Mathematical Biosciences and Engineering

MBE, 19(5): 5223–5240.
DOI: 10.3934/mbe.2022245
Received: 06 September 2021
Revised: 27 February 2022
Accepted: 04 March 2022
Published: 23 March 2022

http://www.aimspress.com/journal/MBE

*Research article*

# A new document representation based on global policy for supervised term weighting schemes in text categorization

**Longjia Jia[1] and Bangzuo Zhang[2,*]**

[1] School of Mathematics and Statistics, Northeast Normal University, Changchun, China
[2] School of Information Science and Technology, Northeast Normal University, Changchun, China

* **Correspondence:** Email: zhangbz@nenu.edu.cn; Tel: +8643185099108.

**Abstract:** There are two main factors involved in documents classification, document representation method and classification algorithm. In this study, we focus on document representation method and demonstrate that the choice of representation methods has impacts on quality of classification results. We propose a document representation strategy for supervised text classification named document representation based on global policy (*DRGP*), which can obtain an appropriate document representation according to the distribution of terms. The main idea of *DRGP* is to construct the optimization function through the importance of terms to different categories. In the experiments, we investigate the effects of *DRGP* on the 20 Newsgroups, Reuters21578 datasets, and using the *SVM* as classifier. The results show that the *DRGP* outperforms other text representation strategy schemes, such as Document Max, Document Two Max and global policy.

**Keywords:** document representation strategy; global policy; text categorization; machine learning

## 1. Introduction

Text categorization (*TC*) is a task of automatically classifying unlabeled natural language documents into a predefined set of semantic categories. As the first and a vital step, text representation converts the content of a textual document into a compact format so that the document can be recognized and classified by classifiers [1]. The vector space model (*VSM*) is the most widely used text representation model in text categorization. In VSM, a document is represented as a vector in term spaces, such as $d = \{t_1, t_2, \ldots, t_n\}$, where $n$ is the number of features in a corpus. The value of $t_i$

represents how much the term $t_i$ contributes to the semantics of document $d$. The terms in *VSM* are extracted from training set. They can be words, phrases, or n-grams, etc [2].

Each document in datasets is represented as a corresponding vector in vector space. The elements in each vector are weighted by term weighting methods. Most studies of term weighting methods for *TC* has showed that supervised term weighting methods are superior to unsupervised term weighting methods [3]. The traditional term weighting methods for *TC* are usually borrowed from *IR* and belong to the unsupervised term weighting methods. The simplest one is binary representation. The most popular one is *tf***idf*. Note that the *tf* here also has various variants such as raw term frequency, *log* (*tf*), *log* (*tf* + 1), or *log* (*tf*) + 1. Besides, the *idf* factor (usually computed as $log(N/n_i)$) also has a number of variants such as $log\ (N/n_i + 1))$, $log\ (N/n_i) + 1)$, and $log\ (N/n_i)$ -1), etc [3]. TC is a supervised learning task because the category labels of training documents are available in advance. Generally, this known information has been used in supervised term weighting methods in the following ways. One approach is to weight terms by adopting feature selection metrics such as chi-square, information gain, gain ratio, etc. The purpose of feature selection is to select the feature with the most discriminative ability for the current classification problem, reduce the feature dimension of the data set, and improve the classification efficiency and accuracy. Another approach is to weight terms in the interaction with a text classifier. For example, in [4], terms are weighted using an iterative approach involving the *k-Nearest Neighbor* at each step. For each iteration, the weight is adjusted according to the classification results on the evaluation set. However, this method is usually time-consuming, especially when dealing with large-scale data problems.

The difference between unsupervised term weighting methods and supervised term weighting methods is that supervised term weighting methods use class information in training set. However, most of the existing methods did not discuss the representation of test documents for supervised term weighting methods [5]. Since test documents do not have any class information and supervised term weighting methods require it. A test document can be first represented as $|C|$ different vectors by using estimated distribution of each class, $c_j$, and then it has to be represented as one vector that well describes the document in vector space. $|C|$ is the number of classes.

There are two major strategies, local policy and global policy [6–8]. In local policy, each test document in the independent binary classification task will be represented as a single vector. This means that the vector representation of each document is not an independent vector but a corresponding vector collection which combines with specific binary classification tasks. Global policy has been widely used. Each document will have a global independent representation. In most classification tasks, each document is generally assigned to one category and labeled with the most similar class label. For example, when using *k-Nearest Neighbor* classifier, the category of an unknown document will obtain the nearest $k$ documents by calculating the Euclidean distance (or Mahalanobis distance or Manhattan distance) from other documents, and then determine the category of the unknown document according to the category of these documents. Thus, most of the classification tasks are regarded as single label task and use global policy [9]. In addition, there are some models based on neural networks [10–12]. This study focuses on models based on global policy, Younghoong Ko proposed Word Max (*W-Max*), Document Max (*D-Max*) and Document Two Max (*D-TMax*) to optimize text representation method and improve the classification performance [9], the idea behind *W-Max* is same as traditional global policy method. In these three methods, any method cannot always obtain optimal values for different data sets. These methods are discussed in Section 2.

Due to the above aspects, is there a relatively general strategy for supervised term weighting

schemes, and, if yes, which one can achieve better performance? This is the first question we wish to address in this study.

In this study, we investigate several well-known document representation strategies, including "global policy", *W-Max*, *D-Max*, and *D-TMax* for supervised term weighting schemes. Since we have not discovered similar work presently, this investigation is significant and valuable in document representation strategy for supervised term weighting schemes in automatic text categorization. At the same time, a new document representation strategy for supervised term weighting schemes is proposed. These five document representation strategies are tested on two famous document collections, i.e., Reuters-21578 (skewed category distribution) and 20 Newsgroups (uniform category distribution). We discuss the experimental results in detail and draw conclusions from different aspects.

The remainder of this paper is organized as follows. We briefly review several document representation strategies in Section 2. Section 3 introduces our proposed document representation strategy, as well as a short discussion of the method. We show experimental results in Section 4, and finally, we draw conclusions and discuss future work in Section 5.

## 2. A brief review of document representation strategies

In the scenario of text categorization, an indexing procedure which converts the raw document into a vector representation is usually necessary since text documents cannot be directly interpreted by a classifier. Document representation is thereby one of the essential components for the construction of a classifier. There are some different types of document representation methods: *VSM*, Latent Semantic Indexing (*LSI*) [13] and Latent Dirichlet Allocation (*LDA*) [14] and the representation method based on word vector model, such as Word2Vec [15].

*LSI* approximates the source space with fewer dimensions which uses matrix algebra technique. Probabilistic Latent Semantic Indexing (*PLSI*) has a more solid statistical foundation compared with *LSI*, since it's based on the likelihood principle and defines a proper generative model of the data [16]. Blei, et al. proposed a more widely used topic model, *LDA* after *PLSI*. It can recognize the latent topics of documents and use topic probability distribution for representations. *Word2vec* model and application by Mikolov et al. have attracted a great amount of attention. Its core idea is to get the vectorial representation of words through the context of words. There are two methods: *CBOW* and Skip-gram. The vector representations of words learned by *word2vec* models have been shown to carry semantic meanings. In this representation method, words with similar semantics will have similar vector representation. In this section, we briefly review several document representation strategies based on *VSM*, the most classical method.

