	87.97	90.61	**90.71**	91.06	89.71	89.14	91.14	**91.15**
C34	85.05	82.86	81.32	85.76	**86.20**	92.76	92.69	92.44	**92.94**	92.88	88.74	87.50	86.52	89.21	**89.42**
C35	**87.50**	86.96	**87.50**	86.96	86.96	**40.39**	38.46	**40.39**	38.46	38.46	**55.26**	53.33	**55.26**	53.33	53.33
C36	90.91	88.89	**93.33**	90.91	90.91	**37.74**	30.19	26.42	**37.74**	**37.74**	**53.33**	45.07	41.18	**53.33**	**53.33**
C37	78.85	**84.09**	81.40	78.43	78.85	**53.95**	48.68	46.05	52.63	**53.95**	**64.06**	61.67	58.82	62.99	**64.06**
C38	84.30	83.12	82.16	85.46	**85.55**	92.11	**92.59**	92.01	92.20	92.30	88.03	87.60	86.81	88.70	**88.80**
C39	89.30	88.29	87.30	**89.39**	**89.39**	93.86	93.22	92.66	**94.02**	**94.02**	91.52	90.69	89.90	**91.64**	**91.64**
C4	50.00	**100.00**	**100.00**	50.00	50.00	**5.88**	**5.88**	**5.88**	**5.88**	**5.88**	10.53	**11.11**	**11.11**	10.53	10.53
C5	80.00	75.00	71.43	**87.50**	77.78	**13.12**	9.84	8.20	11.48	11.48	**22.54**	17.39	14.71	20.29	20.00
C6	**90.00**	**90.00**	**90.00**	**90.00**	**90.00**	**20.00**	**20.00**	**20.00**	**20.00**	**20.00**	**32.73**	**32.73**	**32.73**	**32.73**	**32.73**
C7	69.10	71.76	**72.49**	68.97	69.17	66.88	66.24	64.74	68.38	**69.02**	67.97	68.89	68.40	68.67	**69.09**

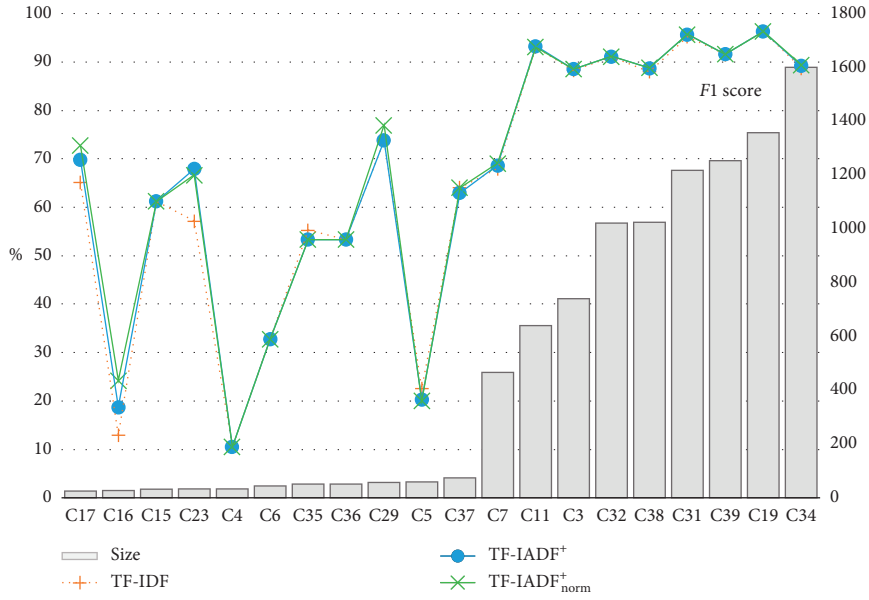


(a)



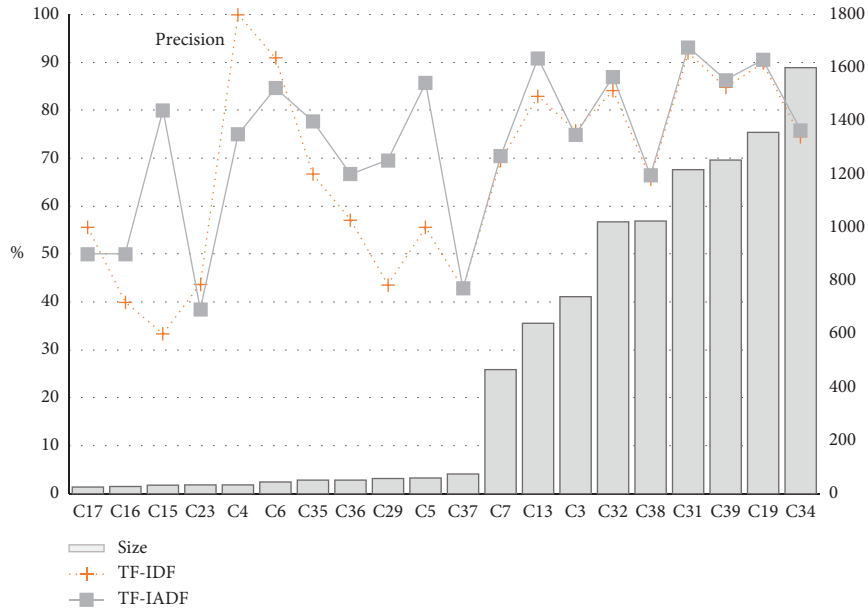
(b)

Figure 12: Continued.



