Expert Systems With Applications 66 (2016) 245–260

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Turning from TF-IDF to TF-IGM for term weighting in text classification

Kewen Chen, Zuping Zhang*, Jun Long, Hao Zhang

School of Information Science and Engineering, Central South University, Changsha 410083, China

ARTICLE INFO

Article history:
Received 26 April 2016
Revised 9 August 2016
Accepted 5 September 2016
Available online 13 September 2016

Keywords:
Term weighting
Text classification
Inverse gravity moment (IGM)
Class distinguishing power
Classifier

ABSTRACT

Massive textual data management and mining usually rely on automatic text classification technology. Term weighting is a basic problem in text classification and directly affects the classification accuracy. Since the traditional TF-IDF (term frequency & inverse document frequency) is not fully effective for text classification, various alternatives have been proposed by researchers. In this paper we make comparative studies on different term weighting schemes and propose a new term weighting scheme, TF-IGM (term frequency & inverse gravity moment), as well as its variants. TF-IGM incorporates a new statistical model to precisely measure the class distinguishing power of a term. Particularly, it makes full use of the fine-grained term distribution across different classes of text. The effectiveness of TF-IGM is validated by extensive experiments of text classification using SVM (support vector machine) and *k*NN (*k* nearest neighbors) classifiers on three commonly used corpora. The experimental results show that TF-IGM outperforms the famous TF-IDF and the state-of-the-art supervised term weighting schemes. In addition, some new findings different from previous studies are obtained and analyzed in depth in the paper.

1. Introduction

As electronic documents, web pages, messages, etc., all of which mainly contain texts, are increasing dramatically, effective organization, retrieval and mining of the massive textual data are becoming more and more important. Automatic text classification, also known as text categorization, is one of the widely used technologies for the above purposes. In text classification, text documents are usually represented by the so-called vector space model (VSM) and then assigned to predefined classes through supervised machine learning. According to VSM, each document is represented as a numerical feature vector, which consists of the weights of many terms (words or features) extracted from the text corpus. So how to weight terms in an appropriate way is a basic problem in text classification tasks and directly affects the classification accuracy. Although the well-known TF-IDF (term frequency & inverse document frequency) has been proved to be an effective scheme for term weighting in information retrieval (Jones, 1972, 2004) and many text mining tasks (Sebastiani, 2002), it is not the most effective one for text classification because TF-IDF ignores the available class labels of training documents. Therefore, researchers have been seeking more effective term weighting schemes for text classification.

Debole and Sebastiani (2003) have firstly proposed the idea of supervised term weighting (STW), i.e., weighting terms by exploiting the known categorical information in training corpus. They proposed three STW schemes, i.e., TF-CHI, TF-IG and TF-GR by replacing the IDF global factor in TF-IDF with such feature selection functions as χ^2 statistic (CHI), information gain (IG) and gain ratio (GR) respectively. However, they have found from experiments that STW is not consistently superior to TF-IDF. Even so, STW seems to be more reasonable and promising than TF-IDF. Later, STW has attracted a lot of interest from researchers, e.g., Lan, Tan, Su, and Lu (2009), Altınçay and Erenel (2010), Liu, Loh and Sun (2009), Wang and Zhang (2013), Ren and Sohrab (2013), Nguyen, Chang, and Hui (2013), Peng, Liu, and Zuo (2014), and Deng et al. (2014), etc. Term weighting becomes one of the hot research topics in text classification and various new STW schemes have been proposed from time to time.

However, most of the STW schemes consider only term distribution in two classes of the positive and negative text at the level of coarse granularity while weighting a term (Lan et al., 2009). These schemes may not be optimal for multiclass text classification with more than two classes. Accordingly, the experimental results in the previous studies are mostly obtained from the binary text classification experiments. Even though the experiments were conducted on multiclass datasets, multiple independent bi-

* Corresponding author. Fax: +86 0731 82539926.
E-mail addresses: kewencsu@csu.edu.cn (K. Chen), zpzhang@csu.edu.cn (Z. Zhang), jlong@csu.edu.cn (J. Long), hao@csu.edu.cn (H. Zhang).

http://dx.doi.org/10.1016/j.eswa.2016.09.009
0957-4174/© 2016 Elsevier Ltd. All rights reserved.

nary classifications between the positive and negative classes were actually done in some "one-against-rest" or "one-versus-all" way (Nguyen et al., 2013). So the effectiveness of some previous STW schemes for multiclass text classification remains unknown. In addition, although some STW schemes are based on inverse class frequency (ICF) over multiple classes (Lertnattee & Theeramunkong, 2004; Wang & Zhang, 2013), they also have some deficiencies, e.g., class frequency is too coarser than document frequency.

Moreover, most of the STW schemes, e.g., Debole and Sebastiani (2003), Deng et al. (2004), Liu et al. (2009), and Peng et al. (2014), etc., incorporate the information of term intra-class distribution, i.e., a term's distribution within a class of text. And the intra-class distribution and inter-class distribution across different classes are even taken as equally important in STW, e.g., as in Liu et al. (2009). But in our view, the importance (or weight) of a term in text classification depends mainly on its class distinguishing power. While determining the class distinguishing power of a term, the inter-class distribution is intuitively more important than the intra-class distribution. So the effectiveness of those weighting schemes based on term intra-class distribution is questionable. This will be clarified later in this paper. Although some STW schemes were proved to be effective in some specific text classification tasks, in fact they obtained only modest performance improvement.

All the aforementioned STW schemes are based on statistics. In recent years some new term weighting schemes based on semantics have also been put forward, e.g., in Wei, Feng, He, and Fu (2011) and Luo, Chen, and Xiong (2011). However, semantic term weighting schemes are more complex than statistical counterparts but fail to obtain significant performance improvement.

The main purpose of this study is to address such a question: How to make full use of the information of the fine-grained statistical distribution of a term over multiple classes of text so as to make term weighting more reasonable and further improve the performance of text classification?

The main contributions of this paper are following: Firstly, a new statistical model is built to characterize the inter-class distribution and measure the class distinguishing power of a term in a text corpus. Secondly, a new term weighting scheme called TF-IGM (term frequency & inverse gravity moment) is proposed as well as its variants. Thirdly, comparative studies on different term weighting schemes are made both by theoretical analyses and by extensive experiments of text classifications on three commonly used benchmark datasets. Finally, some new findings different from or contrary to previous studies are obtained and they are analyzed thoroughly as well as some related issues are discussed in depth. The experimental results show that the proposed TF-IGM outperforms the famous TF-IDF and the state-of-the-art STW schemes. Particularly, one of its improved versions performs best in almost all the cases.

The remainder of the paper is organized as follows. Section 2 analyzes current term weighting schemes. Section 3 elaborates the proposed IGM model and TF-IGM weighting scheme. Section 4 introduces the experimental work and analyzes the experimental results with some related issues discussed. Section 5 concludes the paper.

2. Analyses of current term weighting schemes

Various term weighting schemes for text classification and some relevant concepts are analyzed in this section.

2.1. Traditional term weighting schemes

Traditional term weighting schemes are Binary (or Boolean), TF and TF-IDF weighting (Lan et al., 2009; Sebastiani, 2002), which are originated from information retrieval (Jones, 1972, 2004; Salton & Buckley, 1988). As the weight of a term, the term frequency (TF) in a document is obviously more precise and reasonable than the binary value, 1 or 0, denoting term presence or absence in the document because the topic terms or key words often appear in the document frequently and they should be assigned greater weights than the rare words. But term weighting by TF may assign large weights to the common words with weak text discriminating power. To offset this shortcoming, a global factor, namely inverse document frequency (IDF), is introduced in the TF-IDF scheme. The reason is that, for a term with high frequency in a document, the lower its document frequency in the corpus is or the less other documents containing it are, the higher its representativeness of the specific document is. So TF-IDF is more reasonable than the first two schemes and has been widely applied. For term t_k, its TF-IDF weight, $w(t_k)$, in a document is usually represented as follows in text classification (Sebastiani, 2002).

$$w(t_k) = tf_k \cdot \log\left(\frac{N}{df_k}\right) \tag{1}$$

where tf_k is the term frequency that t_k occurs in the document, and df_k is the document frequency of t_k, i.e., the number of documents containing t_k, while N is the total number of documents in the corpus. In addition, the IDF factor has many variants such as $\log(N/df_k+1)$, $\log(N/df_k)+1$, etc.

It is well known that text classification is a supervised machine learning task, which needs a set of text with classes labeled to train the learning model or classifier. But TF-IDF does not adopt the known class information of training text while weighting a term, so the computed weight cannot fully reflect the term's importance in text classification. For example, suppose that there were two terms with the same document frequencies, one appearing in multiple classes of text while another appearing in only one class of text. Apparently the second term has stronger class distinguishing power than the first one, but their global weighting factors, IDF, are the same. That is not reasonable. On the other hand, TF-IDF overemphasizes a term's importance in a document but ignores its contribution to text classification. A topic term specific to a class of text is very likely assigned a lower TF-IDF weight than a rare word.

Table 1
Distribution of a term in the training corpus.

Class	c_j	$\bar{c}_j$
t_k	A	B
no t_k	C	D

2.2. Feature-selection-metrics based term weighting schemes

In consideration of the deficiencies in TF-IDF, researchers have proposed supervised term weighting (STW) (Debole & Sebastiani, 2003), i.e., weighting a term using the known information about the classes of text. The distribution of a term in different classes of text can be described with a contingency table. As a typical example, Table 1 illustrates the distribution of term t_k with respect to class c_j in the training corpus.

In Table 1, A and B are the document frequencies of term t_k, i.e., the number of documents containing t_k, in the positive class c_j and negative class $\bar{c}_j$ respectively. C and D are the numbers of documents without t_k in c_j and $\bar{c}_j$ respectively. The total number of documents in the training corpus, $N = A + B + C + D$.

Most of current STW schemes adopt feature selection metrics (Yang & Pedersen, 1997) as the global weighting factors, for example, the chi-square statistic in TF-CHI (Debole & Sebastiani, 2003; Deng et al., 2004), information gain in TF-IG (Debole & Sebastiani, 2003; Lan et al., 2009), gain ratio (Quinlan, 1986) in TF-GR (Debole & Sebastiani, 2003), correlation coefficient (Hwee, Wei, & Kok, 1997) in TF-CC (Liu et al., 2009; Ren & Sohrab, 2013), mutual information in TF-MI (Altinçay & Erenel, 2010; Ren & Sohrab, 2013), odds ratio in TF-OR (Altinçay & Erenel, 2010; Lan et al., 2009; Liu et al., 2009; Ren & Sohrab, 2013) and so on. The above metrics can reflect a term's importance in text classification and are effective in feature selection, so they are naturally taken as the global weighting factor to be used for assigning appropriate weights to terms. For example, the popular chi-square or χ^2 statistic is incorporated in the TF-CHI scheme (Debole & Sebastiani, 2003). The χ^2 value of term t_k with respect to class c_j can be computed with (2) (Schütze, Hull, & Pedersen, 1995).

$$\chi^2(t_k, c_j) = \frac{N \cdot (A \cdot D - B \cdot C)^2}{(A+B) \cdot (C+D) \cdot (A+C) \cdot (B+D)} \tag{2}$$

Then, by combining the term frequency and χ^2, we can calculate the TF-CHI (Debole & Sebastiani, 2003) weight of term t_k as in

$$w(t_k, c_j) = tf_k \cdot \chi^2(t_k, c_j)\,. \tag{3}$$

According to (3), the TF-CHI weight is class-specific, that is, the term has different weights in individual classes of text. Even so, we can find from (2) that the χ^2 factor in TF-CHI is not bias to the positive class, but also closely related to the negative class, and moreover the two classes are treated equally. But in fact, the size of the positive class is often far smaller than that of the negative counterpart. Generally, the ratio between them is 1:(m - 1), where m is the number of classes, for example, in a balanced multiclass dataset. On the other hand, the χ^2 factor takes into account the term's distributions both within each class and across different classes as well as all the distributions are taken as equally important. We question the rationality of the above two aspects of TF-CHI, which will be further discussed later. As a matter of fact, TF-CHI performs worse than TF-IDF in some text classification experiments (Lan et al., 2009).

2.3. Modified supervised term weighting schemes

Researchers have further studied STW extensively and proposed some modified or new STW schemes, e.g., Lan et al. (2009), Liu et al. (2009), Nguyen et al. (2013), Peng et al. (2014), and Ren and Sohrab (2013), etc. The typical example of those schemes is TF-RF proposed by Lan et al. (2009). According to TF-RF, a term is weighted by its so-called relevance frequency (RF), which is related to the ratio, A/B, between the term's document frequencies in the positive and negative classes. The TF-RF weight of term t_k with respect to class c_j is represented as (4).

$$w(t_k, c_j) = tf_k \cdot \log_2\left(2 + \frac{A}{\max(B, 1)}\right) \tag{4}$$

For a term specific to class c_j, usually $A >> B$, so it is assigned a great weight. The basic idea behind TF-RF is that the more concentrated a high-frequency term is in the positive class than in the negative class, the more contributions it makes in selecting the positive text from the negative text (Lan et al., 2009). Some experimental results (Altinçay & Erenel, 2010; Lan et al., 2009; Wang & Zhang, 2013) show that TF-RF performs better than most supervised and traditional term weighting schemes.