### *2.1. Global policy*

In local policy, each test document in the independent binary classification task will be represented as a single vector.

Global policy is defined as follows.

$$TW(t) = \max_{i=1}^{|C|} TW(t, c_i) \tag{1}$$

where *TW(t)* is the final weight of a term *t*; $TW(t,c_i)$ is weight of term *t* in category $c_i$ obtained with supervised term weighting methods. In the process of initial representation, a test document can be

represented as $|C|$ different vectors. After using appropriate selection policy, it can be represented as one vector which well describes the document. Global policy selects the maximum term value among all categories for each term. Although this method is effective in some cases, it can not ensure that it has the ability to select the most effective term weighting vector for current test samples. Since test documents do not have any class information and supervised term weighting schemes required. Besides local policy and global policy, Younghoong Ko proposed the following three solutions for this problem, i.e., *W-Max*, *D-Max* and *D-TMax* [9]. Now, they are described as follows. Based on the idea of global policy, there are also some methods, such as the sum $f_{sum}(t_k) = \sum_{i=1}^{|C|} f(t_k, c_i)$, or the weighted sum $f_{wsum}(t_k) = \sum_{i=1}^{|C|} P(c_i) f(t_k, c_i)$, or the maximum $f_{max}(t_k) = max_{i=1}^{|C|}(t_k, c_i)$.These functions try to capture the intuition that the best terms for one category are the ones distributed most differently in the sets of positive and negative examples of the category [17]. Now, we described Younghoong Ko 's methods as follows.

### *2.2. W-Max*

Each term's value of term weighting vector will be replaced by the maximum value of the corresponding dimension's term weight in all categories. After comparing with global policy, we may find that they have the same idea.

### *2.3. D-Max*

The sum of all term weights in each term weighting vector is first calculated and then one term weighting vector with the maximum sum value is selected as the document representation vector.

### *2.4. DT-Max*

The sum of all term weights in each term weighting vector is calculated and then two term weighting vectors with the two largest sum values are selected. Then the term weighting vector is constructed by choosing the higher term weighting value from the selected two term weighting vectors for each corresponding dimension's term weight.

To discuss the difference between the above strategies, we take some examples as follows. Assuming that there is a training set $D = \{d_1, d_2, \ldots, d_n\}$ with $m$ terms, where $n$ is the number of documents and these documents belong to $|C|$ categories. Corresponding to category set $C = \{c_1, c_2, \ldots, c_{|C|}\}$, there is a term weighting vector set $V = \{v_1, v_2, \ldots, v_{|C|}\}$. The matrix $M$ (consist of $v_1$ to $v_{|C|}$) is defined as follows.

$$M = \begin{bmatrix} t_{11} & t_{21} & \cdots & t_{1j} & \cdots & t_{1m} \\ t_{21} & t_{22} & \cdots & t_{2j} & \cdots & t_{2m} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ t_{i1} & t_{i2} & \cdots & t_{ij} & \cdots & t_{im} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ t_{|c|1} & t_{|c|2} & \cdots & t_{|c|j} & \cdots & t_{|c|m} \end{bmatrix} \tag{2}$$

In Eq (2), $t_{ij}$ is *j-th* element of the term weighting vector for category $c_i$, *it is calculated using a certain term weighting method* (such as *tf*tf, tf*or* or other term weighting *methods*). Assuming that the term weighting vector of test set is $v_d = \{w_1, w_2, \ldots, w_m\}$, $w_k$ is the *k-th* element of $v_d$.

When *W-Max* is used, $w_k$ can be defined as follows.

$$w_k = \max_{i=1}^{|C|}(w_{ik}); k \in [1, m] \tag{3}$$

In Eq (3), $w_{ik}$ is the *k-th* element of the vector when category $c_i$ is used as a positive category. When *D-Max* is used, the sum of all term weights in each vector $v_i$ is first calculated.

$$sum_i = \sum_{j=1}^{m} t_{ij} \tag{4}$$

After sorting all $sum_i$, *D-Max* selects the maximum value $sum_{maxi}$. The term weighting vector corresponding to $sum_{maxi}$ will be selected as $v_d$.

When *D-TMax* is used, the sum of all term weights in each vector $v_i$ is first calculated using the above Eq (4). Then the two largest sum values are selected, which are labeled as $sum_x$ and $sum_y$, respectively. The subscript $x$ and $y$ are indexs of different categories. The *k-th* element of $v_d$ is calculated as Eq (5).

$$w_k = max(t_{xk}, t_{yk}); x \neq y \in [1, c], k \in [1, m] \tag{5}$$

When compared with *W-Max* (global policy) method, the other two methods achieved good performance in Youngjoong Ko's experiments, i.e., *D-TMax* achieved good performance in Reuters21578 (skewed category distribution) while *D-Max* achieved good performance in 20 Newsgroups (uniform category distribution) [8]. Through experiments on the two famous document collections, we can draw a conclusion that the same method may have different effects on different data set.

## 3. Methodology

Our review of document representation shows that different data sets require different methods to achieve good performance. How can we know the effect of document representation method before we choose it? In other words, when selecting a document representation strategy for an unknown data set, which method is a better choice? This is the second question we asked in this study. We will present the answer at the end of this paper. Table 1 records the numbers of documents which contain term $t_k$ and do not contain term $t_k$ under category $c_i$ and $\overline{c_i}$. Usually, $d >> a, b, c$.

**Table 1.** The Contingency Table for Category $c_i$ and Term $t_k$.

| | $t_k$ | $\overline{t_k}$ |
|---|---|---|
| Positive Category: $c_i$ | $a$ | $b$ |
| Positive Category: $\overline{c_i}$ | $c$ | $d$ |

The improper selecting of document representation strategy would lead to the problem of inappropriate to assign the weight to terms. A test document can be first represented as $|C|$ different vectors by using estimated distribution of each category. For some categories, the weight they assign to terms would has a negative impact on the role of terms in classification. To illustrate this, suppose

the training set is skewed with 19 documents, 5 terms and 5 categories. The relationship between term, document, and category is shown in Table 2. The number in Table 2 represents the times that a term occurs in a document.