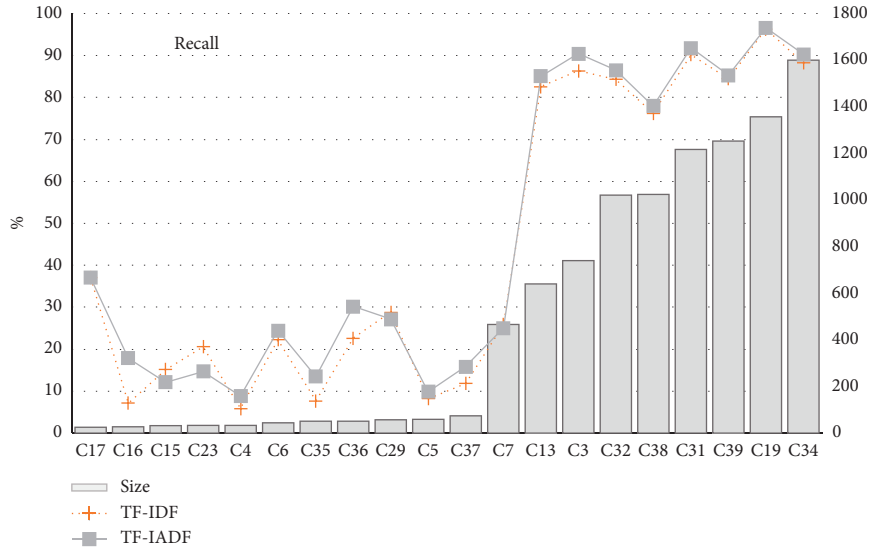
(c)

Figure 12: Detailed performances on the Fudan corpus using the NB classifier.

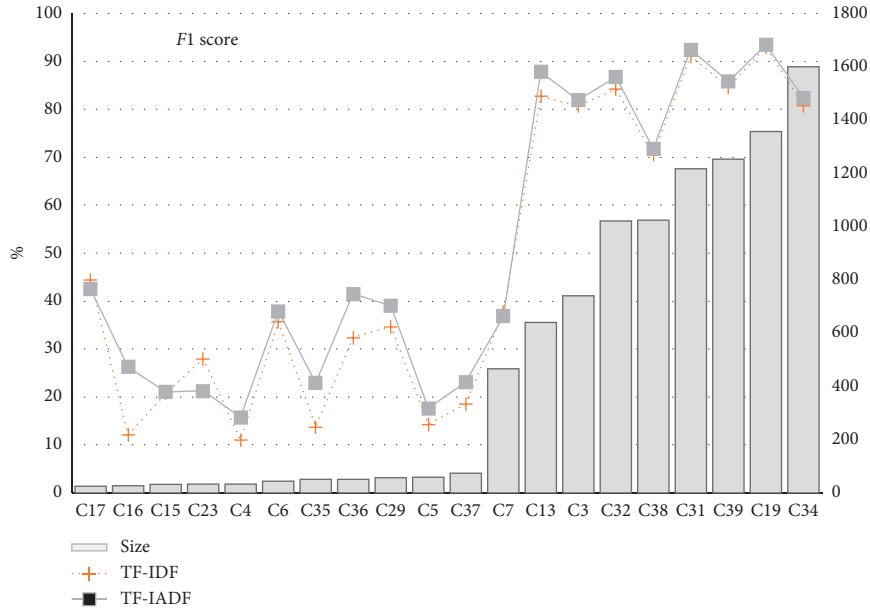


(a)

Figure 13: Continued.



(b)



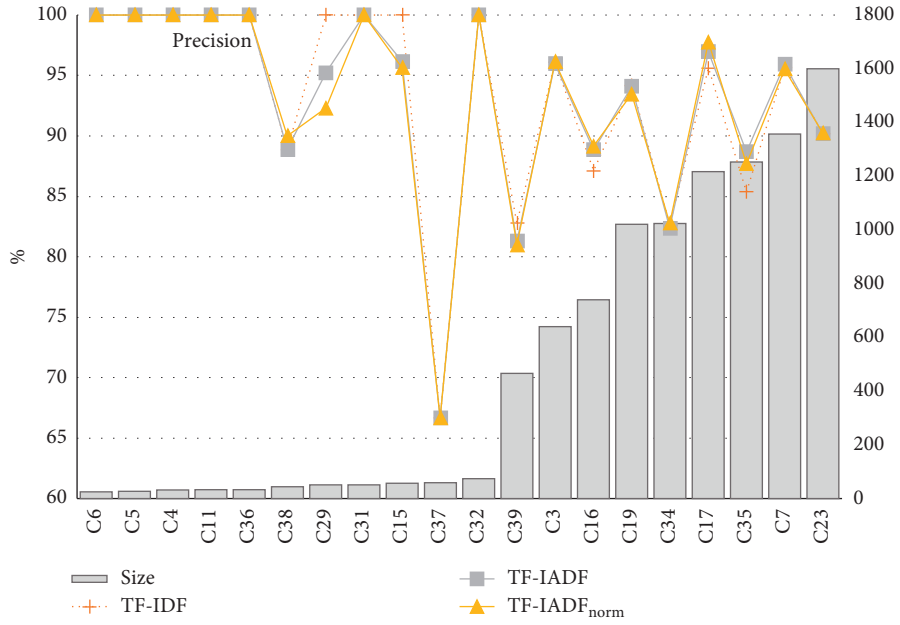
(c)

Figure 13: Detailed performances on the Fudan corpus using the RF classifier.

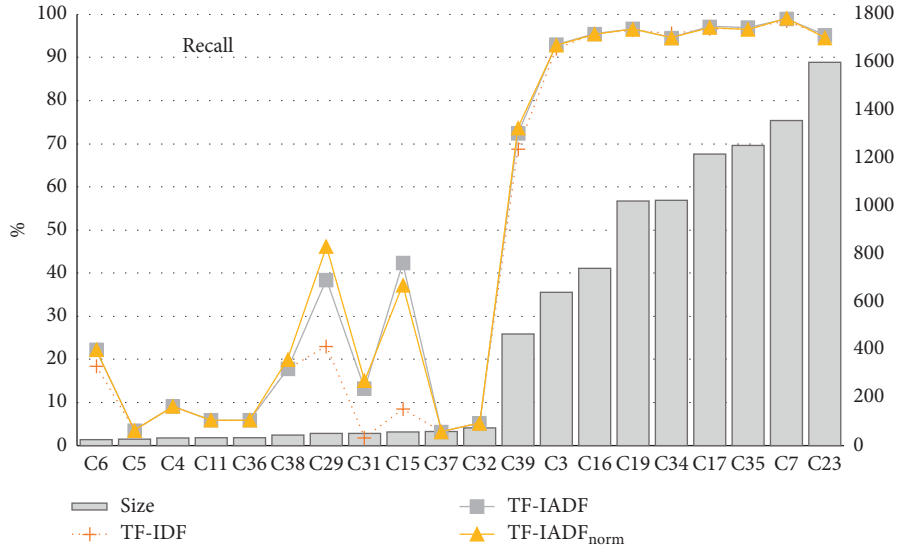
methods and the well-known TF-IDF method when combining with different classifiers. More detailed results are shown in Tables 16–18.