However, TF-RF also weights terms by grouping multiple classes into a single negative class just like TF-CHI. So TF-RF may not be the optimal term weighting scheme for multiclass text classification because it ignores the term's precise distribution across multiple classes of text.

In another modified STW scheme (Liu et al., 2009) called TF-Prob, a probability-based weighting factor (Prob) is defined, which combines A/B and the ratio, A/C, between the numbers of positive documents containing and not containing the term. The TF-Prob weight of term t_k with respect to c_j is represented as (5).

$$w(t_k, c_j) = tf_k \cdot \log\left(1 + \frac{A}{B}\frac{A}{C}\right)\,. \tag{5}$$

TF-RF incorporates only the inter-class distribution (across different classes) of a term, represented by A and B. But TF-Prob incorporates the intra-class distribution within the positive class, represented by A and C, besides the inter-class distribution of a term. The reason for the introduction of the term's intra-class distribution (within a class) into TF-Prob is that the higher frequency or probability its occurrence within a class is of, especially for $A >> C$, the more a term is representative of the class, so it should be assigned a greater weight. Because of almost the same reason, some other STW schemes such as TF-CHI (Debole & Sebastiani, 2003), TFIPNDF (Peng et al., 2014), etc., also take the term's intra-class distribution into account. However, we doubt that the intra-class distribution factor may have no apparently positive effect on term weighting. This will be discussed in depth in Section 4.6.

2.4. ICF-based term weighting schemes

Unlike the above STW schemes which take into account only the term's coarse distribution on the positive and negative classes, the ICF-based weighting schemes, e.g., TF-ICF and TF-IDF-ICF (Lertnattee & Theeramunkong, 2004), take the term distribution across multiple classes into account. As we know, the inverse document frequency (IDF) of a term reflects only the term's specificity or uniqueness with respect to a single document. Similarly, a term's specificity to a class of documents should be reflected by its inverse class frequency (ICF) which is defined as the inverse ratio of the number of classes in which the term occurs to the total number of classes. ICF means that the fewer classes a term occurs in, the greater its specificity to a class or its class distinguishing power is. Indeed, TF-ICF performs better than TF-IDF in some text classification tasks (Lertnattee & Theeramunkong, 2004; Wang & Zhang, 2013). However, ICF adopts only the term distribution at the coarse class level instead of the document level. In order to bear a term's distinguishing power among documents, ICF and IDF are combined in the TF-IDF-ICF scheme (How & Narayanan, 2004; Lertnattee & Theeramunkong, 2004; Ren & Sohrab, 2013) which has been proved effective in text classification. Furthermore, in consideration of the fact that a term representative of a class may occasionally appear in other classes, so a new factor called inverse class-space-density-frequency (ICSDF) as a variant of ICF has recently been introduced in the TF-IDF-ICSDF scheme (Ren & Sohrab, 2013). For ICSDF, the average number of classes where a term occurs, which is so called class-space-density-frequency, is calculated as the sum of the probabilities of the term's occurrence in individual classes. Hence, the TF-IDF-ICSDF weight can be calculated by (6).

$$w(t_k) = tf_k \cdot \left(1 + \log\frac{N}{df_k}\right) \cdot \left(1 + \log\frac{m}{\sum\limits_{j=1}^{m}\frac{df_{kj}}{N_j}}\right), \tag{6}$$

where df_{kj} and N_j are respectively the document frequency of term t_k and the total number of documents in class c_j (j=1, 2,..., m). According to ICSDF, a term representative of a class while occurring occasionally in other classes is assigned a greater weight than those terms occurring frequently in as many classes as it. However, a problem is ignored in the TF-IDF-ICSDF scheme, that is, many terms without any class representativeness or specificity, especially some relatively rare words, appear in almost all classes of text with low probabilities. For these terms, their class-space-density-frequencies are low while the corresponding ICSDF factors are large. So the resulting large weights of these terms are not consistent with their weak class distinguishing powers.

2.5. Semantic term weighting schemes

The aforementioned STW schemes are based on statistics. On the other hand, new term weighting schemes based on semantics, e.g., in Luo et al. (2011) and Wei et al. (2011), have been put forward in recent years. The basic idea of semantic weighting schemes is that the weight of a term is determined by its semantic similarity with a specific class which can be represented with some key words. The semantic similarity or distance between words (terms) can be computed with the help of a thesaurus such as WordNet.[1] Due to the need to access WordNet, Wikipedia[2] or other external knowledge bases, as well as the consideration of some additional issues such as word sense disambiguation and class keyword selection, the complexity of the semantic scheme is raised while compared with the statistical counterpart. So the studies on term weighting based on statistics are still the mainstream. And moreover, some tokens (strings) with strong class distinguishing power, but without explicit semantics or exact meaning, can be found from the text corpus by statistical methods. These tokens may be used as the feature terms and should be assigned greater weights. In fact, the semantic weighting schemes (Luo et al., 2011; Wei et al., 2011) have not shown significant superiority over the statistical ones. Therefore, this study described next focuses on term weighting based on statistics, too.

3. New term weighting scheme: TF-IGM

In this section, we propose a so-called IGM (inverse gravity moment) model to measure the class distinguishing power of a term and then put forward a new term weighting scheme, TF-IGM, by combining term frequency (TF) with the IGM measure.

3.1. Main observations and motivations

As described in Section 2, most supervised term weighting (STW) schemes incorporate the statistics of the inter-class distribution (across different classes) and intra-class distribution (within a class) of a term. However, we doubt that the intra-class distribution factor has any apparently positive effect on term weighting. In our view, the weight of a term should be determined mainly by its class distinguishing power, which is embodied primarily by its uneven distribution across different classes.

In general, the more uniformly a term distributes in the text dataset, the weaker its class distinguishing power is. The examples are some common words, which have no peculiarity or representativeness with respect to any class. On the contrary, a term with class representativeness or uniqueness often concentrates in only one class or few classes of text, and obviously it has strong class distinguishing power. Of course, some class-specific terms may occasionally appear in other classes of text. That is to say, though these terms may appear in multiple classes of text, but they occur very frequently in only a few classes or even a single class of text. Because of the high non-uniformity of their distribution on the whole dataset, these class-specific terms can be used to distinguish text among different classes and should be assigned greater weights. In general, a term with more concentrated inter-class distribution tends to have stronger class distinguishing power than others. Therefore, a term can be weighted according to its inter-class distribution concentration or non-uniformity. As we know, the non-uniformity of sample distribution is measured commonly by the entropy in information theory or variance in mathematics. However, in this study we proposed a new statistical model called "inverse gravity moment" (IGM) to measure the non-uniformity or concentration level of a term's inter-class distribution, which reflects the term's class distinguishing power. On this basis, an appropriate weight is assigned to the term.

3.2. Inverse gravity moment

To measure the inter-class distribution concentration of term t_k, first of all, one needs to sort all the frequencies of t_k's occurring in the individual classes of text in descending order. The resulting sorted list is $f_{k1} \geq f_{k2} \geq \ldots \geq f_{km}$, where f_{kr} ($r = 1, 2, \ldots, m$) is the frequency of t_k's occurring in the r-th class of text after being sorted, and m is the number of classes. If f_{kr} is regarded as a class-specific "gravity", then for the sorted list, the head on the left is heavier than the tail on the right, and the center of gravity of the overall inter-class distribution is bias to the left. Obviously, the fewer classes the occurrences of a term concentrates in, the closer the center of gravity is to the left head. Especially, when the term occurs in only one class of text, the center of gravity is at the starting position ($r = 1$). Only when the inter-class distribution is uniform, that is, $f_{k1} = f_{k2} = \ldots = f_{km}$, the center of gravity is located at the center position ($m/2$). Therefore, the inter-class distribution concentration of a term can be reflected by the position of the gravity center. However, we do not directly adopt the position of the gravity center, but instead a new metric associated with it, to measure the inter-class distribution concentration. For the class-specific "gravity" f_{kr}, if its rank r is regarded as the distance to the origin 0, the product of $f_{kr} \cdot r$ is so called "gravity moment" (GM) in physics. For a total frequency of a term occurring in the text corpus, the more concentrated the term inter-class distribution is, the shorter the distance of the gravity center is to the origin and the less the sum of all the class-specific gravity moments is, and meanwhile the higher the maximum class-specific frequency (f_{k1}) is, too. The above facts show that the concentration level of the term inter-class distribution is proportional to the reciprocal value of the total gravity moment. So, a new statistical model called "inverse gravity moment" (IGM) is proposed to measure the inter-class distribution concentration of a term, which is defined as follows.

$$igm(t_k) = \frac{f_{k1}}{\sum_{r=1}^{m} f_{kr} \cdot r} . \tag{7}$$

where $igm(t_k)$ denotes the inverse gravity moment of the inter-class distribution of term $t_{k,}$ and f_{kr} ($r = 1, 2, \ldots, m$) are the frequencies of t_k's occurring in different classes, which are sorted in descending order with r being the rank. Usually, the frequency, $f_{kr,}$ refers to the class-specific document frequency (DF), i.e., the number of documents containing the term t_k in the r-th class, denoted as df_{kr}.

The inverse gravity moment of term inter-class distribution ranges from $2/((1+m) \cdot m)$ to 1.0. Since the first element, f_{k1}, is the maximum in the list of $\{f_{kr} \mid r = 1, 2, \ldots, m\}$ in descending order, formula (7) can be rewritten in the form of the following

$$igm(t_k) = \frac{1}{\sum_{r=1}^{m} \frac{f_{kr}}{\max\limits_{1 \leq i \leq m}(f_{ki})} \cdot r} \tag{8}$$

The formula (8) shows that the IGM of a term is the inverse of the total gravity moment calculated from the normalized frequen-

[1] http://wordnet.princeton.edu/.

[2] http://www.wikipedia.org/.

cies of the term's occurrence in all the individual classes. The IGM value, $igm(t_k)$, is the minimum, $2/((1+m)\cdot m)$, for a uniform distribution where $f_{k1}=f_{k2}=\ldots=f_{km}$. But it is 1.0 for $f_{k1}>0$ and $f_{k2}=\ldots=f_{km}=0$ when term t_k occurs in only one class of text.

As a measure of the distribution concentration level, the IGM value can be transformed to fall in the interval [0, 1.0], as follows.

$$nigm(t_k)=\frac{(1+m)\cdot m\cdot igm(t_k)-2}{(1+m)\cdot m-2} \tag{9}$$

where $nigm(t_k)$ denotes the normalized IGM value, m is the number of classes. However, the above transformation is not necessary because the minimum value of $2/((1+m)\cdot m)$ is close to zero. So, the standard IGM model defined in (7) is adopted in our proposed term weighting schemes.

3.3. Term weighting by TF-IGM

The weight of a term in a document should be determined by its importance in the document and its contribution to text classification, which correspond respectively to the local and global weighting factors in term weighting. A term's contribution to text classification depends on its class distinguishing power which is reflected by its inter-class distribution concentration. The higher the concentration level is, the greater weight should be assigned to the term. The former can be measured by the IGM model. Hence, instead of the traditional IDF factor, a new global factor in term weighting is defined based on the IGM metric of the term, as shown in (10).

$$w_g(t_k)=1+\lambda\cdot igm(t_k) \tag{10}$$

where $w_g(t_k)$ denotes the IGM-based global weighting factor of term t_k, and λ is an adjustable coefficient. The purpose of introducing the coefficient λ is to keep the relative balance between the global and local factors in the weight of a term. Generally, the coefficient is set empirically to be $5.0\sim 9.0$. The default value of λ is set to be 7.0, which is usually appropriate for different datasets according to our experiments. The optimal value of λ can also be found through incrementally testing for a specific dataset or text classification task.

Just as in TF-IDF, the local weighting factor is generally the term frequency (TF), i.e., the number of a term's occurrences in a document, denoted as tf_{kd} for term t_k occurring in document d. But a term occurring 20 times in a document is generally less than 20 times as important as a term occurring only once in that document. So reducing the effect of high TF may result in more reasonable term weighting. It has been found from some researches (Erenel & Altinçay, 2012; Xuan & Quang, 2014) and our preliminary experiments that if the local TF factor is reduced properly, the accuracy of text classification is improved for some text corpora. The local TF factor is usually reduced by a logarithm operation, for example, replacing tf_{kd} with $\log(tf_{kd}+1)$ (Dumais, 1991). However, we adopt an alternative method, i.e., calculating the square root of TF, i.e., sqrt(tf_{kd}) or $\sqrt{tf_{kd}}$, which results in the text classification performance equivalent to or sometimes even better than the logarithm method (Chen & Zong, 2003). So there are two options for the TF-based local weighting factor, $w_l(t_k, d)$, in our proposed scheme, i.e., $w_l(t_k, d)=\{tf_{kd}, \text{sqrt}(tf_{kd}) \mid tf_{kd}>0\}$.

