Soft Computing (2023) 27:3563–3579
https://doi.org/10.1007/s00500-022-07597-5

An improved term weighting method based on relevance frequency for text classification

Chuanxiao Li[1,2] · Wenqiang Li[1,2] · Zhong Tang[1,2] · Song Li[1,2] · Hai Xiang[1,2]

Accepted: 14 October 2022 / Published online: 10 November 2022

Abstract
As a vital step of text classification (TC) task, the assignment of term weight has a great influence on the performance of TC. Currently, masses of term weighting methods can be utilized, such as term frequency-inverse documents frequency and term frequency-relevance frequency (TF-RF). It can be found that they are both consisted of local part (TF) and global part (e.g., IDF, RF). However, most of these methods adopt the logarithmic processing on their respective global parts, so it is natural to consider whether the logarithmic processing applies to all these methods or not. Actually, for a specific term weighting method, due to its different ratio of local weight and global weight resulting from logarithmic processing, it usually shows diverse text classification results on different text sets, which shows poor robustness. To explore the influence of logarithmic processing imposed on the TC performance of term weighting methods, TF-RF is selected as the representative because it can achieve relatively stable performance among these methods adopting logarithmic processing. Then, in order to balance the local part and global part of TF-RF, an improved term weighting method based on TF-RF is proposed, named as term frequency-exponential relevance frequency (TF-ERF). And two groups of experiments are conducted on TF-ERF and other existing term weighting methods based on two general standard corpora. The results show that the improved term weighting method TF-ERF has better text classification performance and robustness.

Keywords Text classification · Term weighting · Relevance frequency · Logarithm processing

1 Introduction

Nowadays, with the development of Internet technology, we have entered the information age quickly. Massive data is created and transformed to the virtual environment rapidly every day and most of them exists in a form of textual document (Sebastiani 2002; Al-Mubaid and Umair 2006; Debole and Sebastiani 2003). Meanwhile, people's demand that improving the matching degree between retrieval words and provided documents is also raising (Li et al. 2011). Therefore, it is essential to classify these documents accurately according to their content. However, confronted with the continuously growing text data, it is inefficient and impractical to deal with a large amount of information only by manual work (Labani et al. 2018). Luckily, text classification (TC) technology rising in recent years can be utilized to accomplish this target (Shang et al. 2013; Tellez et al. 2018). TC is such a task which aims at assigning corresponding category labels to the documents in accordance with their respective topics by specific classification algorithms (Shang et al. 2007; Zhang et al. 2011; Haddoud et al. 2016). This technology has been applied in many fields, including spam filtering (Li and Liu 2018), web page detection (Deng et al. 2020) and function extraction of patent texts (Li et al. 2017; Liu et al. 2020). However, computers cannot identify document texts like human beings do. Therefore, it is essential to transform the format of these texts into an appropriate one in order to make them recognized by computers and classifiers successfully, and this process of transformation is called text representation (Lan et al. 2009). Currently, vector space model (VSM) is a widely-used text representation method in which documents are transformed into vectors weighted by specific

✉ Wenqiang Li
liwenqiang@scu.edu.cn

[1] School of Mechanical Engineering, Sichuan University, Chengdu 610065, China

[2] Innovation Method and Creative Design Key Laboratory of Sichuan Province, Chengdu 610065, China

measurements (Salton et al. 1974; Zong et al. 2015). In this model, the document can be represented by the form of $d_k = (t_1, t_2, ..., t_n)$ with a corresponding weight of $w = (w_1, w_2, ..., w_n)$, in which n denotes the number of selected features and d_k denotes a specific document and w is the set of all term weights (Sabbah et al. 2017; Wu et al. 2017). In the whole process, the assignment of term weight is a vital step because the weight demonstrates the importance of a specific term and the contribution made by this term in classifying different kinds of texts (Debole and Sebastiani 2003; Lan et al. 2009; Guru et al. 2018).

In order to assign appropriate weights to terms, it is essential to choose a reasonable term weighting method. Generally, term weighting methods can be mainly divided into two categories, supervised term weighting (STW) methods and unsupervised term weighting (UTW) methods according to whether the predefined category information is utilized or not (Lan et al. 2009; Wang and Zhang 2013; Ren and Sohrab 2013; Chen et al. 2016). Due to different term weighting methods adopt various models, therefore, even if these methods belong to the same UTW or STW type, there are still many differences between them. For example, the TF method only takes the term frequency into account when computing the term weight, and it holds the assumption that the higher the frequency of appearance of a term, the more important this term is. On the contrary, the IDF method is based on the assumption that the less documents in which a term appears, the more significant the term will be, which ignores the effect of term frequency completely (Zhang et al. 2011; Sabbah et al. 2016). Considering that both TF and IDF have respective defects as a single term weighting method, TF-IDF was then proposed by combining them together (Li et al. 2016), which supports the idea that the term weight should be measured from both term frequency and document frequency comprehensively (Spärck 2004; Salton and Buckley 1988). It should be noticed that the three term weighting methods mentioned above are all UTW methods as the available category information is not utilized in the text training process. From the core ideas of the above methods, we can see that the term weight measured by them only reflect the relationship between terms and documents as well as documents and documents. However, TC is a supervised learning task which aims at classifying various texts into different categories according to their content (Lan et al. 2009). It seems that UTW methods cannot achieve the goal of TC with a satisfactory result.

In order to obtain a more reasonable term weighting method, it is natural to take the category information into account to match the TC task which is also a supervised learning process (Altınçay and Erenel 2010; Tang et al. 2020). Relative to the UTW method, the method which make use of the prior category information when computing the term weight, is called STW method (Guru et al. 2018). Most STW methods follow the pattern of TF-IDF which is consisted of local part and global part. Besides, it is publicly accepted that term frequency (TF) is an excellent representation of local weight, so the TF part is still retained while the IDF part is substituted by others in these new-proposed STW methods. For example, considering that the methods Chi-square (CHI2), information gain (IG) and mutual information (MI) perform well in the feature selection procedure, there is probability that these methods also apply to measuring the term weight equally. Therefore, replacing the IDF with CHI2, IG and MI separately, three new STW methods are proposed, which are named as TF-CHI2, TF-IG and TF-MI (Debole and Sebastiani 2003). In view of that the number of categories containing the specific term may contain useful information for TC task, then two STW methods named TF-ICF (Wang and Zhang 2013), TF-IDF-ICF (Ren and Sohrab 2013) generated as a result. Chen et al. (2016) proposed that the traditional TF-IDF is not fully effective for TC task, in order to make some improvements, based on a new statistic model, a new STW method named TF-IGM and its variants were proposed which claims that this method can make full use of the fine-grained term distribution across different classes of texts. In addition, there are also a variety of STW methods based on different models like TF-OR (Altınçay and Erenel 2010) and TF-PB (Liu et al. 2009).

Intuitively, the STW methods should have performed better than UTW ones in terms of text classification performance because they make full use of the predefined category information. Actually, as the representative of UTW methods, TF-IDF shows better performance than some STW methods, this phenomenon is conflict with our intuition (Lan et al. 2009; Quan et al. 2011). Aiming at this problem, we have analyzed the above listed supervised term weighting methods and found that part of them become invalid under some special circumstances. For example, the category frequency (CF) part of TF-ICF and TF-IDF-ICF, it represents the total number of categories whose documents contain the chosen term. In the two models, the number of documents containing the chosen term in a specific category has no effect on the term weight, that is to say, one document or ten documents belong to the same category in which the chosen term appears is regarded as no difference, this is obviously unreasonable. To eliminate this defect, TF-IDF-ICSDF (Ren and Sohrab 2013) was proposed by implementing a new model named inverse class-space-density frequency. However, it will degenerate into TF-IDF when the number of documents in each category is the same (Chen et al. 2016). It is not a unique instance, a similar situation will also happen to TF-IGM which makes it become invalid when the number of different kinds of documents meets certain conditions. Hence, in order to offset this shortcoming, a novel method named TF-IGM$_{\text{imp}}$ (Dogan and Uysal 2019) was proposed by adding a ratio to the initial TF-IGM. Among these STW methods, term frequency-relevance frequency (TF-RF) proposed by Lan et al. (2009) is considered as an outstanding method with reasonable theoretical explanation and good classification performance. More importantly, similar failure circumstances will not happen in TF-RF. It is noticeable that most of the listed STW methods adopt the same logarithmic processing borrowed from TF-IDF to their respective global parts, but it is not clear whether the logarithmic processing is beneficial to the performance of TC or not. To explore this problem, two improved methods named TF-ERF and ETF-RF are proposed by strengthen the RF part and TF part separately. As a result, the method TF-ERF is proved to be more helpful to the improvement of TC and it shows certain advantages over other term weighting methods.

The rest of this paper is arranged as follows. Section 2 points out problems existing in the related work. Section 3 proposes improved term weighting methods based on TF-RF. Section 4 introduces the experimental settings. Section 5 analyzes the experimental data in detail. Section 6 concludes this paper.

2 Analysis about current term weighting schemes

In this section, we conduct analysis about some existing term weighting schemes with good TC performance. Meanwhile, the existing problems are also raised. For convenience, the notations utilized in this study are first presented in Table 1 and seven existing representative term weighting methods are shown in Table 2. By the way, the mathematical forms of these term weighting schemes are all not normalized.