**Table 2.** The relationship between term, document, and category.

| category | document | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ |
|---|---|---|---|---|---|---|
| $C_1$ | $d_1$ | 0 | 0 | 2 | 19 | 3 |
| $C_1$ | $d_2$ | 0 | 1 | 3 | 0 | 3 |
| $C_1$ | $d_3$ | 0 | 0 | 5 | 16 | 2 |
| $C_1$ | $d_4$ | 4 | 0 | 1 | 15 | 2 |
| $C_1$ | $d_5$ | 0 | 0 | 2 | 18 | 3 |
| $C_2$ | $d_6$ | 0 | 0 | 2 | 14 | 3 |
| $C_2$ | $d_7$ | 0 | 1 | 3 | 0 | 3 |
| $C_2$ | $d_8$ | 0 | 0 | 5 | 13 | 2 |
| $C_2$ | $d_9$ | 4 | 0 | 1 | 11 | 2 |
| $C_2$ | $d_{10}$ | 0 | 0 | 2 | 17 | 3 |
| $C_3$ | $d_{11}$ | 0 | 0 | 3 | 0 | 3 |
| $C_3$ | $d_{12}$ | 0 | 0 | 1 | 1 | 3 |
| $C_3$ | $d_{13}$ | 0 | 1 | 1 | 0 | 2 |
| $C_4$ | $d_{14}$ | 1 | 99 | 3 | 2 | 1 |
| $C_4$ | $d_{15}$ | 2 | 99 | 1 | 1 | 1 |
| $C_4$ | $d_{16}$ | 1 | 99 | 1 | 2 | 1 |
| $C_5$ | $d_{17}$ | 4 | 0 | 3 | 0 | 3 |
| $C_5$ | $d_{18}$ | 4 | 0 | 1 | 0 | 3 |
| $C_5$ | $d_{19}$ | 4 | 1 | 1 | 0 | 2 |

According to some supervised term weighting schemes, here we use *tf*rf* as an example [1]. Based on notations in table 1, it is defined as follows.

$$tf * rf = tf * log(2 + \frac{a}{mac(1,c)}) \tag{6}$$

The term weights of t1 to t5 for each category are shown in Table 3.

**Table 3.** The term weights of $t_1$ to $t_5$ for each category.

| *category* | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ |
|---|---|---|---|---|---|
| $C_1$ | 1.1155 | 1.1699 | 1.2538 | 1.3626 | 1.2538 |
| $C_2$ | 1.1155 | 1.1699 | 1.2538 | 1.3626 | 1.2538 |
| $C_3$ | 1.0000 | 1.1699 | 1.1375 | 1.0704 | 1.1375 |
| $C_4$ | 1.2630 | 1.4510 | 1.0875 | 1.1520 | 1.0875 |
| $C_5$ | 1.4594 | 1.0000 | 1.1375 | 1.0000 | 1.1375 |

In Younghoong Ko's methods, *D-TMax* selects the two largest sum values. For multi-class classification problems, in order to obtain better performance, can more categories be selected? Now we will select 1 to 5 categories to test this hypothesis. When choosing 1 or 2 categories, it is called "*D-*

*Max*" (Document Max) or "*D-TMax*" (Document Two Max) [9]. Based on this rule, we named 3, 4 and 5 categories as "*D-3Max*" (Document Three Max), "*D-4Max*" (Document Four Max) and "*D-5Max*" (Document Five Max). For multiclass text categorization, we named it "*D-NMax*" (Document Number Max) in this study. The number of selected categories and corresponding results are shown in the Table 4.

**Table 4.** The number of selected categories and corresponding results.

| *method* | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ |
|---|---|---|---|---|---|
| *D-Max* | 1.1155 | 1.1699 | 1.2538 | 1.3626 | 1.2538 |
| *D-TMax* | 1.1155 | 1.1699 | 1.2538 | 1.3626 | 1.2538 |
| *D-3Max* | 1.2630 | 1.4510 | 1.2538 | 1.3626 | 1.2538 |
| *D-4Max* | **1.4594** | 1.4510 | 1.2538 | 1.3626 | 1.2538 |
| *D-5Max* | **1.4594** | 1.4510 | 1.2538 | 1.3626 | 1.2538 |

According to table 2, the frequency of features in each category, $t_3$ and $t_5$ are evenly distributed in each category and do not have the ability to distinguish categories. According to the distribution of $t_1$, $t_2$ and $t_4$, these three terms are relatively distinguishable, especially $t_2$, which is very concentrated in $C_4$, and its term weight should be the highest among these five terms. Although $t_1$ appears centrally in $C_5$, it also appears in individual documents of categories $C_1$, $C_2$ and $C_4$. It has little ability to distinguish categories. For $t_4$, its distinguishing ability is not specific to a single category. In other words, it can only distinguish $C_1$ and $C_2$ from other categories, but cannot distinguish whether the document belongs to $C_1$ or $C_2$. The result in bold in the Table 4 violates our intuition that the weight of $t_1$, $t_2$ and $t_4$ should be large, and the weight of $t_1$ should be relatively small compared to $t_2$ and $t_4$. Since $t_1$ appear with a low frequency in documents compared to $t_2$ and $t_4$. Another unreasonable observation is that $t_1$ in some documents ($d_4$ and $d_9$) in category 1 and category 2 also have same frequency when compared to $t_1$ in the documents in category 5. After observation, we can find that the results of *D-Max*, *D-TMax* and *D-5Max* (*W-Max*) cannot boost the performance of text categorization. The results from *D-3Max* are consistent with our intuition that the weight of $t_1$, $t_2$ and $t_4$ should be large, and the weight of $t_1$ should be relatively small. In order to overcome the shortcomings of Younghoong Ko's methods, in this section we explain our proposed *DRGP* method, which will choose the traversal method to appropriately weigh the contribution of each term. The *DRGP* can select the appropriate "*N*" (in *D-NMax*) to enhance the performance of text categorization. While our previous work in [18] provided experimental evidence for the effectiveness of *DRGP* under particular experimental circumstances, we formally address how and why this new method is proposed using analytical explanation and empirical observation. Results in Section 4 show that the proposed strategy outperforms other methods significantly.

In the *DRGP* method, by traversing the term weighting vectors generated by each class, we compare their weighting effects on the training set. The term weighting vector which produces the best effect on training set will be selected as the term weighting vector of test set. The idea of the proposed method is mainly inspired by Younghoong Ko, Miao and Kamel [19]. They proposed a pairwise optimized Rocchio algorithm, which dynamically adjusts the prototype position between pairs of categories on training set, and record the best prototype position for test set. We summarize the main process of *DRGP* as Algorithm 1.