The TF-IGM weight of term t_k in document d is the product of the TF-based local weighting factor and the IGM-based global weighting factor, i.e., $w(t_k, d)=w_l(t_k, d)\cdot w_g(t_k)$, which is expressed as (11-a) or (11-b).

$$w(t_k,d)=tf_{kd}\cdot\left(1+\lambda\cdot\frac{f_{k1}}{\sum_{r=1}^{m}f_{kr}\cdot r}\right) \tag{11-a}$$

$$w(t_k,d)=\sqrt{tf_{kd}}\cdot\left(1+\lambda\cdot\frac{f_{k1}}{\sum_{r=1}^{m}f_{kr}\cdot r}\right) \tag{11-b}$$

where the TF of t_k in d, $tf_{kd}>0$. Otherwise, $w(t_k, d)=0$ if $tf_{kd}=0$. The frequency, f_{kr} ($r=1, 2, \ldots, m$) usually refers to the class-specific DF of the term. But in fact, we have tried three other alternatives for it, which is discussed later. The two term weighting schemes expressed by (11-a) and (11-b) are respectively called TF-IGM and RTF-IGM, where RTF denotes the root of term frequency. Obviously, RTF-IGM is an improved version of TF-IGM.

Of course, TF-IGM is a supervised term weighting (STW) scheme because it depends on the known class information of training text, too. However, unlike most STW schemes such as TF-CHI, TF-RF, etc. which result in class-specific weights of a term, TF-IGM results in a term's weight independent of classes, just like TF-IDF. Because of its independence of classes, the unique IGM-based global weighting factor for a term can be obtained by one-time calculation from the training dataset. But, for TF-CHI, TF-RF and others alike, several weighting factors specific to different classes should be calculated for a term in multiclass text classification.

3.4. Empirical comparisons against other schemes

According to the previous descriptions, the weight of a term is generally determined by two parts, i.e., the local weighting factor within a document and the global weighting factor in the corpus. Since TF is usually used as the local factor, the main difference among different term weighting schemes is how to calculate the global factor, which reflects the term's importance in text classification. So, in order to investigate the effectiveness and superiority of TF-IGM compared with several typical term weighting schemes described in Section 2, it is enough to analyze and compare the global weighting factors in different schemes. Here, let's illustrate this by the empirical observations from two simple examples.

In the first example, suppose you have two terms t_1 and t_2 with their document frequencies in five classes being {4, 2, 2, 2, 2} and {4, 8, 0, 0, 0} respectively, and the number of documents in each class is 10. By the intuition, term t_2 has stronger class distinguishing power than term t_1 and should be assigned a greater weight. But, according to the descriptions in Section 2, t_1 and t_2 share the same IDF and ICSDF values, as well as the same values of the global factors specific to the first class in TF-CHI, TF-RF and TF-Prob respectively. As a result, t_1 and t_2 are assigned equal weights and that is not reasonable. On the other hand, by formula (7), the IGM values of t_1 and t_2 are 0.125 and 0.5 respectively, and then according to (10) and setting the coefficient $\lambda=7$, the corresponding IGM-based global weighting factors are 1.875 and 4.5 respectively. Apparently, the TF-IGM weight of t_1 is less than the weight of t_2, denoted as $w(t_1)<w(t_2)$, for the same TF. This is consistent with the intuition. Of course, if weighting the terms by their inverse class frequencies (ICF), we also have $w(t_1)<w(t_2)$, similar to the case of IGM.

However, ICF is not so reasonable in another example where the class-specific document frequencies of t_1 and t_2 are {8, 7, 6, 6, 0, 0} and {9, 2, 2, 2, 0, 0} respectively. If weights are assigned by ICF, then $w(t_2)=w(t_1)$ for the two terms share the same ICF. This result contradicts the intuition of $w(t_2)>w(t_1)$. However, the IGM values calculated for t_1 and t_2 are 0.125 and 0.333 respectively, and accordingly, the IGM-based global weighting factors with $\lambda = 7$ are 1.875 and 3.333 respectively. So if weights are assigned based on IGM, then $w(t_2) > w(t_1)$. The result remains the consistence with the intuition.

It can be easily found from the above empirical observations and comparisons that the weight assigned by the TF-IGM scheme is more reasonable than the results obtained from other term weighting schemes described in Section 2, such as the traditional TF-IDF and supervised TF-RF, etc..

3.5. Variants with different inter-class distributions

As described above, the inter-class distribution of a term for IGM is usually represented by the class-specific DF of the term in each class, denoted as df_{kr} ($r = 1, 2, \ldots, m$) for term t_k. We have tried three additional types of the term inter-class distribution, which are represented respectively by the DF ratio, total TF and average TF of the term in each class. The DF ratio is df_{kr}/N_r for term t_k in class c_r, where N_r is the size (document number) of class c_r. The total TF of t_k in the r-th class, $tf_k(c_r) = \mathrm{sum}(tf_{kd})$ for $d \in c_r$, that is, the sum of the term frequencies of t_k in all the documents of c_r. The average TF of t_k in the r-th class is $tf_k(c_r)/N_r$. On an imbalanced dataset which may have quite different numbers of documents in individual classes, replacing df_{kr} with df_{kr}/N_r, or replacing $tf_k(c_r)$ with $tf_k(c_r)/N_r$, may result in a more reasonable representation of the term inter-class distribution for the IGM model.

So, there are four options for the frequency of a term's occurring in a class in the IGM model. That is, the value of f_{kr} ($r = 1, 2, \ldots, m$) in (11-a) and (11-b) is determined by one of $\{df_{kr}, df_{kr}/N_r, tf_k(c_r), tf_k(c_r)/N_r\}$ while weighting terms based on the IGM model. However, the first two DF-related options are more preferable than the last two TF-related options in text classification tasks. Although TF is of finer granularity than DF, our preliminary experiments show that the performance of text classification is not obviously improved when the term inter-class distribution is represented by the class-specific $tf_k(c_r)$ or $tf_k(c_r)$ $/N_r$ instead of the class-specific df_{kr} or df_{kr} $/N_r$ ($r = 1, 2, \ldots, m$). Additionally, the calculation of $tf_k(c_r)$ increases the computational overhead with respect to df_{kr}. Of course, the best choice of the above four options can also be determined by experiments over a specific corpus. Therefore, TF-IGM is adaptive to different text corpora or classification tasks by providing several options or adjustable parameters so as to obtain the optimal performance.

3.6. Computational complexity of TF-IGM

Compared with TF-IDF, TF-RF and other term weighting schemes, TF-IGM needs the sorting operation for the frequencies of a term's occurring in different classes, which time complexity is from $O(m)$ to $O(m^2)$ (Tang, Li, & Huang, 1995) with m being the number of classes. The time complexity depends on the initial state of the term inter-class distribution and the sorting algorithm used. For example, it is $O(m \cdot \log_2(m))$ for the quick-sorting algorithm (Tang et al., 1995). Due to the small number of predefined classes, usually dozens or less, the sorting operation does not increase the computational complexity significantly. Moreover, because of its independence of classes, the unique IGM-based global weighting factor for each term can be obtained by one-time calculation from the training dataset. But, in most STW schemes like TF-CHI, TF-RF and so on, m class-specific global weighting factors should be calculated for each term. However, from the sorted class-specific frequencies, $\{f_{kr} \mid r = 1, 2, \ldots, m\}$, the time complexity for calculating the IGM-based global weighting factor is $O(m)$, just the same as in other term weighting schemes. So, TF-IGM is at the cost of the slightly increased computational complexity while compared with TF-IDF, TF-RF and others.

4. Experiments

In order to validate the effectiveness of the proposed TF-IGM and RTF-IGM schemes for term weighting, we have done extensive experiments of text classification using eight term weighting schemes. The performances of the eight schemes are compared and analyzed later in this section.

4.1. Datasets and preprocessing

Three commonly used benchmark datasets, i.e., 20 Newsgroups, Reuters-21578 and TanCorp, are used in our experiments of multiclass text classification because they have different characteristics. The first two are English corpora while the third is a Chinese corpus. 20 Newsgroups is a class-balanced dataset with almost equal number of documents in each class. However, Reuters-21578 and TanCorp are imbalanced or skewed datasets where individual classes may have quite different number of documents. These three datasets are widely utilized by researchers to test text classification, e.g., in Debole and Sebastiani (2003), Lan et al. (2009), Nguyen et al. (2013), Ren and Sohrab (2013), and Tan, Cheng, Ghanem, Wang, and Xu (2005).

20 Newsgroups: This English corpus contains 19,997 documents of newsgroup messages, which are organized into 20 newsgroups or classes. The 20news-bydate[3] version with 1151 duplicate documents and some headers of the messages already removed was used in the experiments. It includes 18,846 documents which have already been sorted by date and divided into a training set with 11,314 documents and a test set with 7532 documents. Most classes in the training set are of approximately equal size, each with nearly 600 documents. During the preprocessing phase, all the stop words in the stoplist defined in the SMART project (Buckley, 1985), rare words that occur less than 2 times in the dataset, numbers, punctuations and other non-alphabetic characters are removed. Meanwhile, the letters are converted to lowercase and words are stemmed using Porter's stemmer (Porter, 1980, 2006), for example, removing the verb suffixes such as *-s*, *-ed* and *-ing*. Finally, 35,642 terms are extracted from the training set to construct the original feature space of the corpus.

Reuters-21578: This English corpus contains 90 classes of news documents. The top 10 largest classes were selected for the multiclass text classification experiments. The reduced corpus contains 9980 documents, which have been divided into a training set with 7193 documents and a test set with 2787 documents according to ModApte Split.[4] We removed the duplicate documents from this so-called Reuters-21578 ModApte Top 10 corpus. All the duplicates labeled as different classes are removed. But one document is kept while the rest are removed in each group of duplicates being of the same class. After removing the duplicates in the training set and test set respectively, the *corn* and *wheat* classes become empty or have only one document left. So the two classes are finally deleted. The remaining eight classes contain totally 8120 documents including 5798 training documents and 2322 test documents. The eight classes in the training set are severely imbalanced with 2843 documents in the largest *earn* class and only 79 documents in the smallest *grain* class. By preprocessing in the same way as described above for 20 Newsgroups, the resulting corpus has a vocabulary of 8436 words (terms).

TanCorp: The corpus contains 14,150 Chinese documents (Tan et al., 2005). The TanCorp-12 version in the preprocessed format was used for experiments. It consists of 12 classes of different sizes, with 2943 documents in the largest class but only 150 docu-

[3] http://qwone.com/~jason/20Newsgroups.

[4] http://disi.unitn.it/moschitti/corpora.htm.

ments in the smallest one. There are no duplicate documents labeled as different classes and few duplicates being of the same classes, which have slight and ignorable effects on the final performance evaluation. So no duplicates were removed while preprocessing this corpus. All the text documents have already been segmented into words (terms) with the Chinese tokenizer, ICTCLAS.[5] Meanwhile the numbers and punctuations are removed, too. The vocabulary built from the corpus contains 72,601 terms, among which the stop words and other unimportant terms are to be removed later in the process of feature selection. The corpus was divided randomly on per-class basis into a training set with 9345 documents and a test set with 4805 documents.

4.2. Experimental steps and methods

After preprocessing the datasets as described above, such operations as feature selection, term weighting, classifier training and test, performance evaluation are done sequentially in the experiments.

As we know, the goal of feature selection (FS) is to reduce the high dimensionality of feature (term) space by selecting the more important terms for text classification. Since FS is essential for text classification due to the processing time and classification accuracy considerations, it is necessary to evaluate the performances of term weighting schemes in various feature spaces of different dimensionalities. So, various subsets of terms (features) of the above three datasets are tried in our experiments. The FS metric (method) used in our experiments is the χ^2 or chi-square statistics (CHI) of a term with respect to text classes, as defined by (2). CHI has been proved to be an effective FS method and widely used in text classification (Lan et al., 2009; Yang & Pedersen, 1997). Ranked by the maximum chi-square statistics (CHI_{max}) of a term over all the classes, the top {500, 1000, 2000, 4000, 6000, 9000, 12000, 16000} terms are selected for the 20 Newsgroups corpus, {100, 300, 500, 700, 1000, 1500, 2000, 3000, 5000, 8436} terms for the Reuters-21578 corpus, and {200, 500, 1000, 2000, 4000, 6000, 8000, 10000} terms for the TanCorp corpus. We also conduct experiments on each corpus without FS by keeping all the original terms (features) after preprocessing the corpus.