2.1 Unsupervised term weighting method

As a widely used UTW method, TF-IDF shows a good TC performance, which is consisted of a local part and a global part, named as TF and IDF separately. TF supports the assumption that the more frequent a term appears in a specific document, the greater contribution it makes to the representation of this document. That is to say, the term is more important to this document (Lakshmi and Baskar 2019). This conclusion can be intuitively obtained, but is it really reasonable? We can imagine a scenario like this, there is a document in which a term (t_1) appears 20 times while another term (t_2) appears only once, can we directly come to the conclusion that t_1 is 20 times more important than t_2? The answer is obviously not. By the way, we have mentioned that TC is a task whose goal is to classify a variety of documents into corresponding categories according to their content. It is obvious that terms which possess the ability to distinguish between different types of documents ought to be distributed a higher term weight. However, the TF weight only reflects the ability of a term in representing the document containing this term (Zhang et al. 2019), this is against with the target of TC task. The importance of a term is measured only in the dimension of term-document and TC task in only limited in a certain text without considering the relationship between all texts, which results in a bad performance of classification. Therefore, we can conclude that the single TF cannot make meaningful contribution as a term weighting method to serve the TC task.

Table 1 Notations and descriptions

Notation	Description
A_{ij}	Number of documents that contain feature t_j in category c_i
B_{ij}	Number of documents that do not contain feature t_j in category c_i
C_{ij}	Number of documents that contain feature t_j but do not belong to category c_i
D_{ij}	Number of documents that do not contain feature t_j and do not belong to category c_i
N	Total number of documents in the training set, $N=A_{ij}+B_{ij}+C_{ij}+D_{ij}$
$tf(t_j, d_k)$	The term frequency of feature t_j in document d_k
$df(t_j)$	Number of documents containing t_j, and $df(t_j)=A_{ij}+C_{ij}$
m	Total number of categories
λ	λ is an adjustable coefficient and its default value is set to 7.0, and $\lambda\in[5.0,9.0]$
f_{ij}	Frequencies of t_j's occurring in different categories, which are sorted in descending order with i ($i=1,2,\ldots,m$) being the rank
$n_{c_i}(t_j)$	Number of documents containing the term t_j in a certain category c_i
N_{c_i}	Total number of documents in a certain category c_i
k_{tf}	Coefficient added to the TF part of TF-RF
k_{rf}	Coefficient added to the RF part of TF-RF

Table 2 Term weighting methods to be compared

Term weighting method	Name	Mathematical form
Unsupervised	TF-IDF	$tf(t_j, d_k) \times \log\left(\frac{N}{df(t_j)}\right)$
Supervised	TF-CHI2	$tf(t_j, d_k) \times \frac{N \times (A_{ij} \times D_{ij} - B_{ij} \times C_{ij})^2}{(A_{ij}+B_{ij}) \times (C_{ij}+D_{ij}) \times (A_{ij}+C_{ij}) \times (B_{ij}+D_{ij})}$
	TF-MI	$tf(t_j, d_k) \times \log\left(\frac{N \times A_{ij}}{(A_{ij}+B_{ij}) \times (A_{ij}+C_{ij})}\right)$
	TF-OR	$tf(t_j, d_k) \times \log\left(\frac{A_{ij} \times D_{ij}}{B_{ij} \times C_{ij}}\right)$
	TF-RF	$tf(t_j, d_k) \times \log_2\left(2 + \frac{A_{ij}}{\max(1, C_{ij})}\right)$
	TF-IDF-ICSDF	$tf(t_j, d_k) \times \left(1 + \log\left(\frac{N}{df(t_j)}\right)\right) \times \left(1 + \log\left(\frac{m}{\sum_{i=1}^{m}\left(n_{c_i}(t_j)/N_{c_i}\right)}\right)\right)$
	TF-IGM	$tf(t_j, d_k) \times \left(1 + \lambda \times \frac{f_{j1}}{\sum_{i=1}^{m} f_{ji} \times i}\right)$

In order to make up the defect of the single TF part, the IDF part is introduced to balance the excessive influence of TF on the final term weight (Shang et al. 2013). With regard to a specific term, its IDF weight can be defined as

$$\mathrm{IDF}(t_j) = \log\left(\frac{N}{df(t_j)}\right) \tag{1}$$

It can be seen from Eq. (1) that the smaller the value of $df(t_j)$, the larger the value of $\mathrm{IDF}(t_j)$, which holds the assumption that the less documents in which a term appears, the more significant the term will be. By introducing the IDF part to the TF part, term weighting method TF-IDF was generated as a result. Intuitively, this new model is more appropriate as it takes both local part and global part into account, and the actual TC performance is also consisted with the intuition. However, there are still some defects existing in this method, a simple example can be given to explain this. Assuming that there are four kinds of texts in the training text sets and they all consist of 60 documents. Four terms with the same term frequency are selected and their document frequency can be represented as {60, 0, 0, 0}, {30, 30, 0, 0}, {20, 20, 20, 0} and {15, 15, 15, 15} separately. It can be easily seen that the ability to discriminate different classes is ranked as $t_1 > t_2 > t_3 > t_4$. However, the four terms are assigned the same IDF value, which is contrary to our intuition. Actually, although the global factor is taken into account in the TF-IDF method, its incomplete application of global weight leads to these extreme cases occasionally which make the method invalid. This can be attributed to the absence of available category information in the process of computing term weight, so it is essential to measure the importance of a term in the document-category dimension.

2.2 Supervised term weighting method

Due to the TC task is a supervised learning process aiming at classifying different types of documents, it is natural to utilize STW methods to match the TC process (Dogan and Uysal 2019). However, there are some defects in most existing STW methods, which will make the methods invalid in some extreme situations. There we take TF-IDF-ICSDF and TF-IGM as examples to illustrate the extreme situations that can lead to the failure of the term weighing methods. For TF-IDF-ICSDF, assuming that there are four kinds of texts in the training text sets and they all consist of 100 documents. Four terms with the same term frequency are selected and their document frequencies can be represented as {60, 0, 0, 0}, {50, 50, 0, 0}, {40, 40, 40, 0} and {20, 20, 20, 20} separately. It can be easily seen that the ability to discriminate different classes is ranked as $t_1 > t_2 > t_3 > t_4$. However, the four terms are assigned the same ICSDF value because the number of documents in four categories are the same, which can be calculated by the formula given in Table 2. This leads to the fact that the ICSDF part has no effect on term weighting, so the TF-IDF-ICSDF model degenerate into TF-IDF model. For TF-IGM, similarly, assuming a scenario that there are four kinds of texts in the training sets and they all consist of 100 documents. Four terms with the same term frequency are selected and their document frequency can be represented as {90, 0, 0, 0}, {60, 0, 0, 0}, {30, 0, 0, 0} and {10, 0, 0, 0} separately. Intuitively, the order of class distinguishing power must be $t_1 > t_2 > t_3 > t_4$, but by the formula shown in Table 2, the standard IGM values of t_1, t_2, t_3, t_4 are all equal to 1 which represents that all the four terms own the same distinguishing power, this is obviously unreasonable. As can be seen from the extreme cases of the above two examples, the two STW methods have certain requirements for texts and lacks robustness for different types of texts.

2.3 Relevance frequency

Apparently, the above-mentioned term weighting methods regard TC task looks as a multi-label classification problem, but in fact, what we need to do is just separate the chosen category from others instead of taking every unrelated category into account. Following this thought, TF-RF was proposed which simplifies the multi-label classification problem into multiple independent binary classification problems (Lan et al. 2009). Specifically speaking, in the training text corpus, when computing the weight of a specific term, the category of document containing this term is tagged as the positive category and the other categories are uniformly classified as the negative category (Lan et al. 2009). TF-RF supports the idea that the more concentrated the chosen term is in the positive category than in the negative category, the greater ability it possesses to select a correct category for the documents containing it. Besides, TF-RF also holds the assumption that the importance of a term is only related to the documents containing it (Lan et al. 2009). Based on the two thoughts mentioned above, the term weighting method TF-RF was proposed which can be presented with the form shown in Table 2. It can be noticed that the logarithmic processing is also adopted on the global part of TF-RF.

Considering that logarithmic processing has certain restrictions on its parameters, we need to limit the parameters with a certain range to prevent failure. In terms of TF-RF, firstly, in order to avoid the RF part being meaningless when the chosen term doesn't appear in positive documents, i.e., a=0, the minimum of the independent variable is limited to 2. Meanwhile, the base is also set to 2 to match the independent variable. With this processing, the RF value becomes 1 when a=0, it is logical that the term weight depends on the TF weight entirely in this case. Secondly, the minimum of the denominator is limited to 1 for the purpose that preventing the RF value becomes infinite. It seems that the processed formula of TF-RF shown in Table 2 solves the problems existing perfectly. However, due to the TF part depends on the frequency a certain term appears in a document, it is bound to be affected by the length of the document. Therefore, the normalization is always performed to eliminate the influence of different text lengths when computing the term weight. As a result, the normalized TF-RF formula can be defined as.

$$\mathrm{TFRF}(t_j, d_k, c_i) = \frac{tf(t_j, d_k) \cdot \log_2(a/\max(1,c)+2)}{\sqrt{\sum_{j=1}^{n} \left(tf(t_j, d_k) \cdot \log_2(a/\max(1,c)+2)\right)^2}} \tag{2}$$

We have analyzed that the introduction of IDF is to balance the excessive influence of the single TF on the term weight. For the same purpose, the global part RF is introduced to generate a more reasonable term weighting model and the logarithmic processing borrowed from IDF part is adopted on the RF part directly. Although TF-RF presents outstanding performance in TC task, it is not clear whether the logarithmic processing adopted on the RF part contribute to the good performance or not. Maybe there is possibility that the logarithmic processing restricts TF-RF to achieve better TC performance. Therefore, a problem can be naturally proposed, that is, "does the logarithmic processing adopted on IDF also apply to RF?" In terms of this problem, related analysis will be carried out in the next section.