**Algorithm 1:** document representation strategy based on global policy (DRGP)

**Input:**

*fea*: feature matrix of training set

*gnd*: a vector of labels for documents in training set

**Output:**

*selectedC*: the most appropriate *N* value (in *D-NMax*) for current dataset

**Local variables**

$|C|$: total number of categories;

$M$: total number of features;

*termWeightingVec1*: the set of $|C|$ original term weighting vectors;

$termWeightingVec1_i$: $i$-th vector of the original term weighting vectors;

$termWeightingVec2_i$: $i$-th vector of the reconstructed term weighting vectors;

$sumVec_i$: sum value of all terms in $i$-th term weighting vector;

*sortSum*: sorted list of each sum values;

$weightedFea_i$: the weighted *fea* by using $termWeightingVec2_i$;

$MicroF1^i$: result of 10-fold cross validation on $weightedFea_i$;

**begin**

apply supervised term weighting method to *fea*, and get *termWeightingVec1*;
for $i = 1$ to $|C|$
    for $j = 1$ to $M$
        compute $sumVec_i$ for $termWeightingVec1_i$;
    end for
end for
sort all $sumVec_i$, and get *sortSum*;
for $i = 1$ to $|C|$
    for $j = 1$ to $M$
        for $k$=1 to $i$
            temp$C$ = *sortSum*[$k$]
            construct $termWeightingVec2_i$ by the following ways. The $j$-th dimension of each vector in the selected $k$ term weighting vectors is obtained, i.e., *termWeightingVec1*[temp$C$][$j$].
            the maximum value of all *termWeightingVec1*[temp$C$][$j$] will be selected as the $j$-th value of the $termWeightingVec2_i$;
        end for
    end for
end for
for $i = 1$ to $|C|$
    compute $weightedFea_i$;
end for
**for $i = 1$ to $|C|$**
    **compute $MicroF1^i$;**
**end for**
record $i$ corresponding to the maximum $MicroF1^i$, and assign it to *selectedC*;

**end**

After the algorithm 1 is executed, we can get *selectedC*, that is, the *N* value in *D-NMax*. Then we can use the following Eq (7) to calculate the term weighting vector of test set. The sum of all term weights in each vector $v_i$ is first calculated using Eq (4). Then the *N* largest sum values are selected, which are labeled as $sum_1$, $sum_2$ ...$sum_i$... and $sum_N$, respectively. The subscript 1, 2 ...*i*... and *N* are index of different categories. The *k-th* element of $v_d$ is calculated as Eq (7).

$$w_k = max(t_{1k}, t_{2k}, \ldots t_{ik} \ldots, t_{Nk}); i\epsilon[1, N], k\epsilon[1, m] \quad (7)$$

In algorithm 1, steps 1 to 6, first apply the term weighting method to the original data set to obtain |*C*| original term weighting vectors. Sum the original feature weight vectors separately to get $sumVec_i$. Steps 8 to 16, sort all $sumVec_i$ to get *sortSum*, and reconstruct |*C*| term weighting vectors by the following way. Select the first 1 to |*C*| vectors of *sortSum* in turn, and construct one term weighting vector each time. The *j*-th dimension element of the term weighting vector is selected in the following method: the maximum value of the *j*-th dimension of the currently selected *k* term weighting vectors. Steps 17 to 19, through the newly constructed |*C*| term weighting vector *termWeightingVec2*, weight the *fea* matrix respectively, and each $termWeightingVec2_i$ will get the corresponding $weightedFea_i$. Steps 20 to 22, verify each $weightedFea_i$ through ten-fold cross-validation, and record the corresponding result $MicroF_1^i$. Step 23, record the maximum $MicroF_1$ and the corresponding weight vector sequence number *i* as the result of *selectedC*. The $MicroF_1$ is the popular performance measure in text categorization and the formula will present in Section 4.

However, the algorithm has a shortcoming for its high time complexity. The most time-complex part is to calculate $MicroF_1^i$ (steps 19 to 21, in bold), which is calculated using ten-fold cross-validation. According to Lin's 2006 "machine learning summer school", "the time complexity of the support vector machine algorithm is $O(N^3)$". Therefore, the time complexity of this algorithm can be estimated as $O(|C|*N^3)$. Compared with some algorithms that use time in exchange for space, the *DRGP* algorithm uses time in exchange for precision. This algorithm is more suitable for classification systems that use a fixed training set, that is, after one training on the training set to find the optimal model, it can be applied to the classification system without changing the parameters for a long time.

## 4. Experimental results

### *4.1. Data corpora*

#### 4.1.1. The 20 Newsgroups corpus

The 20 Newsgroups corpus is a generally used benchmark dataset in the TC [20]. In the corpus, there are 20,000 newsgroup documents nearly uniformly distributed into 20 classes. In this study, we use the 20 Newsgroups sorted by the date. After removing duplicates and headers, the remaining 18,846 documents are partitioned into 11,314 (about 60 percent) training documents and 7532 (about 40 percent) testing documents. After preprocessing, there are 26,214 distinct words in this data set.

#### 4.1.2. The Reuters21578 corpus

The Reuters21578 corpus is used in many experiments and it contains 21,578 documents in 135 categories [21,22]. We use its ModApte version. There are 5946 training documents and 2347 testing documents in this version. In the study, we choose the top 10 largest categories which have 5228

training documents and 2057 testing documents. After preprocessing, the resulting vocabulary has 18,221 distinct words. Compared with 20 Newsgroups, it is a skewed data set and 80 percent of the categories have less than 7.5 percent instances.

The statistics of datasets are listed in Table 5.

**Table 5.** Statistics of datasets.

| *Datasets* | # of documents | distinct words | # of classes | # of training | # of testing |
|---|---|---|---|---|---|
| 20 Newsgroups | 18,846 | 26,214 | 20 | 11,314 | 7532 |
| Reuters21578 | 7285 | 18,221 | 10 | 5228 | 2057 |

### *4.2. Combined document representation strategies*

Some different document representation strategies listed in Table 6 are selected in this study. Besides *D-Max*, *D-TMax* and *W-Max*, we also investigate the *D-NMax* in the experiment. In the experiment, we list the results of the selected categories from 1 to the maximum. We can see the corresponding results when the number of selected categories is changed.

**Table 6.** Summary of document representation strategies.

| *Denoted by* | *Description* |
|---|---|
| *D-Max* | The sum of all term weights in each term weighting vector is first calculated and then one term weighting vector with the maximum sum value is selected as the document representation vector. |
| *D-TMax* | The sum of all term weights in each term weighting vector is calculated and then two term weighting vectors with the two largest sum values are selected. Then the term weighting vector is constructed by choosing the higher term weighting value from the selected two term weighting vectors for each corresponding dimension's term weight. |
| *D-NMax* | The sum of all term weights in each term weighting vector is calculated and then *N* term weighting vectors with the *N* largest sum values are selected. Then the term weighting vector is constructed by choosing the highest term weighting value from the selected *N* term weighting vectors for each corresponding dimension's term weight. |
| *W-Max* (*global policy*) | Each term's value of term weighting vector will be replaced by the maximum value of the corresponding dimension's term weight in all categories. |

### *4.3. Learning algorithms*

To evaluate classification performance of the proposed method, we choose the promising Support Vector Machines (*SVM*) learning algorithm in this study [23,24]. Although other algorithms such as *k* Nearest Neighbor, Decision Tree and Naive Bayes are also widely used, they are not included because the real number format of term weights could not be used except for the binary representation (see an

exception in [25]). Finally, the *SVM* learning algorithm scale to large classification problems with thousands of features and examples.