For the SVM classifier, TF-IADF$_{norm}$ comes with the best performance which is 0.86% higher than TF-IDF in the micro-$F1$ score. According to the detailed results shown in Table 15 and Figure 16, the improvement of TF-IADF$_{norm}$ is mainly contributed by the recall score, which is greatly improved in the category of crude, grain, interest, and money-fx. Especially in the category of money-fx, the recall score has increased by 10.35%. The other categories also improved in different extents, namely, crude increased by 3.31%, interest increased by 4%, and grain increased by 10%. From the perspective of the $F1$ score, in the total eight categories, the max $F1$ score of five categories is obtained by TF-IADF$_{norm}$, and the largest increase is obtained in the category of grain, where the $F1$ score increased from 57.14% of TF-IDF to 66.67% of TF-IADF$_{norm}$, close to 10%.

For the NB classifier, the results are different from those on the Chinese dataset, where TF-IADF^{+} and TF-IADF$^{+}{}_{norm}$ achieved better performance, and performances of TF-IADF^{+} and TF-IADF$^{+}{}_{norm}$ were worse on this corpus. However, the proposed TF-IADF and TF-IADF$_{norm}$ showed an improvement, the micro-$F1$ score of which is 0.59% and 0.5% higher than that of TF-IDF, respectively. As shown in Table 16 and Figure 17, TF-IADF has achieved the highest $F1$ score among all methods in all categories of this corpus, and the maximum improvement, 3.83%, occurs in the category of interest. Specifically, in terms of precision, we see that the four categories of cloud, interest, ship, and trade have

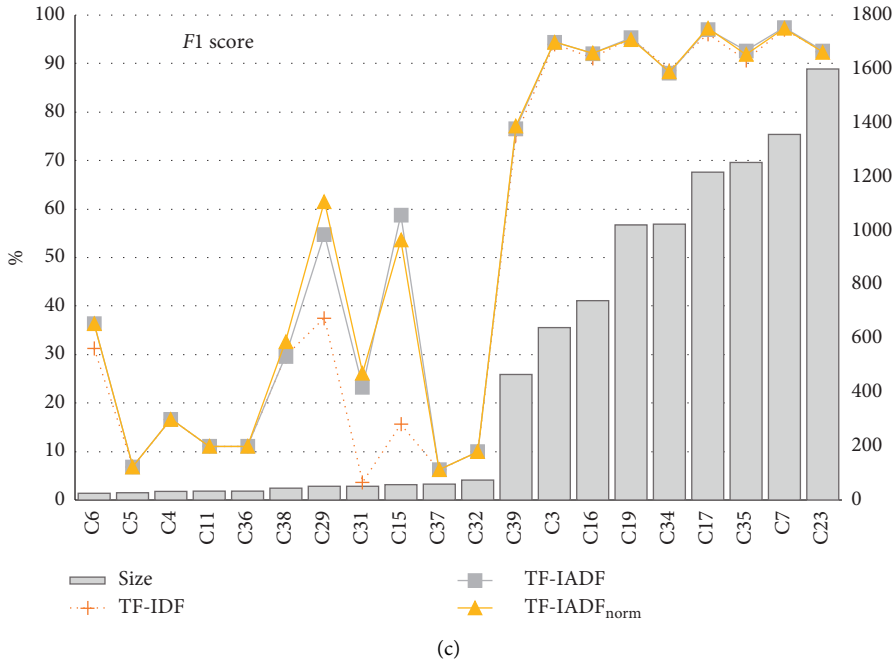


(a)



(b)

Figure 14: Continued.



(c)

Figure 14: Detailed performances on the Fudan corpus using the SVM classifier.

Table 15: Overall performances on Reuters-21578.

	TF-IDF (%)	TF-IADF (%)	TF-IADF$_{norm}$ (%)	TF-IADF^{+} (%)	TF-IADF$^{+}_{norm}$ (%)
micro-*F*1 (SVM)	91.00	**91.41**	91.86	90.59	90.23
macro-*F*1 (SVM)	79.63	**80.33**	81.97	78.42	75.77
micro-*F*1 (RF)	87.45	**85.23**	87.77	87.41	88.14
macro-*F*1 (RF)	69.90	**59.03**	65.43	67.91	67.76
micro-*F*1 (NB)	89.82	90.41	90.32	89.41	**89.36**
macro-*F*1 (NB)	76.22	77.86	77.77	76.21	**75.95**

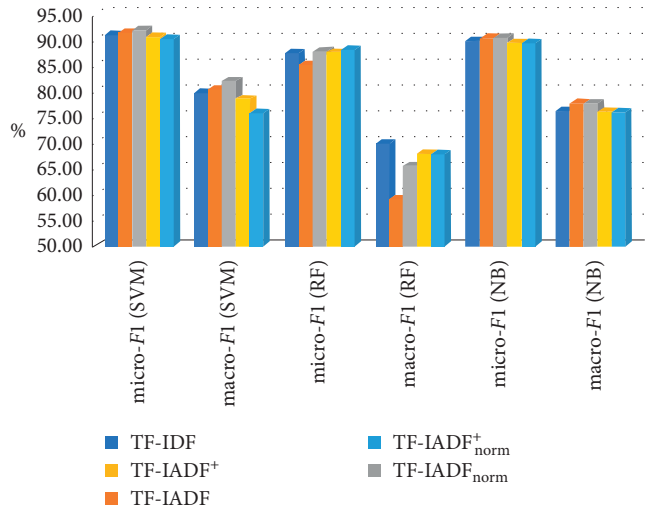


Figure 15: Performance of the NB classifier on Reuters-21578 (precision, recall, and *F*1 score).

improved significantly, with an improvement of 2.36%, 4.35%, 6.82%, and 3.75%, respectively. In terms of the recall score, the increase in the money-fx category is 4.60%.