3 Proposed method

In this section, aiming at solving the problem mentioned in the previous section, two assumptions are proposed along with corresponding strategy. In addition, the reliability of the two assumptions are also discussed at the end of the section.

3.1 Motivation

Intuitively, if terms are assigned the same term weight, the conclusion that they possess equivalent importance can be drawn intuitively. But is the actual situation really as simple as it seems? As mentioned before, most term weighting methods consist of two parts, local part and global part, and the term weight is the product of their respective weights of the two parts. It seems that the term weight gives a comprehensive consideration from the two parts. Conversely, we can interpret it from another perspective, that is, the respective characteristics of the two parts cannot be fully shown because the size relation between local weight and global weight is neglected when multiplying them. Here an example can be given to make an explanation about this. Assuming that there are two terms named t_1 and t_2, the TF weights of t_1 and t_2 are 1 and 100 separately and the RF weights are 100 and 1 separately, it is obvious that their term weights are the same. However, there is great difference between the two terms because the RF weight has absolute dominance over term weight for t_1 while t_2 is just the opposite, which leads to the influence of the weak part on the term weight is suppressed by the strong part.

In terms of the TF-RF model we are studying, it has a relatively good performance for TC task, but we don't know whether the logarithmic processing adopted on the global part RF of TF-RF borrowed from TF-IDF contributes to the excellent performance or not. Besides, there is even possibility that the influence of TF part and RF part on the term weight is out of balance due to the logarithmic processing, resulting that the characteristic of one part is concealed by the other part. As a result, TF-RF cannot measure term weight from the two parts appropriately. Based on this problem, two assumptions can be proposed as following.

Assumption 1 The RF part is weakened too much through logarithmic processing. In other words, the impact of the TF part is overemphasized, which results in the RF part is suppressed by the TF part in contributing the discriminating power to the selected term.

Assumption 2: The weakening of RF part is not enough only by logarithmic processing. The RF part still occupies excessive dominance on the term weight, which leads to suppressing the effect of the TF part in contributing the discriminating power to the selected term.

The two assumptions mentioned above can be presented as Fig. 1. It can be seen from the figure that some terms with different characteristics are assigned the same weight for initial TF-RF model. Among these terms, some are bias to *tf* and others prefer *rf*. For Assumption 1, due to the RF part has a greater dominance on the term weight which will restrain the expression of the characteristic of the TF part, so the dominance of TF part should be strengthened to balance the overemphasis on the RF part in order to make the method more reasonable. For Assumption 2, it is just opposite to Assumption 1. Therefore, the impact of TF part should be enhanced to balance the excessive influence caused by the RF part. Naturally, the result is also contrary to the result of Assumption 1. In view of the two assumptions, their respective strategies will be analyzed in detail in the next section.

3.2 Improvement approaches

In view of the two assumptions mentioned above, in order to gain a more reasonable term weighting method, the part which is suppressed by the other part for the dominance on the term weight, is ought to be strengthened. And there are two strategies shown in Fig. 2 which can be used to achieve this purpose.

Strategy 1: Adding a coefficient k as multiple to the certain part ($k{\cdot}tf$ or $k{\cdot}rf$, $k{>}1$).

Strategy 2: Adding a coefficient k as power exponent to the certain part (tf^k or rf^k, $k{>}1$).

For assumption 1, adopting strategy 1 to the normalized TF-RF shown as Eq. (2). Then the deformed formula can be defined as Eq. (3), in which $k{=}k_{rf}$.

$$\begin{aligned}\mathrm{TFRF}(t_j,d_k,c_i)* &= \frac{tf(t_j,d_k)\cdot k_{rf}\cdot \log_2(a/\max(1,c)+2)}{\sqrt{\sum_{j=1}^{n}\left(tf(t_j,d_k)\cdot k_{rf}\cdot \log_2(a/\max(1,c)+2)\right)^2}}\\ &= \frac{tf(t_j,d_k)\cdot \log_2(a/\max(1,c)+2)}{\sqrt{\sum_{j=1}^{n}\left(tf(t_j,d_k)\cdot \log_2(a/\max(1,c)+2)\right)^2}}\\ &= \mathrm{TFRF}(t_j,d_k,c_i)\end{aligned} \tag{3}$$

It can be noticed from Eq. (3) that the coefficient k_{rf} added to the RF part will be offset, so strategy 1 becomes meaningless. Then adopting strategy 2 to Eq. (2), the deformed formula can be defined as Eq. (4). Obviously, the failure will not happen in strategy 2. So after the identification, strategy 2 is chosen as a feasible to strengthen the RF part, and the improved method Eq. (4) is named as term frequency-exponential relevance frequency (TF-ERF).

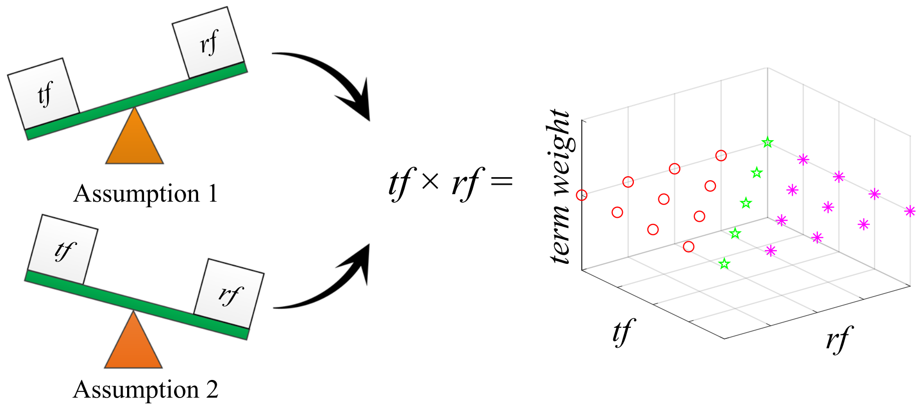


Fig. 1 Two assumptions based on TF-RF

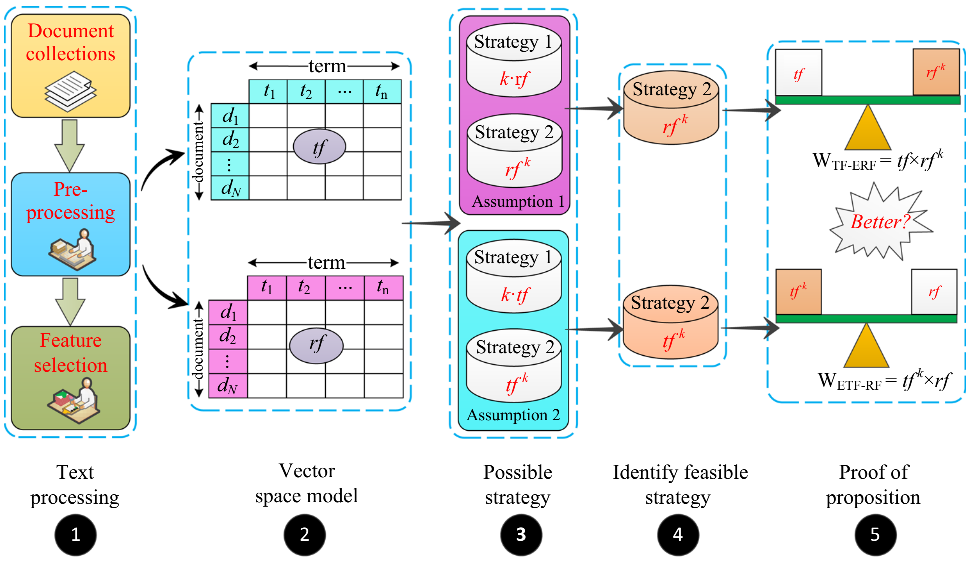


Fig. 2 Framework of the proposed method

$$\mathrm{TFERF}(t_j, d_k, c_i) = \frac{tf(t_j, d_k) \cdot (\log_2(a/\max(1,c)+2))^{k_{rf}}}{\sqrt{\sum_{j=1}^{n}\left(tf(t_j, d_k)\cdot(\log_2(a/\max(1,c)+2))^{k_{rf}}\right)^2}} \tag{4}$$

For assumption 2, the same conclusion can be drawn as proposition 1. Strategy 1 will also become invalid due to the coefficient k_{tf} added to the TF part is offset. As a result, strategy 2 is selected to strengthen the TF part. Adopting approach 2 to Eq. (2), then the deformed formula can be defined as Eq. (5), which is named as exponential term frequency relevance frequency (ETF-RF), in which $k=k_{tf}$.

$$\mathrm{ETFRF}(t_j, d_k, c_i) = \frac{tf(t_j, d_k)^{k_{tf}} \cdot \log_2(a/\max(1,c)+2)}{\sqrt{\sum_{j=1}^{n}\left(tf(t_j, d_k)^{k_{tf}}\cdot\log_2(a/\max(1,c)+2)\right)^2}} \tag{5}$$