The *SVM* is a relatively efficient machine learning algorithm which shows a good performance. It is based on the structural risk minimization principle from computational learning theory [26]. It can handle the large-scale and high-dimensional data sets with high classification accuracy. According to different kernel functions, *SVMs* are divided into two categories: nonlinear (such as polynomial, sigmoid function, radial-based function) and linear methods. Leopold and Kindermann pointed out that term weighting schemes dominate the performance of *SVM* classifiers rather than the kernel functions [27]. Moreover, the literatures have proven that the linear *SVM* is superior to nonlinear *SVM* [28]. So, we select the linear kernel function in this study. The other parameters of SVM are set to their default values. The *SVM* software used is from *LIBSVM*-3.14 [29].

### *4.4. Performance evaluation*

The standard measures to determine the performance of a classification task are precision and recall [30,31]. However, it is well known that we may receive low recall when we obtain high precision. It will be ineffective if precision and recall are separated [1]. The widely used measure is $F_1$ measure which combines the precision and recall. For a given category *i*, Precision, Recall and $F_1$ measure are defined as follows.

$$Precision_i = TP_i/(TP_i + FP_i) \tag{8}$$

$$Recall_i = TP_i/(TP_i + FN_i) \tag{9}$$

$$F_1^i = (2 * Precision_i * Recall_i)/\ (Precision_i + Recall_i) \tag{10}$$

where $TP_i$ is the number of documents assigned correctly to class *i*, $FP_i$ is the number of documents that do not belong to class *i* but are assigned to class *i*. $FN_i$ is the number of documents that actually belong to class *i* but are not assigned to the class *i*.

The $F_1$ measure is estimated by $MicroF_1$ and $MacroF_1$ [32]. In this study, $MicroF_1$ and $MacroF_1$ are employed to measure the performance of the proposed method. They are computed as in Eqs (11) and (12).

$$MicroF_1 = 2 * Precision * Recall \tag{11}$$

$$MacroF_1 = \frac{1}{m} * \sum_{i=1}^{m} F_1^i \tag{12}$$

In Eq (10), *m* is the number of categories. The $MicroF_1$ assigns equal weight to each document and it is considered as an average over all the document/category pairs. The $MacroF_1$ assigns equal weight to each category and is influenced by the results of rare categories.

In order to verify the effectiveness of the proposed method in the experiment, besides *tf*rf* mentioned before, we also include the *tf*or* which is widely used in the previous study [33]. Its formula is expressed as

$$tf * or = tf * \log\left(\frac{a*d}{b*c}\right) \tag{13}$$

*4.5. Experiments*

The main purpose of the experiments is to address the two questions, i.e., to explore the superiority of document representation strategy for supervised term weighting schemes and find a measure before choosing a document representation strategy. To accomplish this, we compare the methods on two popular benchmark data corpora, i.e., 20 Newsgroups and Reuters-21578, using *SVM* in terms of $MicroF_1$ and $MacroF_1$ measure. In addition to showing the corresponding results of *DRGP* in the experiment, the corresponding results of selecting other number of categories are also listed. The experimental results and discussion on these two corpora are in Sections 4.5.1–4.5.3.

4.5.1. Results and discussion on the 20 newsgroups corpora

The returned value of *selectedC* is 20 when *DRGP* is used on 20 Newsgroups which are weighted by *tf*or* term weighting method. Then we can use the formula (7) to calculate the term weighting vector of test set. Figure 1 shows the results of $MicroF_1$ and $MacroF_1$ when using supervised term weighting method *tf*or* and *SVM* classification algorithm with linear kernel functions on 20 Newsgroups. The best $MicroF_1$ (0.7884) and $MacroF_1$ (0.7837) are achieved by using *DRGP* and *W-Max*. The result obtained by *D-Max* is 0.5539 ($MicroF_1$) and 0.5521 ($MacroF_1$), and the result obtained by *D-TMax* is 0.6036 ($MicroF_1$) and 0.6027 ($MacroF_1$). It is more intuitive that the results and trends of $MicroF_1$ and $MacroF_1$ on the 20 newsgroups are basically the same. The main reason is that 20 Newsgroups is a balanced dataset.

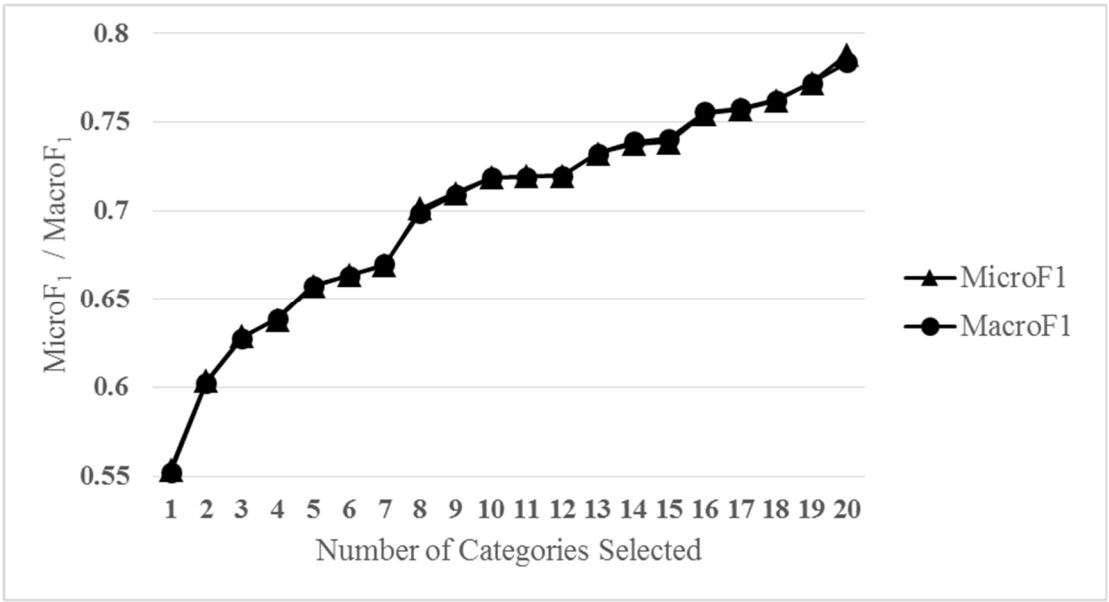


**Figure 1.** A comparison on $MicroF_1$ and $MacroF_1$ using *DRGP*, *tf*or* and *SVM.*

The returned value of *selectedC* is 20 when *DRGP* is used on 20 Newsgroups which are weighted by *tf*rf* term weighting method. Then we can use the formula (7) to calculate the term weighting vector of test set. Figure 2 shows the results of $MicroF_1$ and $MacroF_1$ when using supervised term weighting method *tf*rf* and *SVM* classification algorithm with linear kernel functions on 20 Newsgroups. The best $MicroF_1$ (0.7958) and $MacroF_1$ (0.7909) are achieved by using *DRGP* and *W-Max*. The result

obtained by *D-Max* is 0.7562 ($MicroF_1$) and 0.7502 ($MacroF_1$), and the result obtained by *D-TMax* is 0.7592 ($MicroF_1$) and 0.7529 ($MacroF_1$).