For the RF algorithm, TF-IADF$^{+}_{norm}$ and TF-IADF$_{norm}$ have improved the effect, micro-*F*1 score of which is 0.69% and 0.32% higher than that of TF-IDF, respectively. TF-

Table 16: Performance on Reuters-21758 using the SVM classifier.

SVM	Precision					Recall					$F1$ score				
	TF-IDF (%)	TF-IADF (%)	TF-$IADF_{norm}$ (%)	TF-$IADF^{+}$ (%)	TF-$IADF^{+}{}_{norm}$ (%)	TF-IDF (%)	TF-IADF (%)	TF-$IADF_{norm}$ (%)	TF-$IADF^{+}$ (%)	$TFIADF^{+}{}_{norm}$ (%)	TF-IDF (%)	TF-IADF (%)	TF-$IADF_{norm}$ (%)	TF-$IADF^{+}$ (%)	TF-$IADF^{+}{}_{norm}$ (%)
Acq	90.17	91.43	**93.03**	90.06	90.42	92.24	91.95	92.10	**92.39**	92.24	91.19	91.69	**92.56**	91.21	91.32
Crude	90.39	89.72	89.91	**92.63**	92.55	77.69	79.34	**80.99**	72.73	71.90	83.56	84.21	**85.22**	81.48	80.93
Earn	92.06	92.00	**92.48**	91.35	90.21	97.42	97.69	97.69	97.51	**97.88**	94.66	94.76	**95.02**	94.33	93.89
Grain	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	40.00	40.00	**50.00**	40.00	30.00	57.14	57.14	**66.67**	57.14	46.15
Interest	**91.30**	90.00	88.24	91.11	90.48	56.00	**60.00**	**60.00**	54.67	50.67	69.42	**72.00**	71.43	68.33	64.96
Moneyfx	79.71	**84.06**	80.00	79.10	79.03	63.22	66.67	**73.56**	60.92	56.32	70.51	74.36	**76.65**	68.83	65.77
Ship	**100.00**	**100.00**	95.83	95.65	95.24	**63.89**	61.11	**63.89**	61.11	55.56	**77.97**	75.86	76.67	74.58	70.18
Trade	90.63	89.80	88.78	89.58	**92.47**	94.57	**95.65**	94.57	93.48	93.48	92.55	92.63	91.58	91.49	**92.97**

Table 17: Performance on Reuters-21758 using the RF classifier.

RF	Precision					Recall					*F*1 score				
	TF-IDF (%)	TF-IADF (%)	TF-IADF$_{norm}$ (%)	TF-IADF^{+} (%)	TF-IADF$^{+}{}_{norm}$ (%)	TF-IDF (%)	TF-IADF (%)	TF-IADF$_{norm}$ (%)	TF-IADF^{+} (%)	TF-IADF$^{+}{}_{norm}$ (%)	TF-IDF (%)	TF-IADF (%)	TF-IADF$_{norm}$ (%)	TF-IADF^{+} (%)	TF-IADF$^{+}{}_{norm}$ (%)
Acq	86.52	82.25	85.41	81.77	**88.06**	90.37	90.52	92.53	**94.11**	90.09	88.41	86.18	88.83	87.51	**89.06**
Crude	92.68	**93.24**	84.88	91.25	86.81	62.81	57.03	60.33	60.33	**65.29**	**74.88**	70.77	70.53	72.64	74.53
Earn	89.89	87.78	90.48	**93.11**	89.74	96.86	96.86	96.58	94.83	**97.69**	93.24	92.10	93.44	**93.96**	93.55
Grain	**100.00**	**100.00**	33.33	**100.00**	50.00	**30.00**	10.00	10.00	20.00	10.00	**46.15**	18.18	15.39	33.33	16.67
Interest	75.51	74.29	78.00	72.73	**86.05**	49.33	34.67	52.00	**53.33**	49.33	59.68	47.27	62.40	61.54	**62.71**
Moneyfx	75.47	73.17	**82.69**	72.41	71.43	45.98	34.48	**49.43**	48.28	45.98	57.14	46.88	**61.87**	57.93	55.94
Ship	90.00	90.00	92.31	**100.00**	90.48	50.00	25.00	33.33	41.67	**52.78**	64.29	39.13	48.98	58.82	**66.67**
Trade	72.73	78.21	**84.88**	80.23	81.25	78.26	66.30	79.35	75.00	**84.78**	75.39	71.77	82.02	77.53	**82.98**

Table 18: Performance on Reuters-21758 using the NB classifier.

NB	Precision					Recall					*F*1 score				
	TF-IDF (%)	TF-IADF (%)	TF-$IADF_{norm}$ (%)	TF-$IADF^{+}$ (%)	TF-$IADF^{+}{}_{norm}$ (%)	TF-IDF (%)	TF-IADF (%)	TF (%)	TF-$IADF^{+}$ (%)	TF-$IADF^{+}{}_{norm}$ (%)	TF-IDF (%)	TF-IADF (%)	TF-$IADF_{norm}$ (%)	TF-$IADF^{+}$ (%)	TF-$IADF^{+}{}_{norm}$ (%)
Acq	**91.83**	91.74	91.60	91.12	91.34	92.10	**92.53**	92.39	91.38	90.95	91.97	**92.13**	91.99	91.25	91.15
Crude	85.71	**88.07**	**88.07**	86.49	87.16	**79.34**	**79.34**	**79.34**	**79.34**	78.51	82.40	**83.48**	**83.48**	82.76	82.61
Earn	92.65	**92.68**	92.67	92.14	91.84	95.48	**95.85**	95.75	95.20	95.57	94.04	**94.24**	94.19	93.64	93.67
Grain	**71.43**	**71.43**	**71.43**	**71.43**	**71.43**	**50.00**	**50.00**	**50.00**	**50.00**	**50.00**	**58.82**	**58.82**	**58.82**	**58.82**	**58.82**
Interest	78.26	**84.44**	82.61	78.26	79.55	48.00	**50.67**	**50.67**	48.00	46.67	59.50	**63.33**	62.81	59.50	58.82
Moneyfx	76.00	**77.22**	**77.22**	77.03	77.03	65.52	**70.12**	**70.12**	65.52	65.52	70.37	**73.49**	**73.49**	70.81	70.81
Ship	85.19	**92.00**	**92.00**	88.46	85.19	**63.89**	**63.89**	**63.89**	**63.89**	**63.89**	73.02	**75.41**	**75.41**	74.19	73.02
Trade	70.59	**74.34**	**74.34**	69.75	69.75	**91.30**	**91.30**	**91.30**	90.22	90.22	79.62	**81.95**	**81.95**	78.67	78.67