3.3 Qualitative analysis of two improved methods

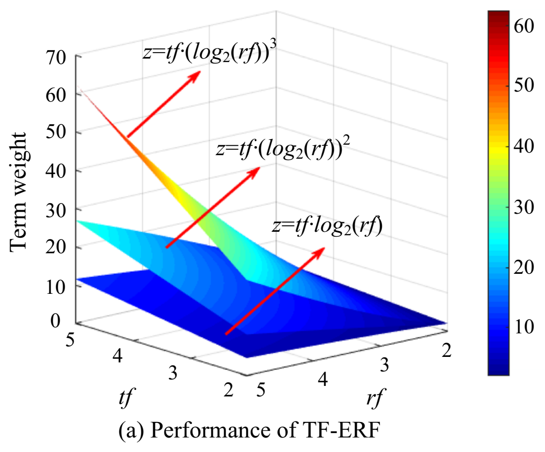
In terms of the methods proposed in previous section, their actual effect performance can be presented as Fig. 3. Figure 3a, b present the result for different values of *tf* and *rf* by adding a coefficient k_{rf} (k_{rf}=1, 2, 3) or k_{tf} (k_{tf}=1, 2, 3) to the corresponding RF or TF part separately. It can be seen from the figure that the slope of the surface shown in Fig. 3b is steeper than that of Fig. 3a for the same k_{tf} and k_{rf}. In addition, we can also notice that there is little difference between the partial derivative in direction *tf* and that of *rf* for Fig. 3a though the coefficient k_{rf} is introduced to strengthen the RF part. While in Fig. 3b, the partial derivative in direction *tf* far exceeds that in direction *rf* due to the introduction of k_{tf}. This proves that the introduction of k_{tf} has greater influence on the term weight than k_{rf}. Due to the initial TF-RF is an excellent term weighting scheme which presents a good performance in TC task, the respective influence of the TF part and RF part on the term weight is reasonable to a certain extent. Therefore, the treatment to strengthen the influence of TF part on the term weight by adding a coefficient k_{tf} may greatly undermine this rationality and make the classification performance reduced. By the way, similar analysis and conclusion also mentioned in Lan et al. (2009), which is consistent with what we have analyzed. So if the initial TF-RF can be improved, there is a great possibility that the dominance of the RF part on the term weight ought to be enhanced, that is to say, TF-ERF may be more helpful to the improvement of TF-RF compared with ETF-RF.

4 Experimental setup

In this section, the experimental datasets, feature selection method, classification algorithms and evaluation of the performance measures used in our experiments are introduced successively.

4.1 Experimental datasets and pre-processing

In order to verify whether the improved methods proposed before is helpful to improve the performance of TC task, a series of experiments are conducted. Experiments are conducted on two datasets, Reuters-21587 corpus and WebKB corpus.



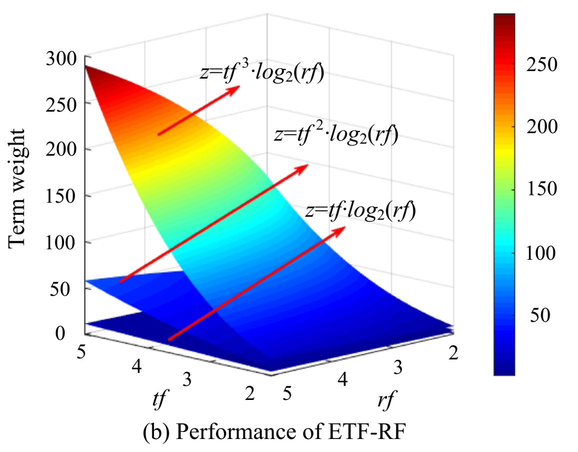


Fig. 3 Performance adding different k_{tf} and k_{rf} to TF part and RF part separately

4.1.1 Reuters-21587 corpus

The first dataset used in our experiments is the Reuters-21587 corpus, which is widely used in the text classification field. This English corpus contains 90 classes of news documents. The top 8 largest classes are selected for the text classification task. The reduced corpus contains 7674 documents which have been divided into a training set with 5485 documents and a test set with 2189 documents (shown in Table 3).

The pre-processing is an important step which will make an effect on the result of text classification to a certain degree. In this step, punctuation marks, numbers and other symbols are all removed. Furthermore, in order to reduce the size of the feature set, the terms which appear less than two times are discarded. At last, all letters are converted to lowercase and words are stemmed using Porter's stemmer (Porter 2006).

Finally, a total of 8541 distinct terms left build the feature set. After the pre-processing stage, we acquire the document-term matrices of training set and test set, which are 5485×8541 and 2189×8541, respectively.

Table 3 Data description on reuters-21578

No	Class label	Training data	Test data
1	Acq	1596	696
2	Crude	253	121
3	Earn	2840	1083
4	Grain	41	10
5	Interest	190	81
6	Money-fx	206	87
7	Ship	108	36
8	Trade	251	75

4.1.2 WebKB corpus

The second dataset used in our experiments is the WebKB corpus, this English corpus contains 4199 documents which have been divided into a training set with 2803 documents and a test set with 1396 documents (shown in Table 4). In the step of pre-processing, the same pre-processing mentioned previous is also carried out on the WebKB corpus. Eventually, a total of 7061 distinct terms (features) are selected to build the feature set, the document-term matrices of training set and test set are 2803×7061 and 1396×7061, respectively.

4.2 Feature selection

The initial feature set is achieved after the pre-processing of the datasets. However, the feature set cannot be directly applied to experiments due to the fact that masses of features are meaningless for TC task. In addition, these invalid features may also cause harmful effect to the classifier, which will lead to a bad TC performance (Meng et al. 2011; Wang et al. 2015). Therefore, it is essential to select the most effective features on the promise of not sacrificing the performance of TC task. As we know, feature selection (FS) is such a task aiming at building a more reasonable model for TC task by selecting relatively valuable terms. Presently, there are many methods that can be applied to feature

Table 4 Data description on WebKB

No	Class label	Training data	Test data
1	Project	336	168
2	Course	620	310
3	Faculty	750	374
4	Student	1097	544

selection. For example, three typical methods have been mentioned in Sect. 1 named CHI2, IG and MI separately. Among them, CHI2 is recognized to be the most effective method due to its outstanding performance (Liu and Yu 2005; Taşcı and Güngör 2013; Şahin and Kılıç 2019), so it is adopted to select features in this study.

4.3 Classification algorithms

Bayes classifier is the general name of a type of classification algorithm which are all based on Bayes law. Among them, Naive Bayes (NB) is the most common and simple one which is widely used in the field of TC (Yang and Pedersen 1997). NB treats all features as independent and no interaction and it regards the TC task as a probability problem that the class with the highest probability will be selected as the final category (Friedman et al. 1997; Ning et al. 2021). Based on this idea, the number of parameters to be estimated is greatly reduced, which simplifies the requirements of feature space and the calculation of solution to a great extent. As a result, the simplicity and efficiency will be greatly promoted when using NB classifier. In view of these advantages, NB classifier is utilized in our experiments and its algorithm interpretation is presented as Eq. (6).

Assuming that the document d_k consisting of a certain number of terms which can be represented as $d_k=(t_1, t_2, \ldots, t_n)$. The probability that the document d_k belongs to category c_i can be defined as Eq. (6), in which $P(d_k)$ denotes a constant for all documents. More details can be seen in Farid et al. (2014) and Ilinskas and Litvinas (2020). Furthermore, default parameter settings for NB classifier in this research.

$$P(c_i|\mathrm{d}_k) = \frac{P(c_i)\prod_{j=1}^{n} P(t_{jk}|\mathrm{c}_i)}{P(d_k)} \tag{6}$$

4.4 Evaluation of the performance

In order to verify the effectiveness of the improved methods, the TC performance need to be evaluated with a certain standard of measurement. Among masses of evaluation indexes, precision and recall are two popular measures for evaluating the performance of TC task. Precision denotes the proportion of correct assignments among all the test documents that should be assigned to the target category and recall represents the proportion of correct assignments among all the test documents assigned to the target category. However, neither of them can be directly used to evaluating the performance for the reason that the higher level of one indicator may be obtained at the expense of sacrificing the level of the other one. As a result, a new measure named F1 was proposed, in which precision and recall are combined together and assigned the same importance. The precision, recall and the F1 measure can be defined as follows and the explanations of corresponding notations in the formula are listed in the.

Table 5.

$$P(c_i) = \frac{\mathrm{TP}(c_i)}{\mathrm{TP}(c_i) + \mathrm{FP}(c_i)} \tag{7}$$

$$R(c_i) = \frac{\mathrm{TP}(c_i)}{\mathrm{TP}(c_i) + \mathrm{FN}(c_i)} \tag{8}$$

$$F(c_i) = \frac{2 \cdot P(c_i) \cdot R(c_i)}{P(c_i) + R(c_i)} = \frac{2 \cdot \mathrm{TP}(c_i)}{2 \cdot \mathrm{TP}(c_i) + \mathrm{FP}(c_i) + \mathrm{FN}(c_i)} \tag{9}$$

In general, the F1 measure is estimated from two ways, micro-averaged F1 (micro-F1) and macro-average F1 (macro-F1). The macro-F1 and micro-F1 can be defined as Eq.(10) and Eq.(11).

$$\mathrm{macro}{-}F1 = \frac{1}{m}\sum_{i=1}^{m} F_1(c_i) \tag{10}$$

$$\mathrm{micro}{-}F1 = \frac{2 \cdot \sum_{i=1}^{m} \mathrm{TP}(c_i)}{2 \cdot \sum_{i=1}^{m} \mathrm{TP}(c_i) + \sum_{i=1}^{m} \mathrm{FP}(c_i) + \sum_{i=1}^{m} \mathrm{FN}(c_i)} \tag{11}$$

5 Experiment results and analysis

In this section, orthogonal experimental design will be firstly described, by which one of ETF-RF and TF-ERF will be proved to be more helpful to the improvement of the initial TF-RF. Then, the chosen method (i.e., TF-ERF or ETF-RF) will be compared with other existing term weighting methods (listed in Table 2) in order to verify its effectiveness for the improvement of TC performance.