Eight term weighting schemes such as TF-IGM, RTF-IGM, TF, TF-IDF, TF-IDF-ICSDF, TF-CHI, TF-Prob and TF-RF are tried so as to compare their performances. Because the global weighting factors in the last three STW schemes are specific to the class of the document but the class of a test document to be classified is taken as unknown, the maximum class-specific global factors, denoted as CHI_{max}, $Prob_{max}$ and RF_{max} respectively, are used in term weighting for the test document. While weighting terms with the proposed TF-IGM or RTF-IGM schemes, the inter-class distribution of each term is represented by its class-specific DFs and the coefficient λ is set as $\lambda = 6.0$ for Reuters-21578 and $\lambda = 7.0$ for 20 Newsgroups and TanCorp. The calculated weights of all the terms in a document are finally normalized by *cosine normalization* (Debole & Sebastiani, 2003; Sebastiani, 2002).

To classify the text, two popular classifiers, i.e., support vector machine (SVM) and k nearest neighbor (kNN) classifiers, are chosen because of their proven good performances in text classification. SVM is a functional learning algorithm which builds a classification model by supervised learning or training. The popular open-source software package, LibSVM[6] is used in our experiments and set up with a linear kernel and default parameters. LibSVM has already been extended to support multiclass classification although the original SVM supports only binary classification (Chang & Lin, 2011). The kNN classifier inherently supports multiclass classification. It is a lazy learning algorithm with no training phase. It directly classifies an unlabeled document by comparing its similarity with the labeled documents in the training set. In our experiments, the similarity between any two documents is measured by the well-known *cosine similarity* (Sebastiani, 2002). The choice of the number of "nearest neighbors", k, is optimized by performance evaluation on a specific dataset. We tried $k = \{5, 10, 15, 20, 30\}$ and found that, for the 20 Newsgroups corpus, the performance for $k = 20$ is optimal in most cases with different feature numbers and term weighting schemes. For the Reuters-21578 and TanCorp corpora, it is appropriate to choose $k = 15$ and $k = 5$ respectively.

The *precision* and *recall* metrics are two popular performance evaluation measures in text classification. For multiclass text classification, the overall performance is usually measured by two comprehensive metrics, i.e., the micro-averaged F_1 (*micro-F1*) and macro-averaged F_1 (*macro-F1*). By definition, $micro\text{-}F1 = 2P{\cdot}R/(P+R)$, where P is the precision or accuracy of the classification results of the test set, R is the recall for the entire test set to be correctly classified. $macro\text{-}F1 = \text{sum}(F_{1j})/m$ for $j = 1,2,\ldots,m$, and $F_{1j} = 2P_j{\cdot}R_j/(P_j + R_j)$, the F_1 measure for the j-th class, c_j ($j = 1, 2, \ldots, m$), where P_j and R_j are respectively the precision and recall for class c_j, and m is the number of classes.

4.3. Experimental results in multiclass classification tasks

Provided in this section are the results of multiclass text classification using the SVM or kNN classifiers respectively on the 20 Newsgroups, Reuters-21578 and TanCorp corpora with feature sets of different sizes. The performance comparisons among the eight term weighting schemes described before are followed.

4.3.1. Performance comparisons on the 20 Newsgroups corpus

The experimental results of text classification on the 20 Newsgroups English corpus using the linear SVM and kNN ($k = 20$) classifiers are shown in Figs. 1 and 2 respectively. Each curve in the figures represents a different term weighting scheme. The vertical axis indicates the macro-averaged or micro-averaged F_1 values of the text classification performance. The horizontal axis shows the corresponding number of terms.

It can be seen from Fig. 1 that, in the text classification using linear SVM on the 20 Newsgroups corpus, the performances of TF-IGM and RTF-IGM are better than other term weighting schemes for any number of features. Especially the RTF-IGM has obvious advantages that, in terms of the classification accuracy (*macro-F1* or *micro-F1*), it outperforms TF-RF by up to 2.6% and TF-IDF by nearly 3.0%. However, the performances of TF-RF and TF-IDF-ICSDF are slightly better than TF-IDF, and in fact, the differences among them are little. In addition, the TF scheme is obviously inferior to TF-IDF. Particularly, TF-CHI and TF-Prob are much worse than others.

When the kNN ($k = 20$) classification is done on the 20 Newsgroups corpus, the performance differences among the eight term weighting schemes shown in Fig. 2 are greater than those in the SVM case shown in Fig. 1. For any number of features, TF-IGM and RTF-IGM have more obvious advantages over other term weighting schemes than ever. In terms of *macro-F1* (or *micro-F1*), TF-IGM exceeds TF-RF by 1.6% (or 1.8%) and TF-IDF by 5.8% (or 6.0%), while RTF-IGM exceeds TF-RF by 4.4% and TF-IDF by 8.6% on average for more than 1500 features. Of course, the performances of TF-RF and TF-IDF-ICSDF are also better than TF-IDF. But the TF-Prob, TF-CHI and TF schemes perform significantly worse than TF-IDF.

[5] http://ictclas.nlpir.org/.

[6] http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html.

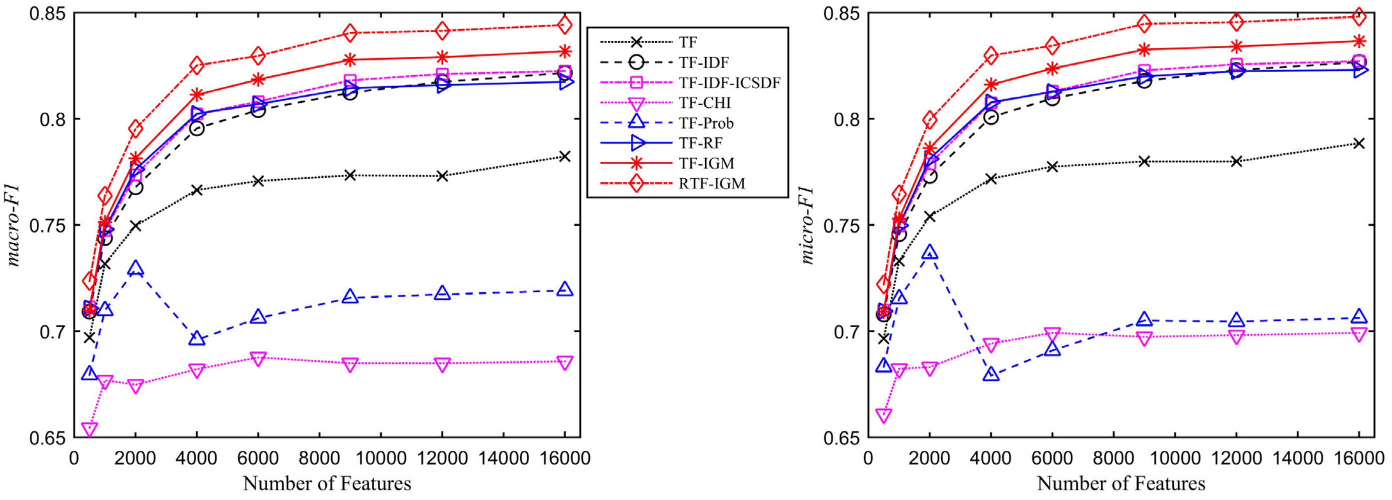

Fig. 1. Macro-averaged F_1 (*macro-F1*) and micro-averaged F_1 (*micro-F1*) measures of multiclass text classification using the linear SVM classifier and eight term weighting schemes on the 20 Newsgroups corpus with different numbers of features.

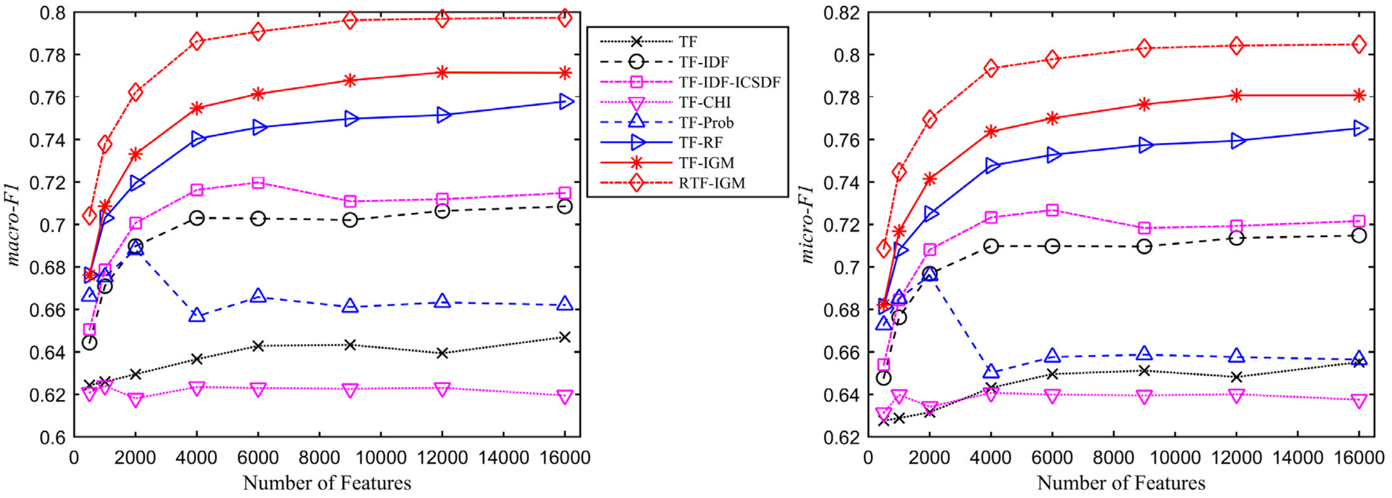

Fig. 2. Macro-averaged F_1 (*macro-F1*) and micro-averaged F_1 (*micro-F1*) measures of multiclass text classification using the *k*NN ($k = 20$) classifier and eight term weighting schemes on the 20 Newsgroups corpus with different numbers of features.

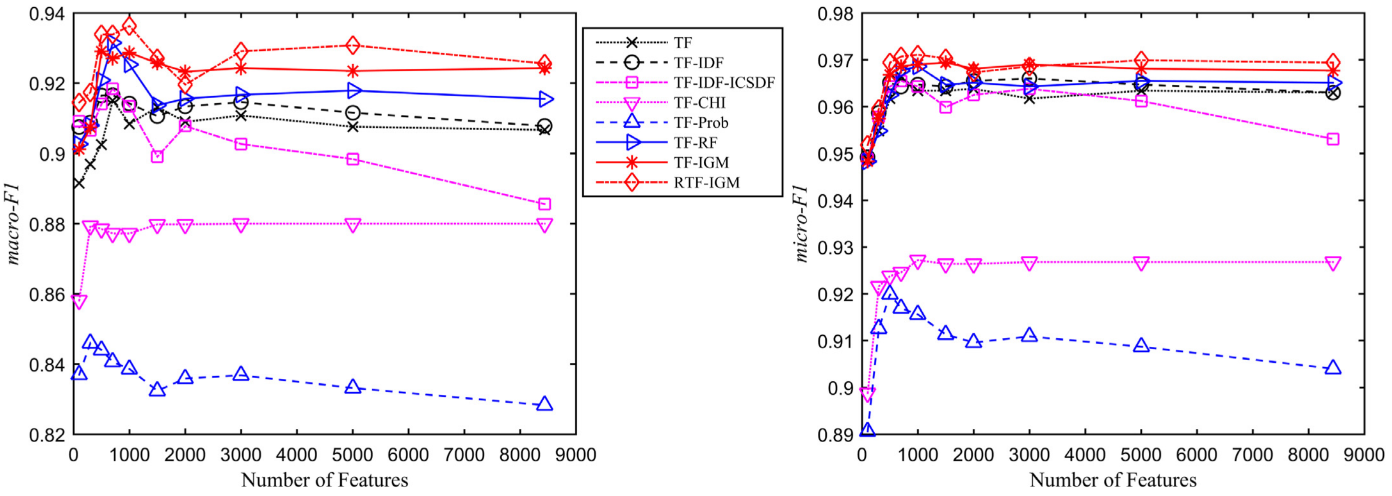

Fig. 3. Macro-averaged F_1 (*macro-F1*) and micro-averaged F_1 (*micro-F1*) measures of multiclass text classification using the linear SVM classifier and eight term weighting schemes on the Reuters-21578 corpus with different numbers of features.