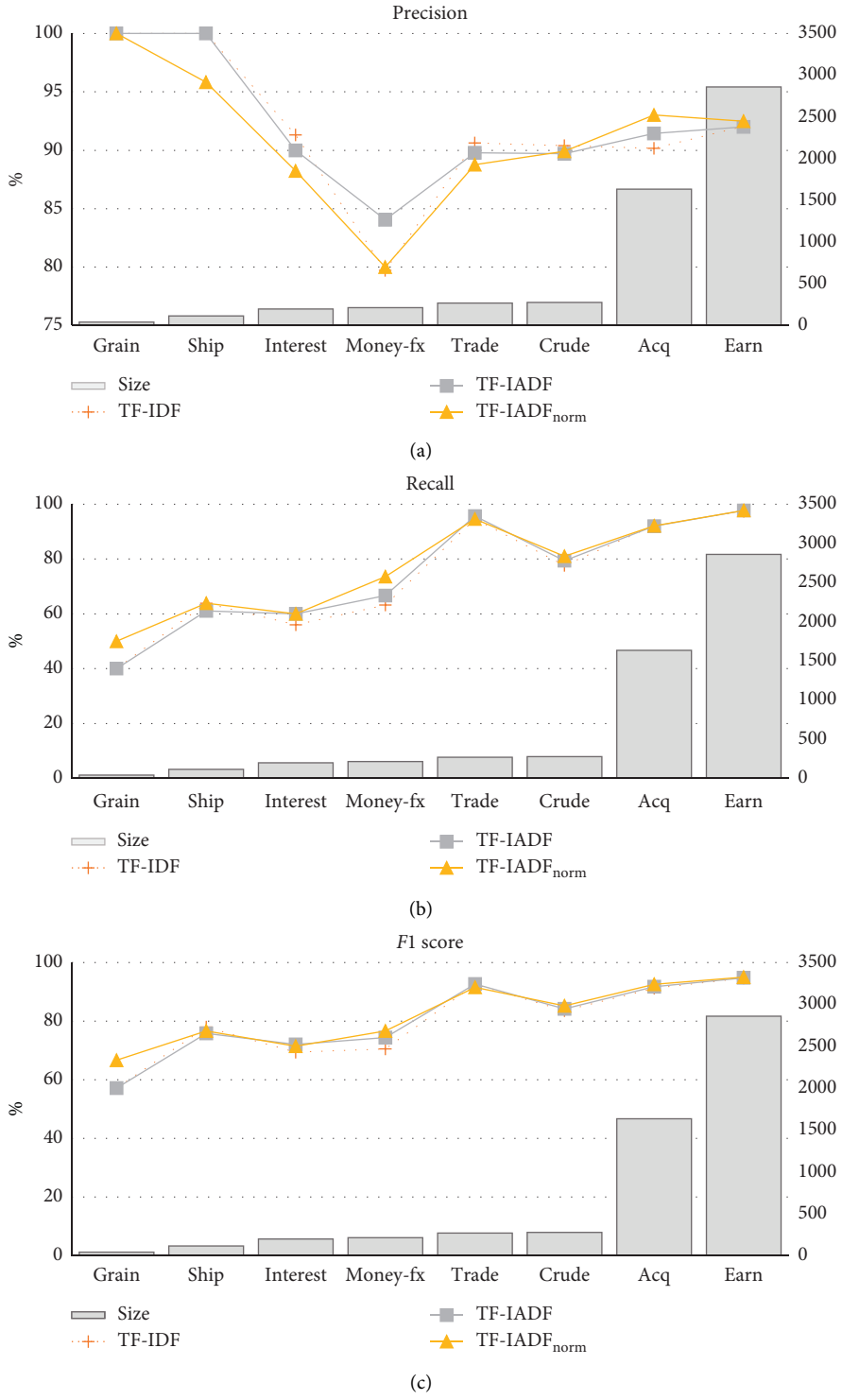


Figure 16: Performance of the SVM classifier on Reuters-21578 (precision, recall, and $F1$ score).