5.1 Performance comparisons between TF-ERF and ETF-RF

The first group of experiments is to distinguish which one of TF-ERF and ETF-RF is more helpful to the improvement of

Table 5 Contingency table for category c_i

	True label c_i	True not c_i
Predicted label c_i	True positive (TP)	False positive (FP)
Predicted not c_i	False negative (FN)	True negative (TN)

the initial TF-RF. Orthogonal experimental design is a kind of experimental design method to study multi-factors and multi-levels. According to orthogonality, some representative points are selected from the comprehensive test to carry out the test. The main tool of orthogonal test design is the orthogonal table, testers can seek the corresponding orthogonal table according to the requirements of the number of factors, the level of factors and whether there is interaction, and then select some representative points from the comprehensive test to test based on the orthogonality of the orthogonal table. In this study, the orthogonal table L25 (5^6) is selected to arrange the combinations of parameters at different level. Through the experiment and analysis of different parameter sets of k_{tf} and k_{rf}, it can be obtained that when k_{tf} and k_{rf} take values in the {1,2,3,4,5} set, the proposed weight distribution model has good classification results for the two test text sets, so this set is selected as a specific parameter set. When k_{tf} and k_{rf} are within the selected parameter set, the term weighting model proposed in this paper has good classification performance and robustness for different text sets.

The figure presents the performance obtained on Reuters-21578 dataset and WebKB dataset separately. It can be seen from the figure that the classification performance will deteriorate rapidly with the increase in k_{tf}. In contrast, the change of k_{rf} doesn't have great influence on the performance of TC as k_{tf} shows, while the increase in k_{rf} presents a positive impact on the improvement of the classification performance. This proves that the improved method TF-ERF is beneficial to the improvement of initial TF-RF model, which is consistent with the analysis result mentioned in Sect. 3.3.

In terms of Fig. 4, taking both macro-F1 and micro-F1 into account, the best classification performances can be observed under k_{tf}=1 and k_{rf}=5 for the Reuters-21578 dataset as well as k_{tf}=1 and k_{rf}=2 for the WebKB dataset. Therefore, the parameters added to TF-ERF is determined, and the improved method TF-ERF will be compared with other term weighting schemes in the following experiments.

5.2 Performance comparisons of existing methods

The second group of experiments are to verify the effectiveness of TF-ERF by comparing its performance with other term weighting schemes listed in Table 2. The classification experiments are carried out on two text test sets, Reuters-21578 corpus and WebKB, which has been introduced before. The two text test sets get 8000 and 7000 features separately after feature selection. In order to reflect the text classification performance of each term weighting model on different feature numbers more succinctly and intuitively, corresponding analysis of the number of features is carried out. Finally, it is determined to divide the total features under two test text sets with the step size of 1000.

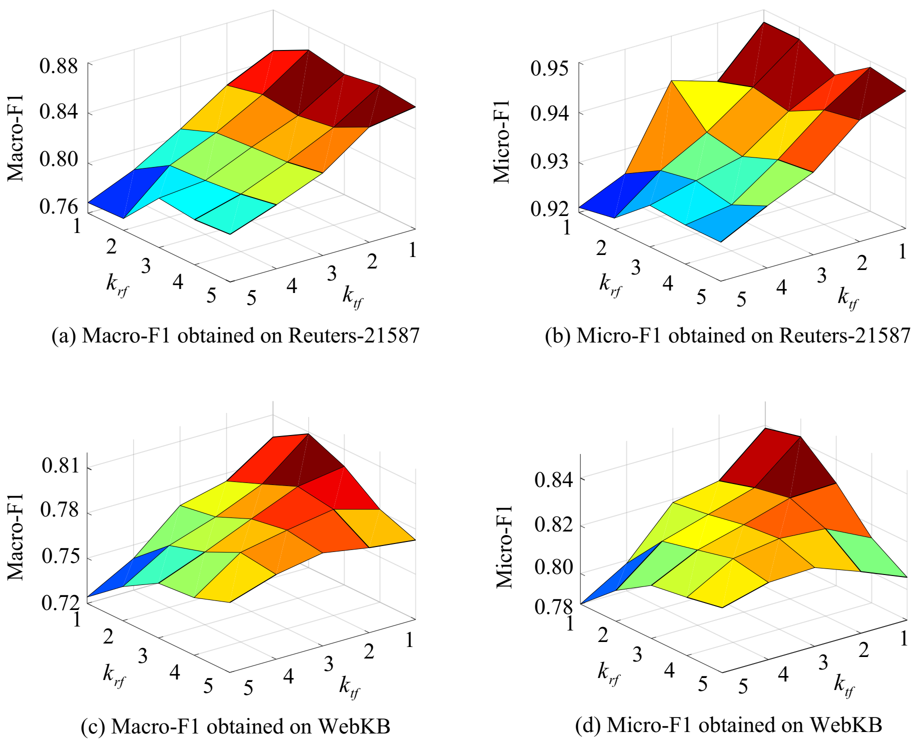
Figure 5 shows the experiment performance of TC on the Reuters-21578 corpus. It can be seen from the figure that TF-IDF and TF-IDF-ICSDF present the worst performance in all feature sets for both macro-F1 and micro-F1. Especially at a small feature set (less than 200), the performance of the two schemes is far worse than the other ones. Meanwhile, the rest schemes perform well even when the number of features is small.

In terms of macro-F1, almost all term weighting schemes reach their peaks at a feature set around 1000 and TF-ERF obtain the best performance at the peak compared with others. On the whole, TF-ERF is superior to other schemes when the number of features is less than 7000. In addition, it can be seen that TF-RF does not show advantages over other schemes and even inferior to TF-CHI2 and TF-MI in most feature sets. By contrast, it is obvious that TF-ERF is very effective in improving the performance of TC as an improved method of TF-RF.

In terms of micro-F1, these schemes do not reach the peak at the same feature set as the figure of macro-F1 shows. But there are also corresponding turning points at a feature set around 1000. After that, the micro-F1 of some schemes begin to decrease like TF-IDF and TF-IDF-ICSDF, while the rest schemes continue to maintain an increasing trend at a slower speed. It can be seen that TF-ERF also shows evident advantage over other schemes for most feature sets in addition to the situation when the number of features is 800. TF-ERF reaches its peak when the number of features is around 5000 and the peak presents the best performance of all the term weighting schemes for all the features.

Figure 6 shows the experimental performance of TC on the WebKB corpus. It can be seen from the figure that most schemes show poor performance when the number of features is relatively small, which is different from what Fig. 5 shows. In addition, TF-ERF does not present great advantage over other schemes as it shows in Fig. 5. However, it cannot be neglected that TF-ERF has a better performance in a certain feature set and the improvement relative to the initial TF-RF.

In terms of macro-F1, it can be seen that TF-CHI2, TF-MI and TF-OR almost maintain the growing trend with the increase in the number of features, and other schemes start to decrease when reaching their respective peaks. As far as TF-ERF is concerned, it is superior to other schemes when the number of features falls in [200, 3000]. After the number of features exceeds 3000, the performance of TF-ERF begins to deteriorate and be surpassed by TF-MI, TF-OR and TF-CHI2 successively. However, we should notice that TF-ERF reaches its peak at a feature set around 2000 and the performance of this point is the best among all the



(a) Macro-F1 obtained on Reuters-21587 (b) Micro-F1 obtained on Reuters-21587

(c) Macro-F1 obtained on WebKB (d) Micro-F1 obtained on WebKB

Fig. 4 Classification performance according to different parameters k_{tf} and k_{rf} under two corpora

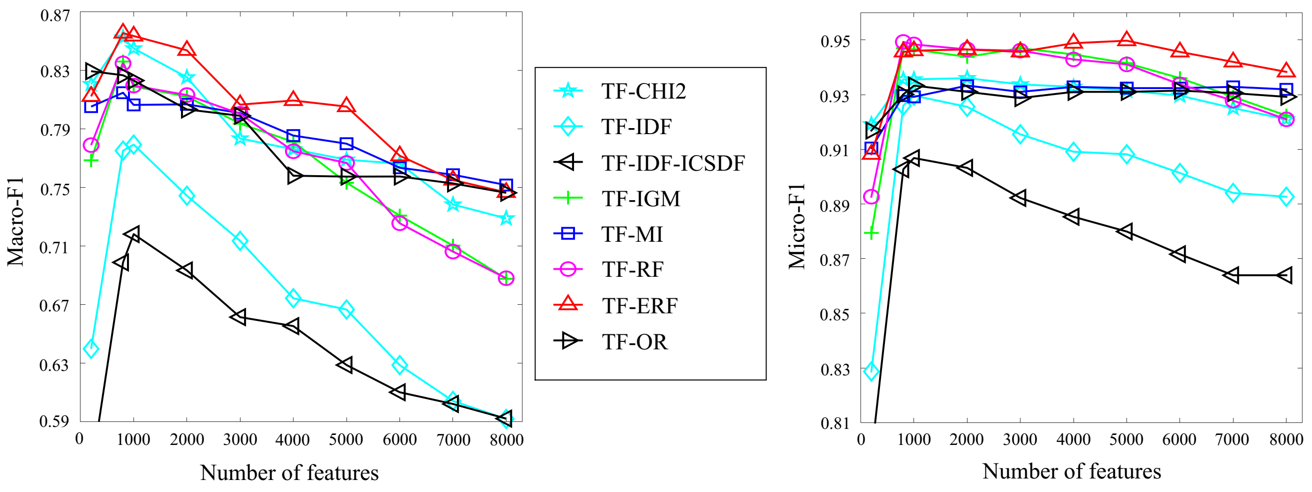


Fig. 5 Macro-F1 and Micro-F1 measure of the eight weighting approaches on the Reuters-21578 corpus with different numbers of features

schemes. In addition, as an improved method of the initial TF-RF, it also shows a better performance than TF-RF for the whole feature sets.