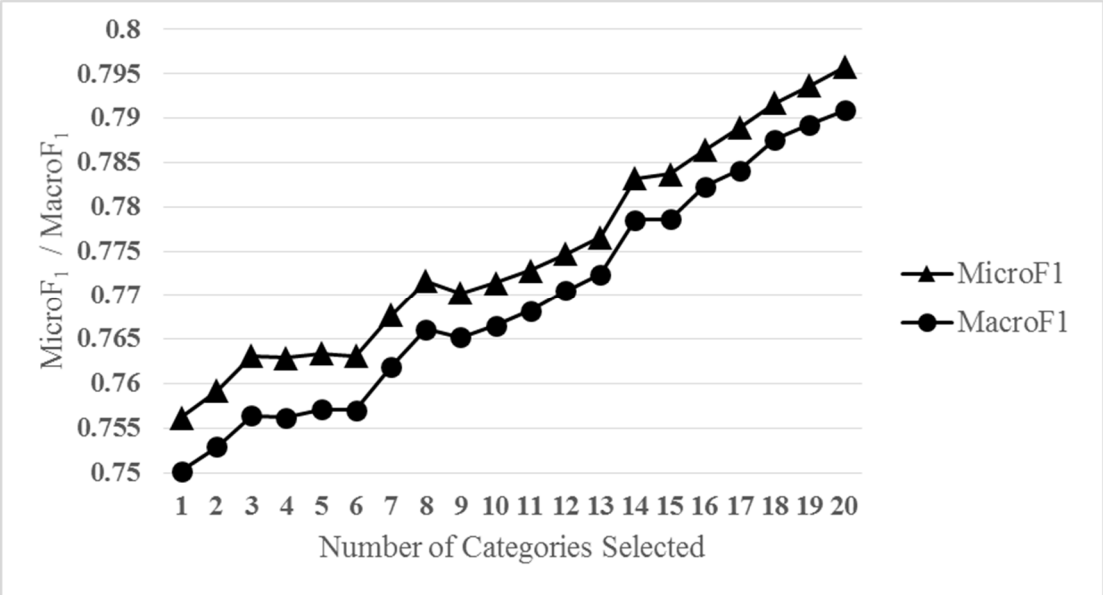


**Figure 2.** A comparison on $MicroF_1$ and $MacroF_1$ using *DRGP*, *tf*rf* and SVM.

4.5.2. Results and discussion on reuters-21578 corpora

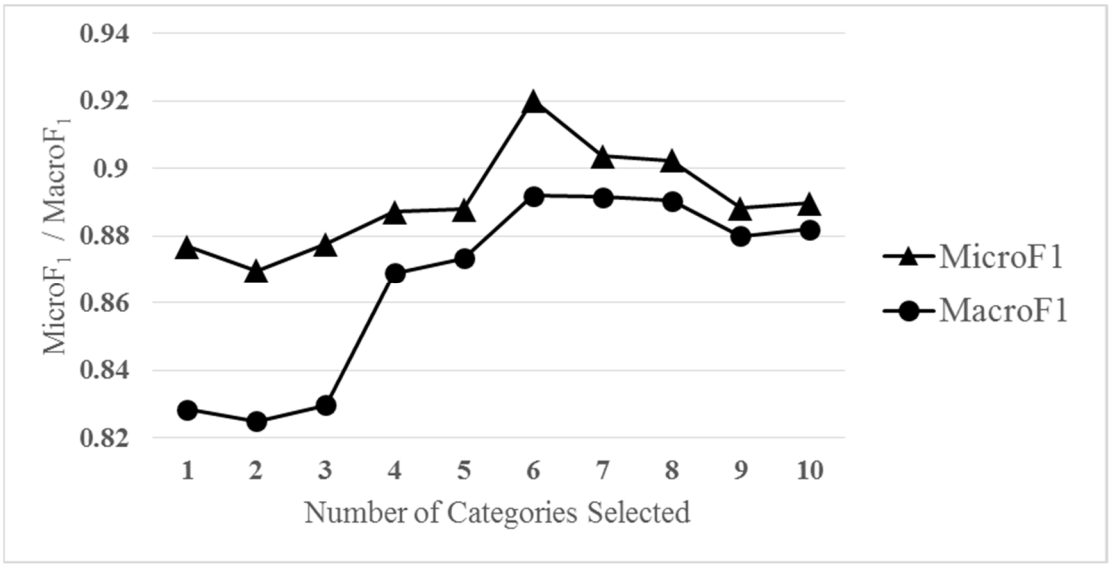


**Figure 3.** A comparison on $MicroF_1$ and $MacroF_1$ using *DRGP*, *tf*or* and *SVM.*

The returned value of *selectedC* is 6 when *DRGP* is used on Reuters-21578 which are weighted by *tf*or* term weighting method. Then we can use the Eq (7) to calculate the term weighting vector of test set. Figure 3 shows the results of $MicroF_1$ and $MacroF_1$ when using supervised term weighting method *tf*or* and *SVM* classification algorithm with linear kernel functions on Reuters-21578. In other

words, the most appropriate *N* value for current dataset is 6. The sum of all term weights in each term weighting vector is calculated and then six term weighting vectors corresponding to the first six largest sum values are selected. Then the term weighting vector is constructed by choosing the highest term weighting value from the selected six term weighting vectors for each corresponding dimension's term weight. The best $MicroF_1$ (0.9203) and $MacroF_1$ (0.8919) are achieved by using *DRGP*. The result obtained by *D-Max* is 0.8769 ($MicroF_1$) and 0.8284 ($MacroF_1$), and the result obtained by *D-TMax* is 0.8697 ($MicroF_1$) and 0.825 ($MacroF_1$), and the result obtained by *W-Max* is 0.8898 ($MicroF_1$) and 0.882 ($MacroF_1$).

The returned value of *selectedC* is 6 when *DRGP* is used on Reuters-21578 which are weighted by *tf*rf* term weighting method. Then we can use the Eq (7) to calculate the term weighting vector of test set. Figure 4 shows the results of $MicroF_1$ and $MacroF_1$ when using supervised term weighting method *tf*rf* and *SVM* classification algorithm with linear kernel functions on Reuters-21578. The best $MicroF_1$ (0.9283) and $MacroF_1$ (0.9086) are achieved by using *DRGP*. The result obtained by *D-Max* is 0.9258 ($MicroF_1$) and 0.9077 ($MacroF_1$), and the result obtained by *D-TMax* is 0.9258 ($MicroF_1$) and 0.8966 ($MacroF_1$), and the result obtained by *W-Max* is 0.9269 ($MicroF_1$) and 0.9049 ($MacroF_1$).