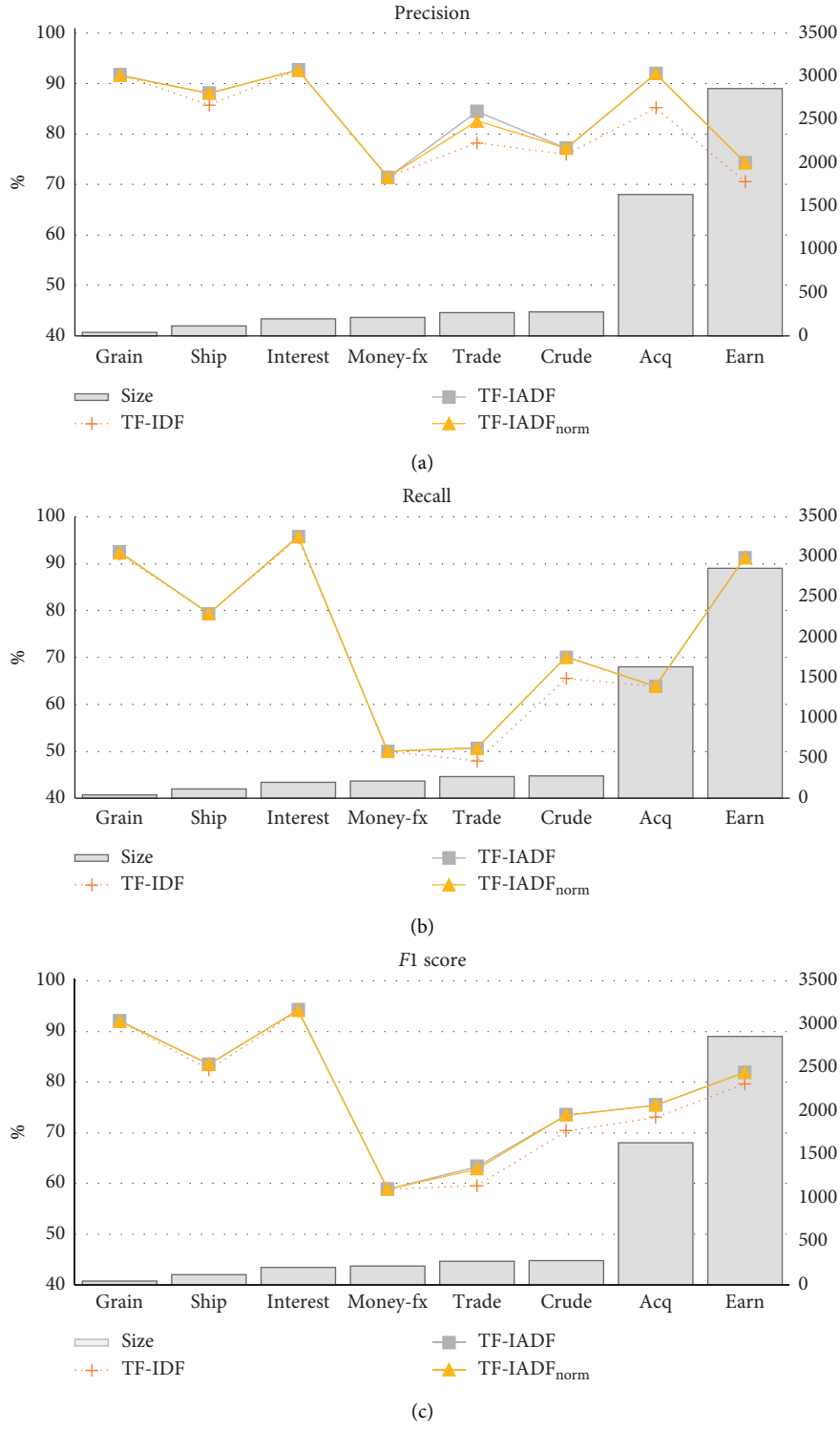
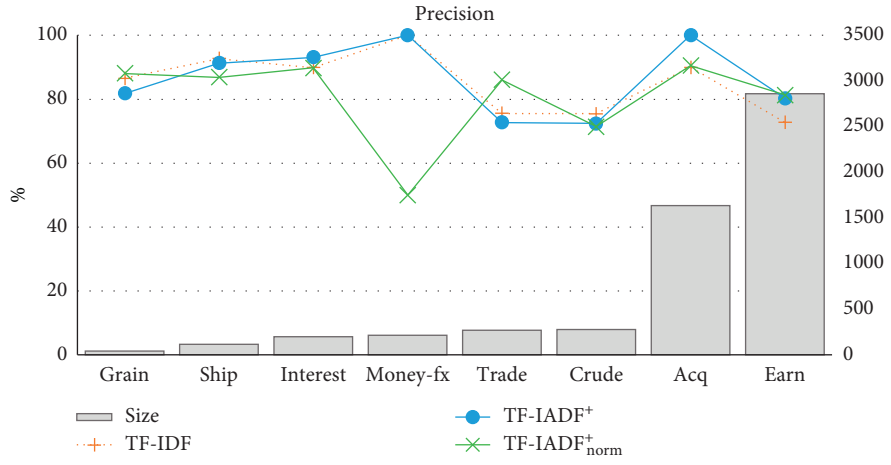
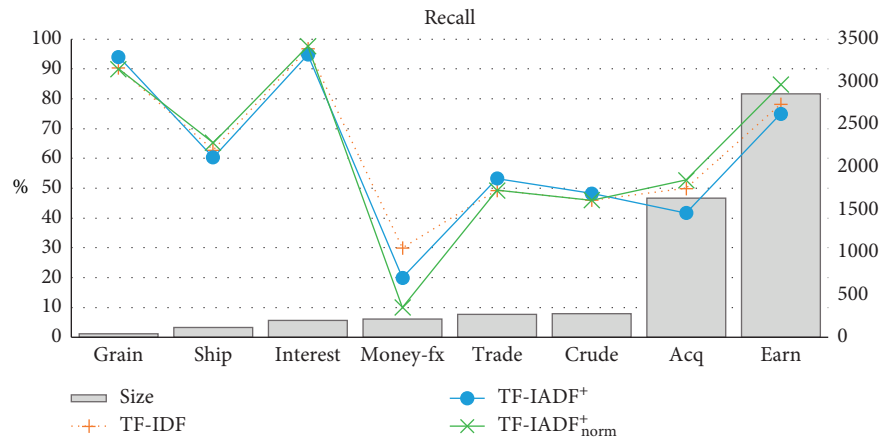


Figure 17: Performance of the NB classifier on Reuters-21578 (precision, recall, and $F1$ score).

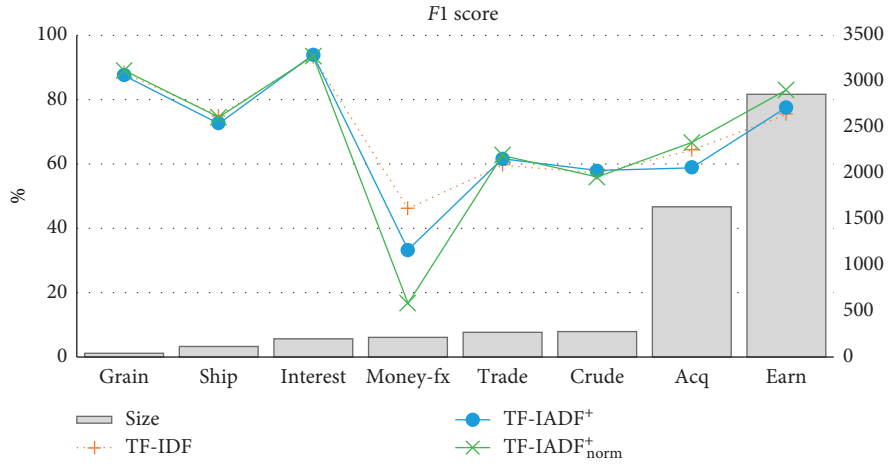
IADF, which has better performance in the Chinese dataset, has worse performance on this corpus. It can be seen in Table 17 and Figure 18 that the better performance of TF-IADF$^{+}_{norm}$ is mainly due to the higher recall score. Comparing with TF-IDF, the recall score obtained in the trade category is 6.52% higher, and in the crude category, it is 2.48% higher. In terms of precision, it has achieved good performance in the categories of interest and trade, with an increase of 10.54% and 8.52%, respectively. On the side of $F1$ score, the trade category improved significantly, up to 7.59%. However, we also see a decrease in the performance on the categories of grain and money-fx. This may be due to the small size of the test dataset, which changes greatly, and so, it is not easy to draw more accurate conclusions.