In terms of micro-F1, the curves become much tighter compared with the left figure which means that the gap of performance between different schemes has become smaller. From the figure we can see that TF-IGM, TF-RF and TF-ERF obtain better performance than the rest schemes for almost the whole feature sets. There is almost no difference in the performance of these three schemes when the number of features is less than 1500. After that they all present a descent trend one after another. Among them, due to the attenuation of TF-ERF is less than the other two schemes, so the performance of TF-ERF is superior to TF-RF and TF-IGM. TF-ERF reaches its peak at a feature set around 3000 and it also represents the best performance of all these eight schemes for all feature sets.

From the above two groups of experiments, it can be seen that the proposed term weighting method TF-ERF achieve better TC performance compared with other methods in most feature numbers. Although the performance of TF-ERF is surpassed by partial methods in some specific feature sets. It is undeniable that TF-ERF still has obvious advantages on the whole. Meanwhile, the best TC

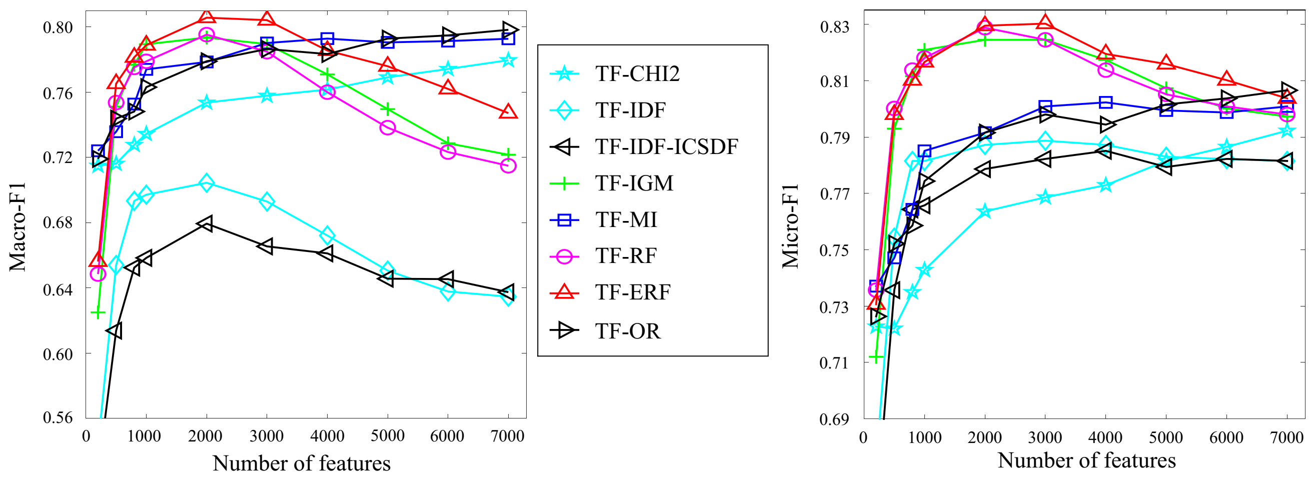


Fig. 6 Macro-F1 and Micro-F1 measure of the five weighting approaches on the WebKB corpus with different numbers of features

performances are always obtained by TF-ERF, which proves that the TC ability of TF-ERF has a higher upper limit. Furthermore, it should be noted that all the term weighting methods have poor TC performance when using fewer features for TC experiments. This shows that most useful features have been filtered out, and the remaining features are not enough to support the TC task to obtain a better classification result. For different term weighting methods, the changing trend of their TC performance is various with the increase in features, and this trend will also be influenced by the selected text set. For example, in terms of the text classification experiment under the Reuters-21587 corpus, the best performances of most term weighting methods are obtained around 1000 features. After the number of features exceeds 1000, most of them show an obvious downward trend. For WebKB dataset, the best performances of most term weighting methods are around 3000 features. After the number of features exceeds 3000, the TC performances of most of these methods show a weak downward trend, but there are exceptions, such as TF-CHI2, TF-MI, TF-IDF-ICSDF. With the increase in the number of features, the performances of these three methods are on the rise. And for the improved term weighting method TF-ERF, we can see that the changing trend of TC performances under different numbers of features are less affected by different text sets compared with other methods, and the best classification performances are all achieved by TF-ERF, which proves that TF-ERF has good text classification ability and robustness.

5.3 Performance improvements of proposed method over others

In order to understand the performance of the proposed scheme compared with other existing schemes more accurately, the specific experimental data of each scheme for Reuters-21578 dataset and WebKB dataset are listed in Tables 6 and 7 separately. The data in parentheses are the improvement of TF-ERF relative to other schemes. Among them, the data with the best performance in each chosen feature set are presented with boldface. In addition, the data with the best performance in the whole feature set is shown in bold and underlined.

It can be seen from Table 6 that most of the best performances of selected feature set focus on TF-ERF. For both macro-F1 and micro-F1, the best performance of TF-ERF is also the best performance of all the eight term weighting methods for the whole feature sets. As an improved method of TF-RF, the performance of TF-ERF is superior to that of TF-RF for each feature set. On the whole, the improvement growth relative to TF-RF raises with the increase in the number of features except for some special points. For Table 7, although TF-ERF doesn't show strong advantage over other schemes as Table 6 shows, in most cases, TF-ERF is still superior than other methods. Besides, what has not changed is that the best performance of TF-ERF still represents the optimal level of all the selected term weighting methods for the whole feature sets. In summary, the improved term weighting method TF-ERF, shows a better text classification performance over other term weighting methods and better robustness for text sets with different characteristics.

6 Conclusion

In this study, considering that the logarithmic processing of the global weight borrowed from TF-IDF may not adapt to other term weighting methods, e.g., TF-RF. So two improved term weighting methods based on TF-RF were proposed to explore this problem. First of all, two assumptions were given to explain the problems faced by TF-RF. In terms

Table 6 Macro-F1 and micro-F1 for reuters-21578 dataset

Classifier, #features	TF-ERF	TF-RF	TF-CHI2	TF-IDF	TF-IDF--ICSDF	TF-IGM	TF-MI	TF-OR
Macro-F1								
200	0.8123	0.7790	0.8206	0.6398	0.5550	0.7686	0.8053	**0.8295**
		(4.28%)	(−1.01%)	(6.95%)	(46.36%)	(5.69%)	(0.87%)	(−2.07%)
800	**0.8555**	0.8348	0.8524	0.7752	0.6989	0.8361	0.8148	0.8267
		(2.48%)	(0.36%)	(10.36%)	(22.41%)	(2.32%)	(5.00%)	(3.48%)
1000	**0.8535**	0.8197	0.8450	0.7793	0.7182	0.8190	0.8064	0.8231
		(4.12%)	(1.00%)	(9.52%)	(18.84%)	(4.21%)	(5.83%)	(3.69%)
2000	**0.8437**	0.8132	0.8253	0.7444	0.6934	0.8122	0.8068	0.8034
		(3.75%)	(2.23%)	(13.34%)	(21.68%)	(3.88%)	(4.57%)	(5.02%)
3000	**0.8066**	0.8004	0.7835	0.7136	0.6615	0.7936	0.8009	0.7990
		(0.78%)	(2.95%)	(13.04%)	(21.94%)	(1.64%)	(0.72%)	(0.95%)
4000	**0.8095**	0.7748	0.7761	0.6744	0.6554	0.7812	0.7855	0.7581
		(4.47%)	(4.30%)	(20.03%)	(23.51%)	(3.62%)	(3.05%)	(6.77%)
5000	**0.8053**	0.7668	0.7690	0.6667	0.6288	0.7533	0.7800	0.7575
		(5.02%)	(4.71%)	(20.79%)	(28.07%)	(6.90%)	(3.25%)	(6.31%)
6000	**0.7717**	0.7256	0.7663	0.6287	0.6102	0.7309	0.7635	0.7576
		(6.36%)	(0.71%)	(22.74%)	(26.47%)	(5.58%)	(1.07%)	(1.86%)
7000	**0.7551**	0.7063	0.7385	0.6042	0.6022	0.7104	0.7588	0.7528
		(6.92%)	(2.26%)	(24.97%)	(25.40%)	(6.30%)	(−0.48%)	(0.31%)
8000	0.7467	0.6881	0.7290	0.5920	0.5922	0.6877	**0.7517**	0.7464
		(8.52%)	(2.43%)	(26.14%)	(26.09%)	(8.58%)	(−0.67%)	(0.04%)
Micro−F1								
200	0.9082	0.8926	0.9191	0.8287	0.7976	0.8794	0.9105	**0.9169**
		(1.75%)	(−1.19%)	(9.59%)	(13.86%)	(3.27%)	(−0.26%)	(−0.95%)
800	0.9456	0.9493	0.9356	0.9260	0.9027	**0.9470**	0.9296	0.9301
		(−0.39%)	(1.07%)	(2.12%)	(4.76%)	(−0.14%)	(1.73%)	(1.67%)
1000	0.9461	0.9484	0.9356	0.9296	0.9068	**0.9466**	0.9292	0.9333
		(−0.24%)	(1.12%)	(1.77%)	(4.33%)	(−0.05%)	(1.82%)	(1.37%)
2000	**0.9466**	0.9465	0.936	0.9255	0.9032	0.9438	0.9333	0.9310
		(0.01%)	(1.13%)	(2.27%)	(4.80%)	(0.29%)	(1.42%)	(1.67%)
3000	0.9456	0.9461	0.9338	0.9155	0.8922	**0.9470**	0.9310	0.9287
		(−0.05%)	(1.27%)	(3.29%)	(5.99%)	(−0.14%)	(1.57%)	(1.82%)
4000	**0.9488**	0.9429	0.9328	0.9091	0.8853	0.9447	0.9328	0.9310
		(0.63%)	(1.72%)	(4.37%)	(7.18%)	(0.44%)	(1.72%)	(1.92%)
5000	**0.9497**	0.9411	0.9315	0.9082	0.8799	0.9415	0.9324	0.9310
		(0.92%)	(1.96%)	(4.57%)	(7.94%)	(0.88%)	(1.86%)	(2.01%)
6000	**0.9456**	0.9338	0.9296	0.9013	0.8716	0.9360	0.9324	0.9315
		(1.27%)	(1.73%)	(4.92%)	(8.49%)	(1.03%)	(1.42%)	(1.52%)
7000	**0.9420**	0.9278	0.9251	0.8940	0.8639	0.9292	0.9328	0.9306
		(1.53%)	(1.82%)	(5.37%)	(9.04%)	(1.38%)	(0.98%)	(0.22%)
8000	**0.9383**	0.9210	0.921	0.8926	0.8639	0.9223	0.9319	0.9292
		(1.88%)	(1.88%)	(5.12%)	(8.62%)	(1.74%)	(0.69%)	(0.98%)