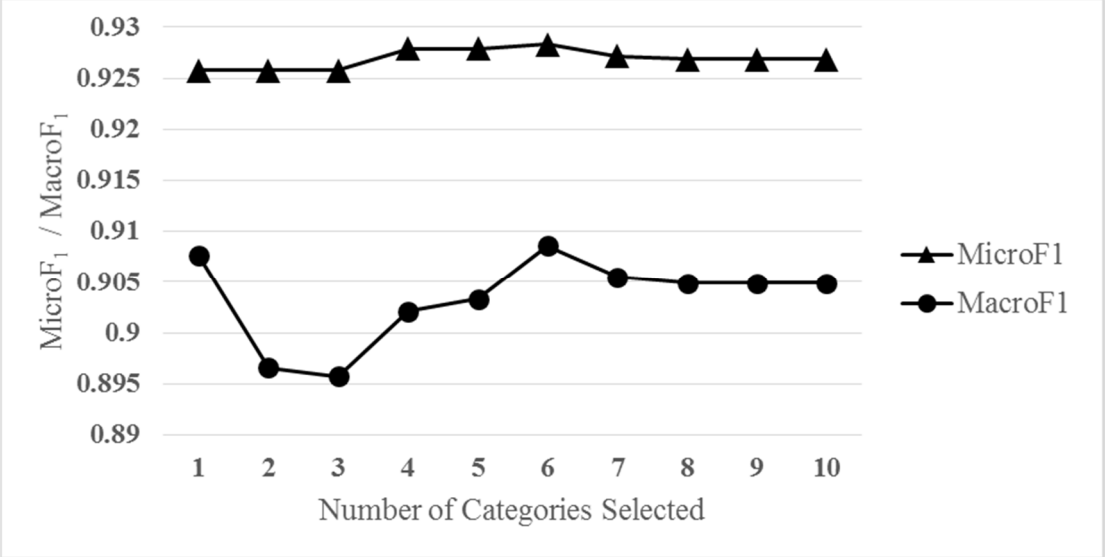


**Figure 4.** A comparison on $MicroF_1$ and $MacroF_1$ using *DRGP*, *tf*rf* and *SVM*.

4.5.3. Discussion on the effects of corpora on different algorithms

The above results showed that *DRGP* and *W-Max* achieved the best performance on 20 Newsgroups while *DRGP* achieved the best performance on Reuters-21578. In addition, the results and trends of $MicroF_1$ and $MacroF_1$ on 20 Newsgroups are basically consistent. We think it is very natural results. Because 20 Newsgroups is a balanced dataset and Reuters-21578 is a skewed dataset. Many documents in Reuters-21578 have two or more labels. For example, if a document with two labels is represented by using only one class distribution between two labels, classifiers could has some difficulty to classify it in the other class.

In skewed data sets, it is not suitable to adopt the document representation strategy of *D-Max*, *D-TMax*, and *W-Max* (global policy). Note that although our experiments use the same corpus and same evaluation measure as Youngjoong Ko's [8], there are minor differences in data preparation such as

the stemming or stop words lists for text preprocessing and the different feature selection measures. On Reuters-21578, the experimental results show the same phenomenon, that the results obtained by *D-TMax* are greater than *D-Max*, and the results obtained by *W-Max* method are the worst.

In the experimental results, we can observe that on the balanced data set (20 Newsgroups), $MicroF_1$ and $MacroF_1$ will increase with the increase of selection categories. However, the result of skewed data set (Reuters-21578) is that $MicroF_1$ and $MacroF_1$ first increase and then decrease with the increase of selection categories. Therefore, in order to obtain better results, it is necessary to select the appropriate method according to the characteristics of the data set. The proposed *DRGP* method has achieved best performance in both balanced and skewed data sets. Especially on skewed dataset, the results achieved are greatly improved compared to the results achieved by traditional methods. It should be noted that here we only use *tf*or* and *tf*rf* weighting methods and *SVM* classification method. Our method can be extended to other supervised term weighting methods, and other classifiers can be used for cross validation or classification.

## 5. Conclusions

Document representation is one of the most important parts for constructing a text classifier. For the supervised term weighting methods, in addition to the global policy, there is local policy, including *D-Max*, *D-TMax* and *W-Max*, etc. Faced with so many strategies, we are not sure how to choose to achieve the best results. That is, the *N* value in *D-NMax* cannot be determined.

In this study, we studied and solved this problem. We found that representation methods should be chosen according to corpora characteristics to have better classification performance. Compared with other methods, such as *D-Max*, *D-TMax* and *W-Max*, the method proposed in this study is not only a weighting method, but also a selection strategy. The method can select the appropriate *N* (in *D-NMax*) value according to distribution of each category in the dataset. By testing the proposed method on two representative supervised term weighting methods (*tf*rf* and *tf*or*) on two datasets (20 Newsgroups and Reuters-21578), it can be obtained that the method is effective on both balanced and unbalanced datasets.

Based on the original document representation method, the proposed method introduces the idea of traversal. Through cross validation, the proposed method first finds the optimal document representation model on the training set, and then applies the selected model to the test set. Due to the different working mechanism of different classifiers, different classifiers may have an impact on the final results, but the method proposed in this paper has no special requirements for classifiers. In this study, we use the SVM. This method can be extended to other supervised feature weighting methods, and other classifiers can be used for cross validation or classification. Compared with the original document representation methods, the proposed method increases the additional computational cost (cross validation on the test set). In the next research, we will continue to study and introduce new optimization methods to reduce the calculation cost.

At the end of this study, we answer the questions we raised in the beginning of the paper as conclusions:

1) Is there a relatively general strategy for supervised term weighting schemes, and, if yes, which one can achieve better performance?

Through a series of evaluations in text categorization, including balanced dataset and skewed dataset, we find that the performances of *W-Max*, *D-Max*, *D-TMax* get different results on different types of datasets. The proposed *DRGP* and *W-Max* achieved the best performance on 20 Newsgroups.

Besides, the proposed *DRGP* achieved the best performance on Reuters-21578.

2) How can we know the effect of document representation method before we choose it? In other words, when selecting a document representation strategy for an unknown data set, which method is the best choice?

In this study, we propose the *DRGP* method, which can select appropriate representation strategy for an unknown dataset. No matter the data set is uniform or not, it will use traversal method on the constructed optimization function to find the document representation strategy on training set, then apply it on test set. The *DRGP* exhibit stable and consistent improvement over most of the previous document representations mentioned in the experiments.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61977015, and the Jilin Province Development and Reform Commission Project of China under Grant 2020C017-3. We would like to thank the organizations for their supports.

## Conflict of interest

All authors declare no conflicts of interest in this paper.