(a)



(b)



(c)

Figure 18: Performance of the RF classifier on Reuters-21578 (precision, recall, and $F1$ score).

Table 19: The best combinations of TWSs with classifiers.

Dataset/classifier	NB	RF	SVM
Balanced Chinese dataset	TF-IADF$^{+}_{\text{norm}}$	TF-IADF	TF-IADF
Unbalanced Chinese dataset	TF-IADF$^{+}_{\text{norm}}$	TF-IADF, TF-IADF$_{\text{norm}}$	TF-IADF, TF-IADF$_{\text{norm}}$
Unbalanced English dataset	TF-IADF	TF-IADF$^{+}_{\text{norm}}$	TF-IADF$_{\text{norm}}$

According to all experimental results on Reuters-21578, the proposed TF-IADF$_{\text{norm}}$ outperformed TF-IDF in nearly all conditions except the micro-$F1$ score of the RF classifier. In fact, the RF classifier's performance was worse among all classifiers, suggesting that the RF classifier may not be suitable for this corpus.

5.4. Discussion. The experimental results obtained on the English dataset (Reuters) are somehow different from those on the Chinese dataset, and the most effective combination of a term weighting method and classification algorithm is different. For example, TF-IADF$^{+}_{\text{norm}}$ is suitable for the Chinese internet corpus using the NB algorithm, whereas TF-IADF performs best in the English unbalanced dataset. It can be inferred from the experimental results that all of the proposed algorithms can generally always be combined with a suitable mathematical model that shows a better performance than the original TF-IDF. For example, TF-IADF$^{+}_{\text{norm}}$ performs better in the RF algorithm, while TF-IADF$_{\text{norm}}$ performs better in the SVM. The best combinations concluded from the experiments are shown in Table 19. For the two Chinese datasets in the experiment, we draw the following conclusions: (1) for the NB algorithm, TF-IADF^{+} and TF-IADF$^{+}_{\text{norm}}$ are more suitable, and TF-IADF$^{+}_{\text{norm}}$ can achieve better performance whether in balanced or unbalanced datasets; (2) for the RF algorithm and SVM algorithm, TF-IADF and TF-IADF$_{\text{norm}}$ can achieve a relatively stable improvement effect. In the experiments with the internet corpus, it can be concluded that TF-IADF has a better improvement effect when the dataset is relatively balanced, and TF-IADF$_{\text{norm}}$ has a better classification effect when the dataset is unbalanced. However, in the Fudan corpus (unbalanced), TF-IADF has achieved better performance than TF-IADF$_{\text{norm}}$, while TF-IADF$_{\text{norm}}$ has also improved. It is possible that TF-IADF is more suitable for RF and SVM algorithms when there are many categories which are unbalanced in the corpus.

6. Conclusions

In this paper, an improved TF-IDF with novel term weighting schemes is proposed to greatly reduce the impact from the unbalanced distribution of datasets. It is easy to be observed that all unbalanced corpuses that are categorized with a larger amount of data will always have an impact on the classification effect due to the larger amount of feature words. Meanwhile, the precision score decreases, while documents from other categories will be easily mistaken into this category. This is due to the increase in the training set; some feature words with strong performance capabilities in other categories have also appeared in this category which causes errors. Meanwhile, categories with smaller amount of data will be reflected in the significant reduction in the recall score due to the insufficient collection of feature words. The training process cannot classify these categories well without sufficient representation in the training process. As a result, many documents are assigned to wrong categories. In these cases, the proposed methods can increase the weight of those feature words which have a document frequency close to the average value, while reducing the weight of low-frequency and high-frequency words in order to obtain better results.

The simulation results show that the proposed methods with ADF are more effective than the original TF-IDF, although different mathematical models may be needed for improvement when utilizing different classification algorithms. Especially in experiments specifically designed in which the size of data in sport decreases while keeping other conditions the same, the results proved that the proposed methods are with a better performance and more stable than the well-known TF-IDF on the unbalanced corpus. It can also be concluded that our proposed methods come with a better performance in the balanced dataset when compared with TF-IDF. Document frequency of specific words may vary across categories, even in cases where the training sets appear roughly the same. Methods with ADF can weight those words more reasonably and form a more representative model.

However, for the purpose of TC on internet media reports, this paper just focuses on the term weighting scheme under unbalanced distribution but ignoring linguistic characteristics, which might be helpful in the term extraction process. Therefore, our next study will focus on enhancing the TC performance by combining the proposed methods with language characteristics.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant nos. 61973024 and 61703027 and Fundamental Research Funds for the Central Universities under Grant nos. JD1914 and XK1802-4.

References

[1] CNNIC (China Ineternet Network Information Center), "The 43rd China statistical report on internet development," 2019, http://www.cac.gov.cn/2019-02/28/c_1124175677.htm.

[2] A. Sboev, T. Litvinova, D. Gudovskikh, R. Rybka, and I. Moloshnikov, "Machine learning models of text categorization by author gender using topic-independent features," *Procedia Computer Science*, vol. 101, pp. 135–142, 2016.

[3] J. S. Li, L. C. Chen, J. V. Monaco, P. Singh, and C. C. Tappert, "A comparison of classifiers and features for authorship authentication of social networking messages," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 14, p. e3918, 2017.

[4] J. R. Méndez, T. R. Cotos-Yañez, and D. Ruano-Ordás, "A new semantic-based feature selection method for spam filtering," *Applied Soft Computing*, vol. 76, pp. 89–104, 2019.