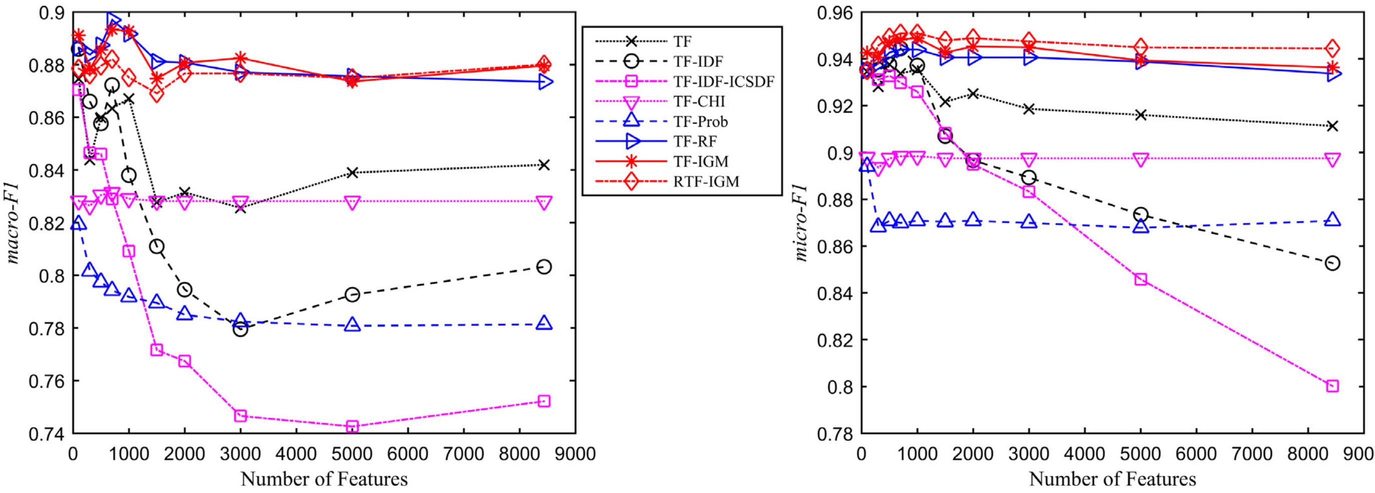

Fig. 4. Macro-averaged F_1 (*macro-F1*) and micro-averaged F_1 (*micro-F1*) measures of multiclass text classification using the *k*NN ($k = 15$) classifier and eight term weighting schemes on the Reuters-21578 corpus with different numbers of features.

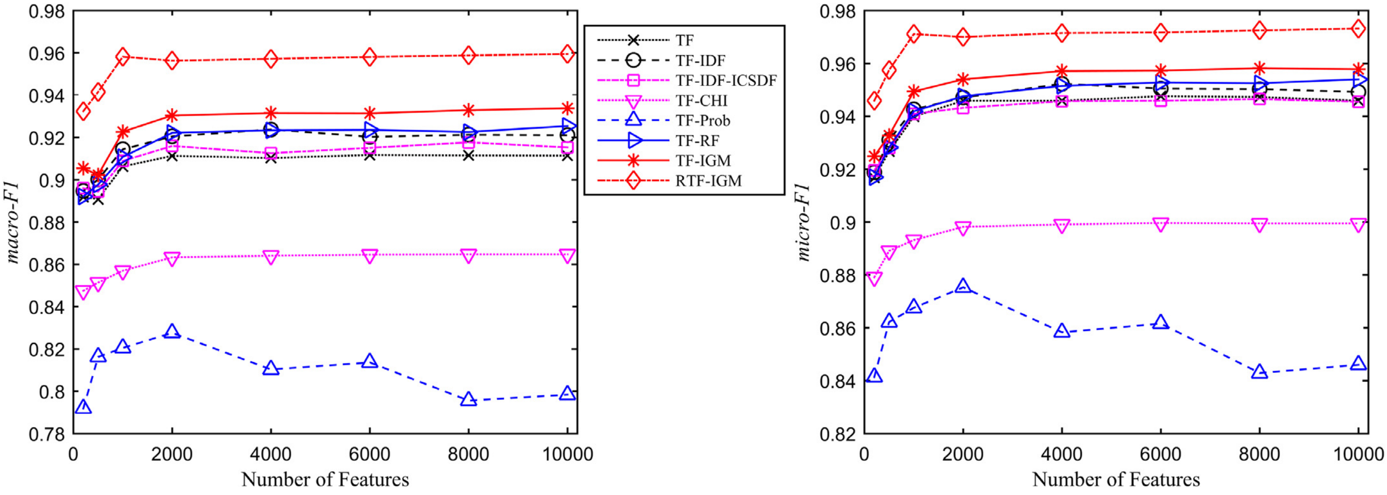

Fig. 5. Macro-averaged F_1 (*macro-F1*) and micro-averaged F_1 (*micro-F1*) measures of multiclass text classification using the linear SVM classifier and eight term weighting schemes on the TanCorp corpus with different numbers of features.

4.3.2. Performance comparisons on the Reuters-21578 corpus

Shown in Figs. 3 and 4 are respectively the experimental results of using the linear SVM and *k*NN ($k = 15$) classifiers for text classification on the Reuters-21578 corpus. Each curve represents a term weighting scheme in the figures, where the vertical direction indicates the classification performance measures and the horizontal direction corresponds to the various numbers of features.

As shown in Fig. 3, while using the linear SVM classifier for text classification on the Reuters-21578 corpus, the performance of TF-Prob and TF-CHI are far worse than other term weighting schemes. However, for the rest six schemes, there are significant differences among their *macro-F1* measures, but little differences among their *micro-F1* measures of the classification results. In most cases with more than 1000 features, the performances of these six schemes are sorted as RTF-IGM > TF-IGM > TF-RF > TF-IDF > TF > TF-IDF-ICSDF.

While classifying text from the Reuters-21578 corpus using the *k*NN ($k = 15$) classifier, it can be found from Fig. 4 that there are large differences among some groups of term weighting schemes. The performances of the TF-Prob, TF-CHI and TF schemes are still relatively poor. However, while using TF-IDF and TF-IDF-ICSDF for term weighting, the classification accuracy, *macro-F1* or *micro-F1*, falls sharply with the increase of the number of features. When the number of features exceeds 3000, TF-IDF-ICSDF becomes the worst and the performance of TF-IDF is close to the inferior TF-Prob. However, TF-IGM, RTF-IGM and TF-RF show obvious advantages over the above five schemes. For example, when the number of features is greater than 1000, in terms of *macro-F1* (or *micro-F1*), TF-IGM exceeds the TF scheme by 4.5% (or 2.3%) on average and TF-IDF by 8.2% (or 5.8%) on average respectively. In addition, the performance differences among the three superior schemes are little. In terms of *macro-F1*, only when the number of features is less than 2000, TF-RF outperforms TF-IGM and RTF-IGM a bit. In terms of *micro-F1*, however, it is shown that RTF-IGM > TF-IGM > TF-RF for all the feature sets of different sizes.

4.3.3. Performance comparisons on the TanCorp corpus

Figs. 5 and 6 show the experimental results of text classification using the linear SVM and *k*NN ($k = 5$) classifiers respectively on the TanCorp Chinese corpus. Each curve represents a term weighting scheme in the figures, where the vertical direction indicates the classification performance measures and the horizontal direction corresponds to the various feature numbers.

As can be seen from Fig. 5, for text classification using linear SVM on the TanCorp corpus, TF-IGM and RTF-IGM perform better as before than all of the other term weighting schemes. In par-

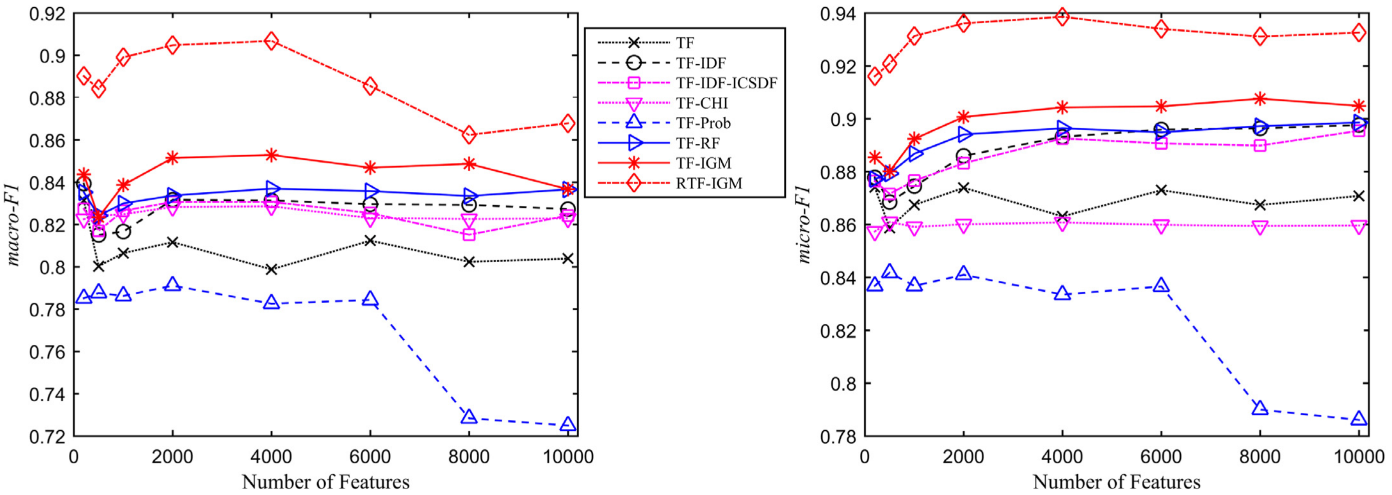

Fig. 6. Macro-averaged F_1 (*macro-F1*) and micro-averaged F_1 (*micro-F1*) measures of multiclass text classification using the *k*NN ($k = 5$) classifier and eight term weighting schemes on the TanCorp corpus with different numbers of features.

Table 2
Performances of eight term weighting schemes in multiclass text classifications using SVM.

Datasets, # Features	Metrics	TF	TF-IDF	TF-IDF-ICSDF	TF-CHI	TF-Prob	TF-RF	TF-IGM	RTF-IGM
20 Newsgroups:									
2000	*macro-F1*	0.7496	0.7678	0.7736	0.6748	0.7292	0.7762	0.7814	**0.7953**
	micro-F1	0.754	0.7731	0.7788	0.6831	0.7366	0.7812	0.7862	**0.7995**
12,000	*macro-F1*	0.7731	0.8174	0.8211	0.6849	0.7173	0.8159	0.829	**0.8414**
	micro-F1	0.7799	0.8229	0.8257	0.6981	0.7045	0.8224	0.834	**0.8455**
Reuters-21578:									
3000	*macro-F1*	0.9108	0.9146	0.9027	0.88	0.8368	0.9167	0.9243	**0.9291**
	micro-F1	0.9617	0.966	0.9638	0.9268	0.9109	0.9643	**0.969**	0.9686
8436	*macro-F1*	0.9067	0.9078	0.8856	0.88	0.8283	0.9155	0.9243	**0.9256**
	micro-F1	0.963	0.963	0.9531	0.9268	0.904	0.9651	0.9677	**0.9694**
TanCorp:									
2000	*macro-F1*	0.9113	0.9205	0.916	0.8633	0.8276	0.9222	0.9304	**0.9563**
	micro-F1	0.9459	0.9471	0.9432	0.8982	0.8753	0.9476	0.954	**0.97**
8000	*macro-F1*	0.9115	0.9213	0.9177	0.8647	0.7956	0.9226	0.9329	**0.9588**
	micro-F1	0.9473	0.9503	0.9465	0.8995	0.8429	0.9525	0.9582	**0.9725**

ticular, the RTF-IGM scheme with the best performance has obvious advantages over others. In terms of *macro-F1* and *micro-F1*, it exceeds TF-IDF by up to 4.4% and 2.9% respectively, and TF-RF by up to 4.7% and 2.9% respectively. And the performances of TF-Prob and TF-CHI are still the worst among the eight schemes of term weighting. As for the rest four schemes such as TF-RF, TF-IDF, TF-IDF-ICSDF and TF, their performances are of little differences.

As shown in Fig. 6, for text classification using *k*NN ($k = 5$) classifier on the TanCorp corpus, only three of the eight term weighting schemes perform better than TF-IDF, and moreover, RTF-IGM is still the best one which has obvious advantages over others and the second is TF-IGM. In terms of *macro-F1* and *micro-F1*, RTF-IGM exceeds TF-RF by up to 7.1% and 4.5%, and TF-IDF by up to 8.2% and 5.7% respectively. Meanwhile, TF-IGM exceeds TF-RF by up to 1.8% and 1.0%, and TF-IDF by up to 2.2% and 1.8% respectively. The performance of TF-Prob is the worst.

4.3.4. Performance improvements of the proposed schemes over others

The experimental results depicted by Figs. 1–6 clearly show that the proposed TF-IGM and RTF-IGM schemes perform obviously better than other term weighting schemes. Although the results were mostly obtained from the reduced feature spaces of the text corpora, the trends of the curves, e.g., in Fig. 2 suggest that TF-IGM and RTF-IGM maintain their superiorities over others as the number of features increases to the maximum when no feature selection is applied to the corpus.

In order to illustrate more clearly the performance improvements of our proposed TF-IGM and RTF-IGM schemes over other term weighting schemes in text classification tasks, part of the experimental data corresponding to Figs. 1–6, i.e., the macro-averaged and micro-averaged F_1 scores obtained by different weighting schemes, are presented in Tables 2 and 3 for multiclass text classifications using SVM and *k*NN respectively.