of the two assumptions, feasible improvement strategies were identified and then applied to them separately. As a result, two methods named TF-ERF and ETF-RF were proposed. Then, through orthogonal experimental design, the improved term weighting method TF-ERF, which holds the assumption that the local part TF of TF-RF suppresses the impact of the global

Table 7 Macro-F1 and micro-F1 for WebKB dataset

Classifier, #features	TF-ERF	TF-RF	TF-CHI2	TF-IDF	TF-IDF--ICSDF	TF-IGM	TF-MI	TF-OR
Macro-F1								
200	0.6562	0.6485	0.7149	0.5433	0.5259	0.6250	**0.7240**	0.7190
		(1.18%)	(−8.21%)	(20.78%)	(24.78%)	(4.99%)	(−9.37%)	(−8.74%)
500	**0.7651**	0.7535	0.7161	0.654	0.6138	0.7511	0.7357	0.7437
		(1.54%)	(6.85%)	(16.99%)	(24.66%)	(1.87%)	(4.00%)	(2.88%)
800	**0.7813**	0.7751	0.7276	0.6933	0.6524	0.7768	0.7526	0.7481
		(0.80%)	(7.38%)	(12.69%)	(19.75%)	(0.58%)	(3.81%)	(4.44%)
1000	0.7887	0.7788	0.7343	0.6971	0.6584	**0.7893**	0.7739	0.7630
		(1.28%)	(7.41%)	(13.14%)	(19.80%)	(−0.07%)	(1.92%)	(3.37%)
2000	**0.8055**	0.7952	0.7535	0.7044	0.6794	0.7934	0.7784	0.7789
		(1.30%)	(6.91%)	(14.36%)	(18.57%)	(1.53%)	(3.49%)	(3.42%)
3000	**0.8040**	0.7847	0.7578	0.6929	0.6654	0.7894	0.7900	0.7864\
		(2.47%)	(6.10%)	(16.04%)	(20.83%)	(1.86%)	(1.78%)	(2.24%)
4000	0.7854	0.7599	0.7615	0.6721	0.6611	0.7708	**0.7927**	0.7833
		(3.36%)	(3.140%)	(16.86%)	(18.80%)	(1.90%)	(−0.92%)	(0.27%)
5000	0.7758	0.7382	0.7690	0.6506	0.6456	0.7495	0.7904	**0.7927**
		(5.09%)	(0.88%)	(19.24%)	(20.16%)	(3.51%)	(−1.85%)	(−2.13%)
6000	0.7619	0.7232	0.7741	0.6378	0.6454	0.7286	0.7912	**0.7947**
		(5.35%)	(−1.58%)	(19.46%)	(18.06%)	(4.57%)	(−3.70%)	(−4.13%)
7000	0.7471	0.7148	0.7795	0.6346	0.6374	0.7216	0.7926	**0.7981**
		(4.52%)	(−4.16%)	(17.72%)	(17.21%)	(3.53%)	(−5.74%)	(−6.39%)
Micro-F1								
200	0.7307	0.7357	0.7228	0.6748	0.6562	0.7120	**0.7371**	0.7264
		(−0.69%)	(1.09%)	(8.28%)	(11.35%)	(2.62%)	(−0.87%)	(0.59%)
500	0.7980	**0.8001**	0.7221	0.7543	0.7357	0.7930	0.7471	0.7521
		(−0.26%)	(10.51%)	(5.79%)	(8.47%)	(0.63%)	(6.81%)	(6.10%)
800	0.8102	**0.8138**	0.735	0.7815	0.7643	0.8116	0.7643	0.7586
		(−0.45%)	(10.23%)	(3.67%)	(6.00%)	(−0.18%)	(6.00%)	(6.80%)
1000	0.8166	**0.8181**	0.7428	0.7815	0.7658	0.8209	0.7851	0.7744
		(−0.18%)	(9.94%)	(4.49%)	(6.64%)	(−0.52%)	(4.01%)	(5.45%)
2000	**0.8295**	0.8288	0.7636	0.7872	0.7787	0.8245	0.7915	0.7915
		(0.09%)	(8.63%)	(5.38%)	(6.53%)	(0.61%)	(4.80%)	(4.80%)
3000	**0.8302**	0.8245	0.7686	0.7887	0.7822	0.8245	0.8009	0.798
		(0.69%)	(8.02%)	(5.27%)	(6.14%)	(0.69%)	(3.66%)	(4.04%)
4000	**0.8195**	0.8138	0.7729	0.7872	0.7851	0.8173	0.8023	0.7944
		(0.70%)	(6.03%)	(4.10%)	(4.38%)	(0.27%)	(2.14%)	(3.16%)
5000	**0.8159**	0.8052	0.7815	0.7830	0.7794	0.8073	0.7994	0.8016
		(1.33%)	(4.40%)	(4.20%)	(4.68%)	(1.07%)	(2.06%)	(1.78%)
6000	**0.8102**	0.8009	0.7865	0.7822	0.7822	0.8001	0.7987	0.8037
		(1.16%)	(3.01%)	(3.58%)	(3.58%)	(1.26%)	(1.44%)	(0.81%)
7000	0.8037	0.7980	0.7923	0.7815	0.7815	0.7973	0.8009	**0.8066**
		(0.72%)	(1.44%)	(2.84%)	(2.84%)	(0.81%)	(0.35%)	(−0.36%)

part RF in contributing the discriminating power to the selected term, was proved to be more helpful to the improvement of TC than the other one. Meanwhile, the parameters added to TF-ERF was also determined. After that, TF-ERF was compared with seven existing representative term weighting methods to evaluate the text classification performance, the experimental results proved that TF-ERF have a better TC ability over the listed term weighting methods including TF-RF.