[5] B. Parlak and A. K. Uysal, "The impact of feature selection on medical document classification," in *Proceedings of the 2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–5, IEEE, Las Palmas, Spain, June 2016.

[6] H. Lu, D. Zhan, L. Zhou, and D. He, "An improved focused crawler: using web page classification and link priority evaluation," *Mathematical Problems in Engineering*, vol. 2016, Article ID 6406901, 10 pages, 2016.

[7] T. Sabbah, A. Selamat, M. H. Selamat, R. Ibrahim, and H. Fujita, "Hybridized term-weighting method for dark web classification," *Neurocomputing*, vol. 173, pp. 1908–1926, 2016.

[8] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining Text Data*, pp. 163–222, Springer, Boston, MA, USA, 2012.

[9] D. Wang and H. Zhang, "Inverse-category-frequency based supervised term weighting scheme for text categorization," 2010, https://arxiv.org/abs/1012.2609.

[10] P. Soucy and G. W. Mineau, "Beyond TFIDF weighting for text categorization in the vector space model," in *Proceedings of the IJCAI*, pp. 1130–1135, Trento, Italy, July 2005.

[11] T. Sabbah, A. Selamat, M. H. Selamat et al., "Modified frequency-based term weighting schemes for text classification," *Applied Soft Computing*, vol. 58, pp. 193–206, 2017.

[12] N. Passalis and A. Tefas, "Entropy optimized feature-based bag-of-words representation for information retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1664–1677, 2016.

[13] M. Haddoud, A. Mokhtari, T. Lecroq, and S. Abdeddaïm, "Combining supervised term-weighting metrics for SVM text classification with extended term representation," *Knowledge and Information Systems*, vol. 49, no. 3, pp. 909–931, 2016.

[14] Z. Tang, W. Li, and Y. Li, "An improved term weighting scheme for text classification," *Concurrency and Computation: Practice and Experience*, vol. 32, p. e5604, 2019.

[15] B. Altınel and M. C. Ganiz, "Semantic text classification: a survey of past and recent advances," *Information Processing & Management*, vol. 54, no. 6, pp. 1129–1153, 2018.

[16] G. Rao, W. Huang, Z. Feng, and Q. Cong, "LSTM with sentence representations for document-level sentiment classification," *Neurocomputing*, vol. 308, pp. 49–57, 2018.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[18] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the International Conference on Machine Learning*, pp. 1188–1196, Beijing, China, January 2014.

[19] H. K. Kim, H. Kim, and S. Cho, "Bag-of-concepts: comprehending document representation through clustering words in distributed representation," *Neurocomputing*, vol. 266, pp. 336–352, 2017.

[20] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec," *Information Sciences*, vol. 477, pp. 15–29, 2019.

[21] Y. Wu, S. Zhao, and W. Li, "Phrase2Vec: phrase embedding based on parsing," *Information Sciences*, vol. 517, pp. 100–127, 2020.

[22] J. Kim, S. Jang, E. Park, and S. Choi, "Text classification using capsules," *Neurocomputing*, vol. 376, pp. 214–221, 2020.

[23] Y.-W. Chen, Q. Zhou, W. Luo, and J.-X. Du, "Classification of Chinese texts based on recognition of semantic topics," *Cognitive Computation*, vol. 8, no. 1, pp. 114–124, 2016.

[24] A. Elnagar, R. Al-Debsi, and O. Einea, "Arabic text classification using deep learning models," *Information Processing & Management*, vol. 57, no. 1, Article ID 102121, 2020.

[25] K. Chen, Z. Zhang, J. Long, and H. Zhang, "Turning from TF-IDF to TF-IGM for term weighting in text classification," *Expert Systems with Applications*, vol. 66, pp. 245–260, 2016.

[26] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, 1972.

[27] F. Debole and F. Sebastiani, "Supervised term weighting for automated text categorization," in *Text Mining and its Applications*, pp. 81–97, Springer, Berlin, Germany, 2004.

[28] T. Dogan and A. K. Uysal, "Improved inverse gravity moment term weighting for text classification," *Expert Systems with Applications*, vol. 130, pp. 45–59, 2019.

[29] F. Ren and C. Li, "Hybrid Chinese text classification approach using general knowledge from baidu baike," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 11, no. 4, pp. 488–498, 2016.

[30] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," 2013, https://arxiv.org/abs/1305.1707.

[31] M. N. Anwar, *Complexity Measurement for Dealing with Class Imbalance Problems in Classification Modelling*, Massey University, Palmerston, New Zealand, 2012.

[32] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[33] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, p. 27, 2019.

[34] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: experimental evaluation," *Information Sciences*, vol. 513, pp. 429–441, 2020.

[35] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," *Concurrency and Computation: Practice and Experience*, p. e5909, 2020.

[36] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 721–735, 2008.

[37] M. Lichman, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, CA, USA, 2017.

[38] A. Onan, S. Korukoğlu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Systems with Applications*, vol. 57, pp. 232–247, 2016.
[39] D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan, "Hybrid decision tree and naïve bayes classifiers for multi-class classification tasks," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1937–1946, 2014.
[40] G. Bianchi, R. Bruni, and F. Scalfati, "Identifying e-commerce in enterprises by means of text mining and classification algorithms," *Mathematical Problems in Engineering*, vol. 2018, Article ID 7231920, 8 pages, 2018.
[41] Z. Geng, Y. Zhang, and Y. Han, "Joint entity and relation extraction model based on rich semantics," *Neurocomputing*, vol. 429, 2020.
[42] X. Wu, Y. Gao, and D. Jiao, "Multi-label classification based on random forest algorithm for non-intrusive load monitoring system," *Processes*, vol. 7, no. 6, p. 337, 2019.
[43] Q. Wu, Y. Ye, H. Zhang, M. K. Ng, and S.-S. Ho, "Forestexter: an efficient random forest algorithm for imbalanced text categorization," *Knowledge-Based Systems*, vol. 67, pp. 105–116, 2014.
[44] A. Onan, "Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering," *IEEE Access*, vol. 7, pp. 145614–145633, 2019.