The performances of different term weighting schemes in both the low-dimensional and high-dimensional feature spaces of each of the three benchmark datasets are exposed in Tables 2 and 3. The maximum values among the macro-averaged or micro-averaged F_1 scores obtained by different weighting schemes are indicated in bold while the secondary maximum values are underlined in the tables. It is thus clear to see the performance improvements of our proposed TF-IGM and RTF-IGM over other weighting schemes. Since TF-IDF is the most popular scheme and TF-RF has been proved superior over many other schemes for term weighting (Lan et al., 2009), we examine the relative performance improvements of our proposed schemes over TF-IDF and TF-RF, as illustrated in Table 4. The maximum relative performance improvements of our proposed schemes over TF-IDF in multiclass text classifications are 4.07% in terms of *macro-F1* or 3.41% in terms of *micro-F1* while using SVM and 13.21% (*macro-F1*) or 12.7% (*micro-F1*) while using the *k*NN classifier. The maximum relative performance improvements of our proposed schemes over TF-RF are 3.92% (*macro-F1*) or 2.81% (*micro-F1*) while using SVM and 8.53% (*macro-F1*) or 6.15%

Table 3
Performances of eight term weighting schemes in multiclass text classifications using *k*NN.

Datasets, # Features	Metrics	TF	TF-IDF	TF-IDF-ICSDF	TF-CHI	TF-Prob	TF-RF	TF-IGM	RTF-IGM
20 Newsgroups:									
2000	*macro-F1*	0.6295	0.6898	0.7007	0.6181	0.6882	0.7195	0.7332	**0.7623**
	micro-F1	0.6316	0.6968	0.7082	0.6342	0.696	0.725	0.7415	**0.7696**
12,000	*macro-F1*	0.6394	0.7064	0.7119	0.6231	0.6633	0.7515	0.7716	**0.7969**
	micro-F1	0.6482	0.7136	0.7193	0.6401	0.6576	0.7594	0.7808	**0.8042**
Reuters-21578:									
3000	*macro-F1*	0.8256	0.7795	0.7466	0.8282	0.7824	0.8771	**0.8825**	0.8767
	micro-F1	0.9186	0.8893	0.8833	0.8975	0.8699	0.9406	0.9449	**0.9475**
8436	*macro-F1*	0.842	0.8032	0.7522	0.8282	0.7814	0.8735	0.8796	**0.8801**
	micro-F1	0.9113	0.8527	0.8002	0.8975	0.8708	0.9337	0.9363	**0.9444**
TanCorp:									
2000	*macro-F1*	0.8117	0.8317	0.8307	0.8283	0.7911	0.8337	0.8515	**0.9048**
	micro-F1	0.8739	0.886	0.8832	0.8601	0.841	0.8941	0.9007	**0.9361**
8000	*macro-F1*	0.8024	0.8293	0.8152	0.8227	0.7284	0.8335	0.8487	**0.8624**
	micro-F1	0.8674	0.8964	0.8899	0.8595	0.79	0.8972	0.9076	**0.9311**

Table 4
Percentages of relative performance improvements of the proposed schemes over TF-IDF and TF-RF.

Datasets, # Features	Metrics	TF-IGM (for SVM)		RTF-IGM (for SVM)		TF-IGM (for *k*NN)		RTF-IGM (for *k*NN)	
		TF-IDF	TF-RF	TF-IDF	TF-RF	TF-IDF	TF-RF	TF-IDF	TF-RF
20 Newsgroups:									
2000	*macro-F1*	1.77%	0.67%	3.58%	2.46%	6.29%	1.90%	10.51%	5.95%
	micro-F1	1.69%	0.64%	**3.41%**	2.34%	6.42%	2.28%	10.45%	**6.15%**
12,000	*macro-F1*	1.42%	1.61%	2.94%	3.13%	9.23%	2.67%	12.81%	6.04%
	micro-F1	1.35%	1.41%	2.75%	**2.81%**	9.42%	2.82%	**12.70%**	5.90%
Reuters-21578:									
3000	*macro-F1*	1.06%	0.83%	1.59%	1.35%	**13.21%**	0.62%	12.47%	-0.05%
	micro-F1	0.31%	0.49%	0.27%	0.45%	6.25%	0.46%	6.54%	0.73%
8436	*macro-F1*	1.82%	0.96%	1.96%	1.10%	9.51%	0.70%	9.57%	0.76%
	micro-F1	0.49%	0.27%	0.66%	0.45%	9.80%	0.28%	10.75%	1.15%
TanCorp:									
2000	*macro-F1*	1.08%	0.89%	3.89%	3.70%	2.38%	2.14%	8.79%	**8.53%**
	micro-F1	0.73%	0.68%	2.42%	2.36%	1.66%	0.74%	5.65%	4.70%
8000	*macro-F1*	1.26%	1.12%	**4.07%**	**3.92%**	2.34%	1.82%	3.99%	3.47%
	micro-F1	0.83%	0.60%	2.34%	2.10%	1.25%	1.16%	3.87%	3.78%

Table 5
Performances of four term weighting schemes when no feature selection is applied to the datasets.

Classifiers	Metrics	TF-IDF	TF-RF	TF-IGM			RTF-IGM		
20 Newsgroups (with 35,642 original features):									
SVM	*macro-F1*	0.8257	0.8235	0.8365	(1.31%,	1.58%)	**0.8467**	(2.54%,	2.82%)
	micro-F1	0.8307	0.8293	0.8413	(1.28%,	1.45%)	**0.8505**	(2.38%,	2.56%)
*k*NN	*macro-F1*	0.6929	0.7568	0.7717	(11.37%,	1.97%)	**0.795**	(14.74%,	5.05%)
	micro-F1	0.7001	0.7649	0.7813	(11.60%,	2.14%)	**0.8031**	(14.71%,	4.99%)
TanCorp (with 72,601 original features):									
SVM	*macro-F1*	0.8671	0.9077	0.9247	(6.64%,	1.87%)	**0.9541**	(10.03%,	5.11%)
	micro-F1	0.9134	0.9473	0.9548	(4.53%,	0.79%)	**0.9698**	(6.17%,	2.38%)
*k*NN	*macro-F1*	0.3577	0.76	0.7751	(116.69%,	1.99%)	**0.8797**	(145.93%,	15.75%)
	micro-F1	0.3917	0.8385	0.8595	(119.43%,	2.50%)	**0.9253**	(136.23%,	10.35%)

Note: (1) Results for Reuters-21578 (with 8436 original features) are already shown in Tables 2, 3 and 4.
(2) Percentages of relative improvements over TF-IDF and TF-RF are shown in parentheses.

(*micro-F1*) while using the *k*NN classifier, as indicated in bold in Table 4. On average, the relative performance improvements of our proposed RTF-IGM over TF-IDF in multiclass text classifications are 3.01% (*macro-F1*) or 1.98% (*micro-F1*) while using SVM and 9.69% (*macro-F1*) or 8.33% (*micro-F1*) while using the *k*NN classifier. The average relative performance improvements of our proposed RTF-IGM over TF-RF are 2.61% (*macro-F1*) or 1.75% (*micro-F1*) while using SVM and 4.12% (*macro-F1*) or 3.74% (*micro-F1*) while using the *k*NN classifier.

The results for the 8436 original features of Reuters-21578 are obtained from experiments without feature selection (FS) on the corpus. Similarly, when no FS is applied to the 20Newsgroups and TanCorp datasets, the performances of the proposed TF-IGM and RTF-IGM schemes as well as TF-IDF and TF-RF are shown in Table 5. It can be seen that the IGM-based schemes show apparent performance improvements as before over TF-IDF and TF-RF on the text corpora without FS. Such a fact is consistent with the trends of the curves in Figs. 1–6. In addition, we can find that the performance of TF-IDF in text classification using the *k*NN classifier becomes severely worse when no FS is applied to the imbalanced corpus, TanCorp, as shown in Table 5. Similar phenomenon for TF-IDF can also be seen on Reuters-21578, as shown by Fig. 4. However, the performances of TF-IGM and RTF-IGM are more stable than TF-IDF as the feature space is of higher dimensionality or even no FS is applied.

Most of the experimental results presented before are obtained from term weighting in the feature spaces reduced by FS based on the popular chi-square statistics (CHI) metric. It can be seen from

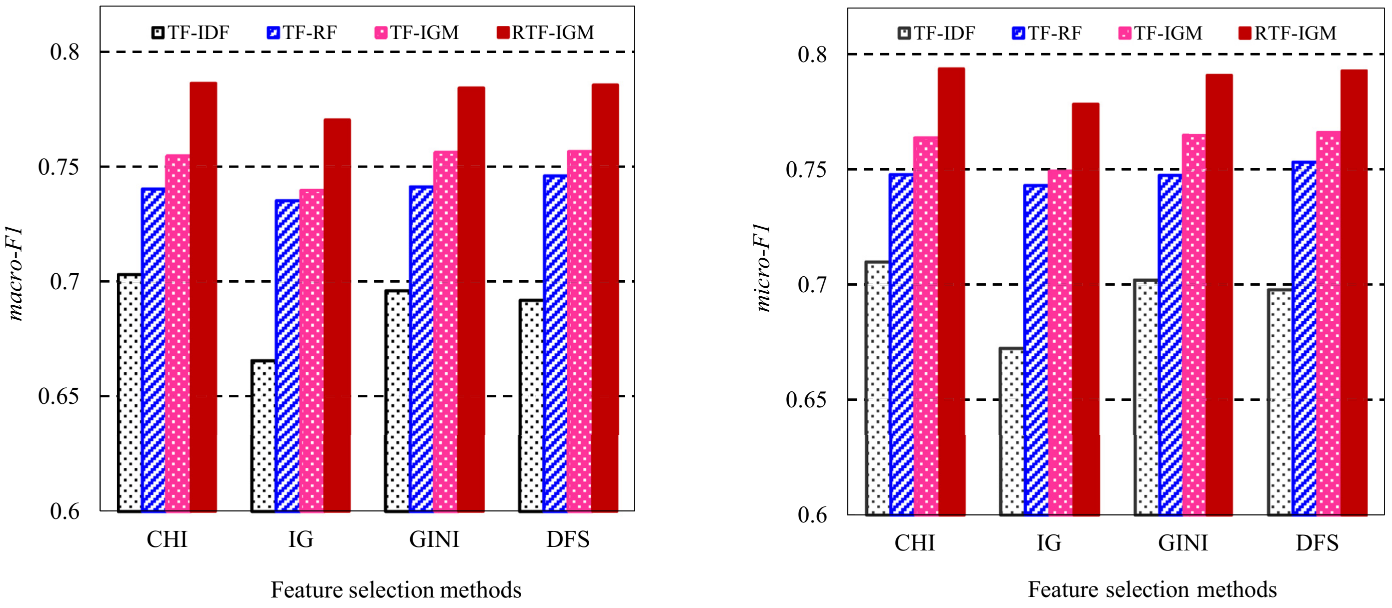

Fig. 7. Macro-averaged F_1 (*macro-F1*) and micro-averaged F_1 (*micro-F1*) scores obtained by four term weighting schemes in multiclass text classification using the *k*NN ($k = 20$) classifier on 20 Newsgroups with 4000 features selected respectively by four feature selection methods.

Figs. 1–6 that, whatever term weighting scheme is used, the accuracy of text classification is not increased significantly but sometimes even decreased as the number of features increases, e.g., over 4000 or when no FS is applied. At the same time, the processing efficiency or classification speed is decreased because of the increased computational overhead for high-dimensional data. So text classification usually involves the feature selection or dimension reduction operation.

Besides CHI (Yang & Pedersen, 1997), we also tried some other FS methods such as information gain (IG) (Yang & Pedersen, 1997), improved Gini index (GINI) (Shang et al., 2007) and distinguishing feature selector (DFS) (Uysal & Gunal, 2012). Whenever a specific FS methods is used, the performances of all the different term weighting schemes are evaluated and compared under the same conditions, e.g., in the same feature space of the corpus with a certain number of features selected by the same FS method. The experimental results show that the performance differences among different term weighting schemes while using other FS methods are similar to those while using CHI for FS. For example, for 4000 features selected respectively by different FS methods from 20 Newsgroups, the performance comparisons among four term weighting schemes such as TF-IGM, RTF-IGM, TF-IDF and TF-RF in multiclass text classification using the *k*NN classifier are shown in Fig. 7. We can notice that when different FS methods are applied, TF-IGM and RTF-IGM always maintain their superiorities over the other two schemes, especially over TF-IDF.

The above experimental results fully demonstrate that the superiorities of TF-IGM or RTF-IGM over other term weighting schemes are independent of FS or any FS method although the amounts of relative performance improvements over other schemes may differ for different feature spaces or FS methods. As a matter of fact, term weighting is generally independent of FS although the accuracy of text classification is also impacted by FS. Instead, FS may sometimes depend on term weighting, e.g., TF-IDF is used as the FS metric (Taşcı & Güngör, 2013). As the two aspects of text representation, term weighting and FS have usually been investigated respectively by researchers (Lan et al., 2009; Ren & Sohrab, 2013; Shang et al., 2007; Uysal & Gunal, 2012; Yang & Pedersen, 1997).