## References

1. M. Lan, S. Sung, H. Low, C. Tan, A comparative study on term weighting schemes for text categorization, in *Proceedings 2005 IEEE International Joint Conference on Neural Networks*, **1** (2005), 546–551. https://doi.org/10.1109/IJCNN.2005.1555890
2. X. Li, A. Zhang, C. Li, J. Ouyang, Y. Cai, Exploring coherent topics by topic modeling with term weighting, *Inf. Process. Manage.*, **54** (2018), 1345–1358. https://doi.org/10.1016/j.ipm.2018.05.009
3. M. Lan, C. Tan, J. Su, Y. Lu, Supervised and traditional term weighting methods for automatic text categorization, *IEEE Trans. Pattern Anal. Mach. Intell.*, **31** (2008), 721–735. https://doi.org/10.1109/TPAMI.2008.110
4. E. H. Han, G. Karypis, V. Kumar, Text Categorization Using Weight Adjusted K-Nearest Neighbor Classification, *Proc. Pacific Asia Conf. Knowl. Discovery Data Min.*, (2001), 53–65. https://doi.org/10.1007/3-540-45357-1_9
5. X. Quan, W. Liu, B. Qiu, Term weighting schemes for question categorization, *IEEE Trans. Pattern Anal. Mach. Intell.*, **33** (2010), 1009–1021. https://doi.org/10.1109/TPAMI.2010.154
6. A. I. Kadhim, Survey on supervised machine learning techniques for automatic text classification, *Artif. Intell. Rev.*, **51** (2019), 273–292. https://doi.org/10.1007/s10462-018-09677-1
7. M. M. Michał, J. Protasiewicz, A recent overview of the state-of-the-art elements of text classification, *Expert Syst. Appl.*, **106** (2018), 36–54. https://doi.org/10.1016/j.eswa.2018.03.058
8. C. Liu, Y. Sheng, Z. Wei, Y. Yang, Research of text classification based on improved TF-IDF algorithm, in *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, 2018. https://doi.org/10.1109/IRCE.2018.8492945

9. Y. Ko, A study of term weighting schemes using class information for text classification, in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012. https://doi.org/ 10.1145/2348283.2348453
10. M. Yurochkin, S. Claici, E. Chien, F. Mirzazadeh, J. Solomon, Hierarchical optimal transport for document representation, preprint, arXiv:abs/1906.10827.
11. W. Zhang, Y. Li, S. Wang, Learning document representation via topic-enhanced LSTM model, *Knowl. Based Syst.*, **174** (2019), 194–204. https://doi.org/10.1016/J.KNOSYS.2019.03.007
12. L. Li, B. Qin, W. Ren, T. Liu, Document representation and feature combination for deceptive spam review detection, *Neurocomputing*, **254** (2017), 33–41. https://doi.org/ 10.1016/j.neucom.2016.10.080
13. S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis, *J. Am. Soc. Inf. Sci.*, **41** (1990), 391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391:: AID-ASI1>3.0.CO;2-9
14. D. M. Blei, A. Ng, M. I. Jordan, Latent dirichlet allocation, *J. Mach. Learn. Res.*, **3** (2003), 993–1022. https://doi.org/10.1016/B978-0-12-411519-4.00006-9
15. T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *Comput. Sci.*, 2013. https://doi.org/10.48550/arXiv.1301.3781
16. Q. V. Le, T. Mikolov, Distributed representations of sentences and documents, *Int. Conf. Mach. Learn. PMLR*, 2014.
17. F. Sebastiani, Machine learning in automated text categorization, *ACM Comput. Surv. (CSUR)*, **34** (2002), 1–47. https://doi.org/10.1145/505282.505283
18. L. Jia, B. Zhang, Optimal document representation strategy for supervised term weighting schemes in automatic text categorization, in *2019 9th International Conference on Information and Social Science*, 2019.
19. Y. Q. Miao, M. Kamel, Pairwise optimized Rocchio algorithm for text categorization, *Pattern Recogn. Lett.*, **32** (2011), 375–382. https://doi.org/10.1016/j.patrec.2010.09.018
20. C. Deng, X. He, Manifold adaptive experimental design for text categorization, *IEEE Trans. Knowl. Data Eng.*, **24** (2011), 707–719. https://doi.org/10.1109/TKDE.2011.104
21. L. Man, C. L. Tan, H. B. Low, Proposing a new term weighting scheme for text categorization, *AAAI*, **6** (2006).
22. M. Revanasiddappa, B. Harish, A new feature selection method based on intuitionistic fuzzy entropy to categorize text documents, *Int. J. Interact. Multim. Artif. Intell*, **5** (2018), 106–117. https://doi.org/ 10.9781/ijimai.2018.04.002
23. M. Goudjil, M. Koudil, M. Bedda, N. Ghoggali, A novel active learning method using SVM for text classification, *Int. J. Autom. Comput.*, **15** (2018), 290–298. https://doi.org/ 10.1007/S11633-015-0912-Z
24. M. Haddoud, A. Mokhtari, T. Lecroq, Saïd Abdeddaïm Combining supervised term-weighting metrics for SVM text classification with extended term representation, *Knowl. Inf. Syst.*, **49** (2016), 909–931. https://doi.org/10.1007/s10115-016-0924-1
25. A. McCallum, K. Nigam, A comparison of event models for naive bayes text classification, in *Proceeding AAAI Workshop Learning for Text Categorization*, 1998.
26. Y. Yang, An evaluation of statistical approaches to text categorization, *Inf. Retr.*, **1** (2004), 69–90. https://doi.org/ 10.1023/A:1009982220290

27. E. Leopold, J. Kindermann, Text categorization with support vector machines. How to represent texts in input space, *Mach. Learn.*, **46** (2002), 423–444. https://doi.org/ 10.1023/A:1012491419635
28. S. Lee, K. Seo, Intelligent fault diagnosis based on a hybrid multi-class support vector machines and case-based reasoning approach, *J. Comput. Theor. Nanosci.*, **10** (2013), 1727–1734. https://doi.org/ 10.1166/JCTN.2013.3116
29. C. C. Chang, C. J. Lin, LIBSVM: a library for support vector machines, *ACM Trans. Intell. Syst. Technol. (TIST)*, **2** (2011), 27:1–27:27. https://doi.org/ 10.1145/1961189.1961199
30. J. Zhang, L. Chen, G. Guo, Projected-prototype based classifier for text categorization, *Knowl.-Based Syst.*, **49** (2013), 179–189. https://doi.org/ 10.1016/j.knosys.2013.05.013
31. F. Ren, M. G. Sohrab, Class-indexing-based term weighting for automatic text classification, *Inf. Sci.*, **236** (2013), 109–125. https://doi.org/10.1016/j.ins.2013.02.029
32. I. Alsmadi, G. K. Hoon, Term weighting scheme for short-text classification: Twitter corpuses, *Neural Comput. Appl.*, **31** (2019), 3819–3831. https://doi.org/10.1007/s00521-017-3298-8
33. Y. Ko, New feature weighting approaches for speech-act classification, *Pattern Recogn. Lett.*, **51** (2015), 107–111. https://doi.org/10.1016/j.patrec.2014.08.014


AIMS Press