Appendix

```
import sklearn
import nltk
import json
import math
import numpy as np
from math import log
from sklearn import metrics
from sklearn import svm, datasets
from sklearn.datasets import load_files
from sklearn.metrics import classification_report
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.feature_selection import SelectKBest,chi2
from sklearn.naive_bayes import MultinomialNB, BernoulliNB, GaussianNB
from openpyxl import Workbook
from openpyxl import load_workbook#引入库
from sklearn.preprocessing import Normalizer
print(“=====读取 r8-train-stemmed 训练集文本=====”)
with open('r8-train-stemmed.txt') as file_object:
    r8_train = file_object.read()
print(“=====读取 r8-test-stemmed 测试集文本=====”)
with open('r8-test-stemmed.txt') as file_object:
    r8_test = file_object.read()
print(“=====建立数据集类别名列表=====”)
r8_target_names=['acq\t','crude\t','earn\t','grain\t','interest\t','money-fx\t','ship\t','trade\t']#建立数据集类别名列表, 其中\t表示制表符, 即tab键, 用于增加类别名的区分度
#因为用word打开r8数据集可以发现， 每个类别名后面都有一个tab键， 因此可以用类别名加tab键的形式（增加了区分度）来计算训练集合测试集中各类别的文档数。因为类别名区分度不强, 可能在文中也会出现, 例如类别名acq可能作为一个单词在文中出现
print(“=====查看训练集中各类别下的文档数=====”)
train_name_numbers=[]#建立一个空列表存储文档数
for r8_target_name in r8_target_names:#依次从类别名列表中读取类别名
    train_name_numbers.append(r8_train.count(r8_target_name))#根据类别名计算其文档数
    print(r8_target_name, r8_train.count(r8_target_name))
print(“用于训练的文档数: ”,sum(train_name_numbers))
print(“=====查看测试集中各类别下的文档数=====”)
test_name_numbers=[]#建立一个空列表存储文档数
for r8_target_name in r8_target_names:
    test_name_numbers.append(r8_test.count(r8_target_name))
    print(r8_target_name, r8_test.count(r8_target_name))
print(“用于测试的文档数: ”,sum(test_name_numbers))
print(“训练和测试的文档总数: ”,sum(train_name_numbers)+sum(test_name_numbers))
print(“=====将训练集中的类别名全部替换为英文句号“点”, 以方便后面分句=====”)
for r8_target_name in r8_target_names:#用 for 循环依次替换20个不同的类别名
    f1 = open(“r8-train-stemmed_sentences.txt”, “r”)#注意这里用循环来替换， 因此f1的打开与关闭都要放到 for 循环中， 否则只能对最后一个类别名进行替换， 因为那样后面分句后的结果 为378
    content = f1.read()
    f1.close()
    t = content.replace(r8_target_name,“. ”)#将类别名全部替换为英文句号 “点”, 以方便后面分句。注意“点”后面有一个空格， 否则分句后的结果为0， 即 print(len(sentences)) 为 0
    with open(“r8-train-stemmed_sentences.txt”,“w”) as f2:
    f2.write(t)
    f2.close()
print(“=====将训练文本中的回车符去掉=====”)#将文本中的回车符“\n”去掉, 即用“”来替换, 否则打印一个句子（相当于一篇文档）时, 都是分了很多个段落。但是不去掉回车符也不影响后面的词频计算
f1 = open(“r8-train-stemmed_sentences.txt”,“r”)
content = f1.read()
f1.close()
t= content.replace(“\n”,“ ”)#将文中的回车符“\n”去掉, 即用“ ”来替换, 否则打印一个句子（相当于一篇文档）时, 都是分了很多个段落的。
#注意这里是替换为空格， 两个引号之间有一个空格。如果两个引号之间没有空格， 分句之后的结果为11213, 可能是因为回车符恰好在一行的结尾, 导致上下两行链接的中间没有空格。
#如果两个引号之间没有空格， 则可以将上述语句放到分句之后才去掉回车符, 其结果仍为 11293.
with open(“r8-train-stemmed_sentences.txt”,“w”) as f2:
    f2.write(t)
f2.close()
```

Authors' contributions The problems studied in this manuscript and the corresponding contributions are stated in the summary and conclusion, respectively.

Funding This work was supported by The National Natural Science Foundation of China (No.52075350), the Major Science and Technology Projects of Sichuan Province (No.22ZDZX0001), and the Special Projects for Strategic Cooperation between Sichuan University and Yibin City (No.2020CDYB-37).

Availability of data and material The sources of relevant data have been described in this manuscript.

Code availability The codes in this manuscript are programmed by Python. Part of the code have been listed in the appendix at the end of the manuscript.

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent to participate All the authors agreed to participate.

Consent for publication All the authors agreed to publish.

References

Al-Mubaid H, Umair SA (2006) A new text categorization technique using distributional clustering and learning logic. IEEE Trans Knowl Data Eng 18(9):1156–1165

Altınçay H, Erenel Z (2010) Analytical evaluation of term weighting schemes for text categorization. Pattern Recogn Lett 31 (11):1310–1323

Chen K, Zhang Z, Long J et al (2016) Turning from TF-IDF to TF-IGM for term weighting in text classification. Expert Syst Appl 66:245–260

Debole F, Sebastiani F (2003) Supervised term weighting for automated text categorization. In: Proceedings of the 2003 ACM symposium on applied computing, pp 784–788

Deng W, Peng Y, Yang F, et al (2020) Feature optimization and hybrid classification for malicious web page detection. In: Concurrency and computation: practice and experience

Dogan T, Uysal AK (2019) Improved inverse gravity moment term weighting for text classification. Expert Syst Appl 130:45–59

Farid DM, Zhang L, Rahman CM et al (2014) Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. Expert Syst Appl 41(4):1937–1946

Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Mach Learn 29(2–3):131–163

Guru DS, Suhil M, Raju LN et al (2018) An alternative framework for univariate filter based feature selection for text categorization. Pattern Recogn Lett 103:23–31

Haddoud M, Mokhtari A, Lecroq T et al (2016) Combining supervised term-weighting metrics for SVM text classification with extended term representation. Knowl Inf Syst 49(3):909–931

Ilinskas A, Litvinas L (2020) A hybrid of the simplicial partition-based Bayesian global search with the local descent. Soft Comput 24 (10):17601–17608

Labani M, Moradi P, Ahmadizar F et al (2018) A novel multivariate filter method for feature selection in text classification problems. Eng Appl Artif Intell 70:25–37

Lakshmi R, Baskar S (2019) Novel term weighting schemes for document representation based on ranking of terms and Fuzzy logic with semantic relationship of terms. Expert Syst Appl 137:493–503

Lan M, Tan CL, Su J et al (2009) Supervised and traditional term weighting methods for automatic text categorization. IEEE Trans Pattern Anal Mach Intell 31(4):721–735

Li C, Liu S (2018) A comparative study of the class imbalance problem in Twitter spam detection. In: Concurrency and computation: practice and experience

Li W, Miao D, Wang W (2011) Two-level hierarchical combination method for text classification. Expert Syst Appl 38(3):2030–2039

Li Z, Lu W, Sun Z et al (2016) A parallel feature selection method study for text classification. Neural Comput Appl 28:513–524

Li W, Li Y, Chen J et al (2017) Product functional information based automatic patent classification: method and experimental studies. Inf Syst 67:71–82

Liu H, Yu L (2005) Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowl Data Eng 17 (4):491–502

Liu Y, Loh HT, Sun A (2009) Imbalanced text classification: a term weighting approach. Expert Syst Appl 36(1):690–701

Liu L, Li Y, Xiong Y et al (2020) A new function-based patent knowledge retrieval tool for conceptual design of innovative products. Comput Ind 115:103154

Meng JN, Lin HF, Yu YH (2011) A two-stage feature selection method for text categorization. Comput Math Appl 62(7):2793–2800

Ning C, Hongpo Z, Zhangbo L (2021) Data sanitization against label flipping attacks using AdaBoost-based semi-supervised learning technology. Soft Comput 25:14573–14581

Porter MF (2006) An algorithm for suffix stripping. Prog Electr Lib Inf Syst 14(3):130–137

Quan X, Wenyin L, Qiu B (2011) Term weighting schemes for question categorization. IEEE Trans Pattern Anal Mach Intell 33 (5):1009–1021

Ren F, Sohrab MG (2013) Class-indexing-based term weighting for automatic text classification. Inf Sci 236:109–125

Sabbah T, Selamat A, Selamat MH et al (2016) Hybridized term-weighting method for Dark Web classification. Neurocomputing 173:1908–1926

Sabbah T, Selamat A, Selamat MH et al (2017) Modified frequency-based term weighting schemes for text classification. Appl Soft Comput 58:193–206

Şahin DÖ, Kılıç E (2019) Two new feature selection metrics for text classification. Automatika 60(2):162–171

Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manag 24(5):513–523

Salton G, Wong A, Yang CS (1974) A vector space model for automatic indexing. Commun ACM 18(11):613–620

Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47

Shang W, Huang H, Zhu H et al (2007) A novel feature selection algorithm for text categorization. Expert Syst Appl 33(1):1–5

Shang CX, Li M, Feng SZ et al (2013) Feature selection via maximizing global information gain for text classification. Knowl Based Syst 54:298–309

Spärck JK (2004) A statistical interpretation of term specificity and its application in retrieval. J Doc 60(5):493–502

Tang Z, Li W, Li Y et al (2020) Several alternative term weighting methods for text representation and classification. Knowl Based Syst 207(9):106399

Taşcı Ş, Güngör T (2013) Comparison of text feature selection policies and using an adaptive framework. Expert Syst Appl 40(12):4871–4886

Tellez ES, Moctezuma D, Miranda-Jiménez S et al (2018) An automated text categorization framework based on hyperparameter optimization. Knowl Based Syst 149:110–123

Wang D, Zhang H (2013) Inverse-category-frequency based supervised term weighting schemes for text categorization. J Inf Sci Eng 29(2):209–225

Wang S, Pedrycz W, Zhu Q et al (2015) Subspace learning for unsupervised feature selection via matrix factorization. Pattern Recogn 48(1):10–19

Wu H, Gu X, Gu Y (2017) Balancing between over-weighting and under-weighting in supervised term weighting. Inf Process Manag 53(2):547–557

Yang YM, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: Proceedings of the 14th international conference on machine learning, Nashville, USA, pp 412–420

Zhang W, Yoshida T, Tang X (2011) A comparative study of TF*IDF, LSI and multi-words for text classification. Expert Syst Appl 38(3):2758–2765

Zhang W, Li Y, Wang S (2019) Learning document representation via topic-enhanced LSTM model. Knowl Based Syst 174:194–204

Zong W, Wu F, Chu LK et al (2015) A discriminative and semantic feature selection method for text categorization. Int J Prod Econ 165:215–222