4.4. Experiments and results in binary classification tasks

Although the proposed TF-IGM and RTF-IGM schemes are designed originally for multiclass text classification, we have also evaluated their effectiveness in binary or two-class classification tasks by comparing with other six schemes such as TF, TF-IDF, TF-IDF-ICSDF, TF-CHI, TF-Prob and TF-RF. The corpus used for experiments is Reuters-21578 with 90 classes, which includes 7770 training documents and 3019 test documents according to ModApte split. Binary classifications were conducted by using the linear SVM classifier on ten two-class datasets derived from the Reuters-21578 corpus. Each dataset takes a different one of the top 10 largest classes in the corpus as the positive class while the rest 89 classes are grouped into the negative class. A document in the original corpus may have more than one class labels. It will be relabeled as positive if it has the corresponding original label in its label list. Otherwise, it will be relabeled as negative. Thus every two-class dataset becomes a single-label corpus in which every document is labeled as only one class (either positive or negative) and there are no duplicates across the two classes. The sizes (document numbers) of the positive and negative classes in the training and test subsets of each dataset are shown in Table 6.

The original vocabulary of the Reuters-21578 corpus contains 24,329 distinct terms (features) after numbers, non-alphabetic characters and stop words being removed. To reduce the feature dimensionality, we used CHI_{max} for feature selection. When the top 6000 features are selected, the classification accuracies (F_1 measures) of the 10 classes obtained by eight term weighting schemes are shown in Table 7. For the proposed TF-IGM and RTF-IGM, the IGM components in term weights are calculated from the class-specific DF ratios of the term instead of its class-specific DFs because the positive and negative classes in each dataset are severely imbalanced as shown in Table 6. The maximum values among the F_1 scores obtained by different weighting schemes are indicated in bold while the secondary maximum values are underlined in Table 7. It can be found that TF-RF, TF-IGM and RTF-IGM have more chances to obtain higher accuracies than other weighting schemes in binary classifications for the 10 classes on the corresponding two-class datasets. In fact, the overall scores averaged over all the 10 classes, *micro-F1* and *macro-F1*, obtained by TF-RF,

Table 6
Ten two-class datasets derived from Reuters-21578 with the top 10 classes being the positive classes respectively.

Subsets	Classes	acq	corn	crude	earn	grain	interest	money-fx	ship	trade	wheat
Training	Positive	**1650**	**181**	**389**	**2877**	**433**	**347**	**538**	**197**	**369**	**212**
	Negative	6120	7589	7381	4893	7337	7423	7232	7573	7401	7558
Test	Positive	**719**	**56**	**189**	**1087**	**149**	**131**	**179**	**89**	**117**	**71**
	Negative	2300	2963	2830	1932	2870	2888	2840	2930	2902	2948

Table 7
Performances of eight term weighting schemes in binary classifications on Reuters-21578 with 6000 features selected.

Classes	TF	TF-IDF	TF-IDF-ICSDF	TF-CHI	TF-Prob	TF-RF	TF-IGM	RTF-IGM
acq	0.9628	0.9656	0.9591	0.9293	0.7527	0.9608	**0.9663**	0.9657
corn	0.8687	0.86	0.7191	**0.9322**	0.9032	0.8972	0.8824	0.8991
crude	**0.9034**	0.8951	0.861	0.7215	0.7574	0.898	0.8929	0.8832
earn	0.9849	0.9848	**0.9875**	0.9678	0.9566	0.9831	0.9867	0.9871
grain	0.9104	0.9263	0.8832	0.9164	0.873	**0.9315**	0.9263	0.9301
interest	0.7607	0.7863	0.7444	0.6296	0.1497	0.7764	0.7848	**0.795**
money-fx	0.8189	0.7989	0.7507	0.7356	0.736	0.8167	**0.8189**	0.8045
ship	0.8101	0.8199	0.7582	0.7564	0.7273	0.8323	0.8176	**0.8606**
trade	0.7644	0.7721	0.7245	0.708	0.2317	0.8051	0.7692	**0.8101**
wheat	0.8217	0.8031	0.768	0.9032	**0.9091**	0.8345	0.8182	0.8671
Overall averaged scores:								
micro-F1	0.929	0.9301	0.9151	0.8932	0.8163	0.9316	0.9318	**0.9349**
macro-F1	0.8606	0.8612	0.8156	0.82	0.6997	0.8736	0.8663	**0.8803**

Table 8
Overall performances of eight term weighting schemes in binary classifications for the top 10 classes in Reuters-21578.

# Features, metrics	TF	TF-IDF	TF-IDF-ICSDF	TF-CHI	TF-Prob	TF-RF	TF-IGM	RTF-IGM
1000 features:								
micro-F1	0.9208	0.9186	0.9102	0.893	0.834	0.921	0.9235	**0.9247**
macro-F1	0.8473	0.8369	0.8124	0.8187	0.7065	0.8568	0.8493	**0.8602**
3000 features:								
micro-F1	0.9266	0.9292	0.9187	0.8932	0.8186	0.9301	0.9288	**0.9321**
macro-F1	0.8557	0.8624	0.8318	0.82	0.6954	0.8729	0.8627	**0.8777**
4500 features:								
micro-F1	0.9311	0.9323	0.9175	0.8934	0.8297	0.9322	0.9343	**0.9356**
macro-F1	0.8636	0.8656	0.8253	0.8203	0.7044	0.8749	0.8752	**0.8829**
15,000 features:								
micro-F1	0.9302	0.9296	0.9084	0.8932	0.8312	0.9332	0.9336	**0.9338**
macro-F1	0.8614	0.8584	0.7961	0.82	0.7471	0.8745	0.8679	**0.8746**
24,329 features:								
micro-F1	0.9305	0.9306	0.9039	0.8932	0.8309	**0.9332**	0.9325	0.933
macro-F1	0.8635	0.8582	0.7815	0.82	0.7469	**0.8753**	0.8677	0.8707

TF-IGM and RTF-IGM are greater than those by other schemes. In particular, RTF-IGM obtains the highest scores.

Moreover, we have further explored the performances of the eight term weighting schemes in the low-dimensional feature spaces consisting of 1000, 3000 or 4500 selected features and the high-dimensional feature spaces consisting of 15,000 features or all the 24,329 original features (under no feature selection). The results are listed in Table 8.

As shown in Table 8, on the whole, RTF-IGM performs best among all the term weighting schemes for binary text classification and it is followed by TF-RF or TF-IGM with just slight performance differences. Especially, RTF-IGM obtains higher scores in lower-dimensional feature spaces than TF-RF. For example, the highest *macro-F*1 obtained by RTF-IGM for 4500 features is greater than the highest *macro-F*1 obtained by TF-RF for 24,329 features. Therefore, if taking the efficiency and accuracy of binary text classification into account, RTF-IGM is superior to TF-RF and more preferable than the later. The above experimental results justify the effectiveness of the proposed TF-IGM and RTF-IGM schemes in binary classification tasks. Although the IGM-based schemes are proposed primarily for multiclass text classification, their performances in binary classification tasks are comparable with or even better than TF-RF, a good term weighting scheme for binary text classification (Lan et al., 2009).

4.5. Comparisons against other work

The overall performances of different term weighting schemes in text classification are revealed by the experimental results in multiclass or binary (two-class) classification tasks. The proposed TF-IGM and RTF-IGM schemes are obviously superior to other term weighting schemes, especially for multiclass text classification. Moreover, RTF-IGM wins the best performance in most cases, and it has significant superiority over others. For multiclass text classification, not only they outperform TF-RF on the imbalanced Reuters-21578 corpus, TF-IGM and RTF-IGM perform much better than TF-RF and TF-IDF on the nearly balanced 20 Newsgroups corpus and the imbalanced TanCorp corpus. Even in binary text classification tasks, TF-IGM and RTF-IGM perform as well as or even better than TF-RF. Since TF-RF has been proved to be an excellent term weighting scheme in previous studies (Lan et al., 2009) and validated in our experiments again, it is enough to prove that the schemes based on the IGM model are effective for term weighting in text classification.

In most cases for multiclass or binary text classification, TF-CHI or TF-Prob performs the worst among the eight schemes, although they sometimes perform well in binary text classification. Both of them are inferior to the traditional TF-IDF. Such a result in our experiments is consistent with the reports of some literatures, such as Altınçay and Erenel (2010), Debole and Sebastiani (2003), Lan et al. (2009), Ren and Sohrab (2013), and Wang and Zhang (2013), but contradicts with the results in Deng et al. (2004) and Liu et al. (2009) where TF-CHI or TF-Prob performs better than TF-IDF and some other schemes. Our experiments have also shown that the TF scheme is inferior to TF-IDF in multiclass text classification, which is consistent with what is expected. But the performances of TF and TF-IDF are comparable in binary text classification. Our experiments confirm once again that the performance TF-RF is better than TF-IDF on the whole just as in Lan et al. (2009), especially when the *k*NN classifier is used. While the SVM classifier is used, TF-RF is slightly better than TF-IDF and sometimes both tie. As for the recent TF-IDF-ICSDF scheme, it is slightly superior to TF-IDF only on the nearly balanced 20 Newsgroups corpus. But it performs worse than TF-IDF on the imbalanced Reuters-21578 and TanCorp corpora. Such results in our experiments is not consistent with what is reported in Ren and Sohrab (2013). Particularly, from the overall performances of TF-CHI, TF-Prob and TF-IDF-ICSDF, we have found that introducing the term intra-class distribution factor into term weighting has a negative effect on text classification with degraded performance. This phenomenon is contrary to the conclusions drawn in previous literatures and is discussed in depth later.

The above results are obtained from extensive experiments done on three public benchmark datasets with different characteristics, e.g., class-balanced or imbalanced, English or Chinese text, quite different lengths of the vocabularies, moderate or large numbers of documents, etc. At the same time, multiple feature sets consisting of various numbers of features are selected from each dataset for the experiments. As we know, text classification usually involves the feature selection operation for the sake of classification efficiency and accuracy. By feature selection based on the popular chi-square statistics (CHI) metric, our experiments on each dataset were done with different feature sets so that the performances of different term weighting schemes can be exposed and compared fully in various feature spaces of different-sized dimensions including the case without feature selection. Additionally, we also tried some feature selection methods other than CHI. In any of the above cases, all the different term weighting schemes are tested and compared in the same feature space of the corpus. Moreover, the performances of eight typical schemes for term weighting in either multiclass or binary text classification are evaluated by using the popular SVM and *k*NN classifiers respectively. Finally, it should be pointed out that all the experiments are for single-label text classification and all the datasets are preprocessed through duplicate checking and removing to ensure that there are no duplicates labeled as different classes. By doing so, the reliability and credibility of the experimental results are ensured.

However, the experimental results in the previous studies on term weighting are mostly obtained from the binary text classification experiments, where even though multiclass datasets were used, multiple independent binary text classifications were actually done for the positive and negative classes in an one-against-rest way. And moreover, when the experiments of single-label text classification were done on a corpus, e.g., Reuters-21578, there is no statement or sign in the literatures that duplicate documents are removed from the corpus. Hence, some of their results are doubtful. In fact, a document in the original Reuters-21578 corpus may have more than one class labels. As we know, single-label classification means that each document is labeled as or classified into only a single class, which is different from multilabel classification where a document may be assigned to multiple classes. When a multilabel document was transformed into the single-label form, duplicate documents were generated, each labeled as a different class. So the Reuters-21578 dataset in the single-label form[7] contains many duplicate documents labeled as different classes. For example, the seven test documents having the same IDs (filenames) being “0010340” are duplicates generated from a multilabel document and they are placed respectively in seven classes (folders) including *grain, wheat, corn,* etc. Since only single-label text classification is implemented in the experiments, the duplicate documents labeled as different classes in the test set will be classified into the same single class by the classifier. So, keeping duplicates labeled as different classes in the dataset will result in incorrect or misleading performance measurement of single-label text classification. In addition, the Reuters-21578 dataset also contains some duplicates labeled as the same classes. For example, the two training documents which IDs are “0006014” and “0006071” respectively are duplicates of the *trade* class. The existence of duplicate documents labeled as different or the same classes in the training set also affects the statistics of the term distribution. Therefore, the corpus should be preprocessed through duplicate removing. As described in Section 4.1, our experiments of single-label text classification were done on datasets with duplicates being removed or without duplicates labeled as different classes. That is one of the differences between our experiments and other work. Taking the Reuters-21578 ModApte Top 10 dataset for example, we first removed the redundant duplicates labeled as the same classes while keeping one instance in each group of such duplicates and then removed all the duplicates labeled as different classes. So there are 1.1% of the first type of duplicates and 18.3% of the second type of duplicates removed from the training set. And there are 0.6% and 16.1% respectively of the two types of duplicates removed from the test set.

4.6. Discussions

Although some STW schemes are proposed by introducing the term intra-class distribution factor into term weighting, we think that while weighting a term according to its class distinguishing power, its inter-class distribution (across different classes) should be the main consideration but its intra-class distribution (within a class) can be ignored. As is known to all, a term with much higher frequency or probability of occurrence in a specific class than in other classes can serve as a good discriminator for that class even if it appears in only part of the documents of that class. For example, the news articles containing “*fencing*” belong to the *sports* class. The term “*fencing*” can be used to distinguish between *sports* news and others. However, it does not stand for *sports*, but only one of the subclasses of *sports*. Although the probability of the term’s occurrence is low within the *sports* class, it is much higher than in other classes. The term “*fencing*” should be assigned a large weight. But as long as taking the intra-class distribution into account, its weight is significantly reduced. Instead, a common term occurring frequently in most or even all of the classes of text has very weak class distinguishing power and should be assigned a small weight, but the introduction of the intra-class distribution factor raises its weight. Therefore, we think that the introduction of the intra-class distribution factor may have negative effects on term weighting and text classification.

[7] http://disi.unitn.it/moschitti/corpora.htm.

A typical example is TF-Prob which introduces the intra-class distribution factor on the basis of TF-RF. TF-Prob was expected to perform better than TF-RF and TF-IDF. But in fact, it is not so as shown in our experimental results and the results reported in some papers (Altinçay & Erenel, 2010; Ren & Sohrab, 2013; Wang & Zhang, 2013). As analyzed and predicted before, TF-CHI also performs worse than the traditional TF-IDF and TF schemes in our experiments. TF-Prob and TF-CHI not only introduces the term intra-class distribution factor, but also take the intra-class and inter-class distributions as equally important, which lead to unreasonable term weighting. In fact, the intra-class distribution is less important than the inter-class distribution to measure a term's contribution to classification. In addition, TF-IDF-ICSDF is not as good as the expectation too, sometimes performs worse than TF-IDF. That is partly resulted from introducing the term intra-class distribution factor.

The main purpose of introducing the term intra-class distribution factor into some STW schemes is to assign greater weights to those terms representative of some class(es), because a term with greater class representativeness distributes more uniformly within the specific class(es) than others. However, a term's class representativeness is not equivalent to its class distinguishing power. A term with high class representativeness generally has strong class distinguishing power. Conversely, it is not true, i.e., a term with strong class distinguishing power may have low class representativeness. For a term, the judgment of its class distinguishing power depends mainly on the non-uniformity of its inter-class distribution while the judgment of its class representativeness must also take the uniformity of its intra-class distribution into account besides its inter-class distribution. On the other hand, the terms with class representativeness are minority after all, i.e., the terms representing a whole class are relatively few in the text corpus. Text classification depends mainly on the terms with strong class distinguishing power. Most of the terms with strong class distinguishing power represent only one or a few implicit or explicit subclasses but not the whole class, especially for large classes. Only when the text corpus contains a lot of terms with class representativeness or the text classes are of small size, introducing the intra-class distribution factor into term weighting is somewhat reasonable. But, by contrast, weighting terms by their class distinguishing powers or inter-class distributions is more reliable and effective.

As shown in the experiments, the term weighting schemes such as TF-IGM, RTF-IGM and TF-RF, where terms are weighted by their class distinguishing powers or inter-class distributions only, are consistently superior to other supervised and traditional ones.

Furthermore, the proposed TF-IGM scheme as well as its improved version performs better than TF-RF. Although TF-RF performs well in binary text classification, but just like TF-Prob and TF-CHI in multiclass text classification it also weights a term by grouping multiple classes into a single negative class, thus resulting in a deviation from the optimal term weighting. That is illustrated by the example described in Section 3.4. However, TF-IGM takes the fine-grained distribution of a term across multiple classes into account. So it makes term weighting more reasonable and further improve the performance of multiclass text classification.

In addition to multiclass text classification, TF-IGM and RTF-IGM are also effective for binary text classification where their performances are comparable with or even better than TF-RF as validated by the experiments. Such a fact justifies once again the effectiveness of the IGM model in measuring a term's class distinguishing power or importance in text classification.

The proposed IGM model is promised to have a wider range of applications. Not only for term weighting, the IGM or TF-IGM scheme can also act as a new method for feature selection or dimension reduction. What's more, as a method of text representation, TF-IGM and its variants can be applied to sentiment analysis, information filtering and some other tasks of Web data analysis or retrieval based on text mining, e.g., the classification or filtering of online news, blog articles, online literatures, Web pages, etc.. In addition, the proposed IGM model can be used as a new measure of sample distribution non-uniformity or unevenness beyond the well-known variance in mathematics or entropy in information theory. So it is also likely to replace the variance or entropy on some other occasions.

5. Conclusions and future work

We have proposed a new supervised term weighting scheme, TF-IGM, for text classification. It offers the following features:

(1) TF-IGM adopts a new statistical model called IGM (inverse gravity moment) to characterize the inter-class distribution and measure the class distinguishing power of a term in the corpus so that terms with stronger class distinguishing power are to be assigned greater weights than others in text representation.
(2) TF-IGM takes into account the fine-grained inter-class distribution of a term across different classes of text so that the calculated weight can reflect the term's importance in text classification more realistically than by other schemes and the classification accuracy is thus improved.
(3) TF-IGM is adaptive to different text corpora or text classification tasks by providing several options or adjustable parameters so as to obtain the optimal performance.

It is proved by extensive experiments on public benchmark datasets that TF-IGM is consistently superior to the famous TF-IDF and the state-of-the-art supervised term weighting schemes. Moreover, if the square root of term frequency (TF) is used as the local weighting factor instead of the raw TF, the improved version, RTF-IGM, performs best in most cases. TF-IGM is especially suitable for multiclass text classification applications. Nevertheless, the experimental results show that it is also applicable to binary text classification.

Through comparative studies on various term weighting schemes for text classification, we arrive at the following additional conclusions:

- Weighting terms by their class distinguishing power is more effective and reliable than by their class representativeness.
- A term's class distinguishing power is determined mainly by its inter-class but not intra-class distribution.
- Introducing the term intra-class distribution factor is likely to result in improper term weighting which impairs the performance of text classification.

In the future, we will conduct comparative studies on the IGM model as a new measure of sample distribution non-uniformity and the traditional statistical models such as variance and entropy. Meanwhile, the scope of experiments for text classification will be expanded, e.g., including multilabel text classification experiments. The possibility of applying the IGM model to feature dimension reduction and sentiment analysis will also be investigated.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61379109) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20120162110077).

References

Altinçay, H., & Erenel, Z. (2010). Analytical evaluation of term weighting schemes for text categorization. *Pattern Recognition Letters, 31*(11), 1310–1323. http://doi.org/10.1016/j.patrec.2010.03.012.

Buckley, C. (1985). *Implementation of the SMART information retrieval system*. Ithaca: Cornell University Technical report.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2*(3), 1–27. http://doi.org/10.1145/1961189.1961199.

Chen, K., & Zong, C. (2003). A new weighting algorithm for linear classifier. In *Proceedings of 2003 international conference on natural language processing and knowledge engineering* (pp. 650–655). Beijing, China: IEEE. http://doi.org/10.1109/NLPKE.2003.1275987.

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

Deng, Z., Luo, K., & Yu, H. (2014). A study of supervised term weighting scheme for sentiment analysis. *Expert Systems With Applications, 41*(7), 3506–3513. http://doi.org/10.1016/j.eswa.2013.10.056.

Deng, Z., Tang, S., Yang, D., Zhang, M., Li, L., & Xie, K. (2004). A comparative study on feature weight in text categorization. In *Advanced web technologies and applications, lecture notes in computer Science: vol. 3007* (pp. 588–597). Hangzhou, China: Springer Berlin Heidelberg. Proceeding. http://doi.org/10.1007/978-3-540-24655-8_64.

Dumais, S. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers, 23*(2), 229–236. http://doi.org/10.3758/BF03203370.

Erenel, Z., & Altınçay, H. (2012). Nonlinear transformation of term frequencies for term weighting in text categorization. *Engineering Applications of Artificial Intelligence, 25*(7), 1505–1514. http://doi.org/10.1016/j.engappai.2012.06.013.

How, B. C., & Narayanan, K. (2004). An empirical study of feature selection for text categorization based on term weightage. In *Proceedings of the IEEE/WIC/ACM international conference on web intelligence (WI'04)* (pp. 599–602). http://doi.org/10.1109/WI.2004.10060.

Hwee, T. N., Wei, B. G., & Kok, L. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of SIGIR-97, 20th ACM international conference on research and development in information retrieval* (pp. 67–73). http://doi.org/http://doi.acm.org/10.1145/258525.258537.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*(1), 11–21.

Jones, K. S. (2004). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 60*(5), 493–502.

Lan, M., Tan, C. L., Su, J., & Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(4), 721–735. http://doi.org/10.1109/TPAMI.2008.110.

Lertnattee, V., & Theeramunkong, T. (2004). Analysis of inverse class frequency in centroid-based text classification. *IEEE International Symposium on Communications and Information Technologies 2004 (ISCIT 2004), 2*, 1171–1176. http://doi.org/10.1109/ISCIT.2004.1413903.

Liu, Y., Loh, H. T., & Sun, A. (2009). Imbalanced text classification: A term weighting approach. *Expert Systems with Applications, 36*(1), 690–701. http://doi.org/10.1016/j.eswa.2007.10.042.

Luo, Q., Chen, E., & Xiong, H. (2011). A semantic term weighting scheme for text categorization. *Expert Systems with Applications, 38*(10), 12708–12716. http://doi.org/10.1016/j.eswa.2011.04.058.

Nguyen, T. T., Chang, K., & Hui, S. C. (2013). Supervised term weighting centroid-based classifiers for text categorization. *Knowledge and Information Systems, 35*(1), 61–85. http://doi.org/10.1007/s10115-012-0559-9 .

Peng, T., Liu, L., & Zuo, W. (2014). PU text classification enhanced by term frequency-inverse document frequency-improved weighting. *Concurrency Computation Practice and Experience, 26*(3), 728–741. http://doi.org/10.1002/cpe.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Porter, M. F. (2006). An algorithm for suffix stripping. *Program, 40*(3), 211–218. http://doi.org/10.1108/00330330610681286.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106. http://doi.org/10.1023/A:1022643204877.

Ren, F., & Sohrab, M. G. (2013). Class-indexing-based term weighting for automatic text classification. *Information Sciences, 236*, 109–125. http://doi.org/10.1016/j.ins.2013.02.029.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management, 24*(5), 513–523. http://doi.org/10.1016/0306-4573(88)90021-0.

Schütze, H., Hull, D. A., & Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 229–237). Seattle WA: ACM.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47. http://doi.org/10.1145/505282.505283.

Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., & Wang, Z. (2007). A novel feature selection algorithm for text categorization. *Expert Systems with Applications, 33*(1), 1–5. http://doi.org/10.1016/j.eswa.2006.04.001.

Tan, S., Cheng, X., Ghanem, M. M., Wang, B., & Xu, H. (2005). A novel refinement approach for text categorization. In *Proceedings of the 14th ACM international conference on information and knowledge management* (pp. 469–476). http://doi.org/10.1145/1099554.1099687.

Tang, C., Li, L., & Huang, L. (1995). *Data structures: described in c language*. Beijing, China: Higher Education Press.

Taşcı, Ş., & Güngör, T. (2013). Comparison of text feature selection policies and using an adaptive framework. *Expert Systems with Applications, 40*, 4871–4886. http://doi.org/http://dx.doi.org/10.1016/j.eswa.2013.02.019.

Uysal, A. K., & Gunal, S. (2012). A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems, 36*, 226–235. http://doi.org/10.1016/j.knosys.2012.06.005.

Wang, D., & Zhang, H. (2013). Inverse-category-frequency based supervised term weighting schemes for text categorization. *Journal of Information Science and Engineering, 29*(2), 209–225.

Wei, B., Feng, B., He, F., & Fu, X. (2011). An extended supervised term weighting method for text categorization. In J. J. Park, H. Jin, X. Liao, & R. Zheng (Eds.), *Proceedings of the international conference on human-centric computing 2011 and embedded and multimedia computing 2011 (HumanCom and EMC 2011), lecture notes in electrical engineering: Vol. 102* (pp. 87–99). DordrechtNetherlands: Springer,. http://doi.org/10.1007/978-94-007-2105-0.

Xuan, N. P., & Quang, H. L. (2014). A new improved term weighting scheme for text categorization. In V.-N. Huynh (Ed.). *Knowledge and systems engineering*: Vol. 244. Switzerland: Springer International Publishing *http://doi.org/10.1007/978-3-319-02741-8_23.*

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the fourteenth international conference on machine learning (ICML'97)* (pp. 412–420). http://doi.org/10.1093/bioinformatics/bth267.