International Journal of Database Theory and Application
Vol.9, No.4 (2016), pp.1-22
http://dx.doi.org/10.14257/ijdta.2016.9.4.01

Text Representation Based on Key Terms of Document for Text Categorization

Jieming Yang *, Zhiying Liu and Zhaoyang Qu

College of Information Engineering, Northeast Dianli University, Jilin, Jilin, China
yjmlzy@gmail.com

Abstract

The text representation, "bag of words" or vector space model, is widely used by most of the classifiers in text categorization. All the documents fed into the classifier are represented as a vector in the vector space, which consists of all the terms extracted from training set. Due to the characteristics of high dimensionality, feature selection algorithm is usually used to reduce the dimensionality of the vector space. Through feature selection, each document is represented by some representative terms extracted from the training set. Although the classification results based on this document representation methodare better, it is inevitable that some documents may contain few even none representative terms, and these documents must be misclassified. In this paper, we proposed a new text representation method, KT-of-DOC, which represents one document using some key terms extracted from this document. We selected key terms of each document based on six feature selection algorithms, Improved Gini Index (GINI), Information Gain (IG), Mutual Information (MI), Odds Ratio (OR), Ambiguity Measure (AM) and DIA association factor (DIA), respectively, and evaluated the performance of two classifiers, Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), on three benchmark collections, 20-Newsgroups, Reuters-21578 and WebKB. The results show that the proposed representation method can significantly improve the performance of classifier.

Keywords: *text representation; feature selection; key term; text categorization*

1. Introduction

The amount of information available in digital form has been increasing exponentially due to the development of the information technology. The feasibility of manual classification decreases as the number of documents increases over time. As a result, the automatic document processing, such as text categorization, information retrieval, natural language processing, has become the hotspot and key technique to which most of the researchers pay attention. The text categorization has been studied by many researchers [1-4], which assigns one or more predefined categories to a new document based on its contents [5]. So far, there exist many algorithms applied for the text categorization, such as Naïve Bayes method (NB) [6], Support Vector Machines (SVM) [7], K-Nearest Neighbors (KNN) [8], decision trees, *etc.* It should be pointed out that the text information should be preprocessed and converted to a general form that fits to one specific classification algorithm. Most of the algorithms are based on the same text representation, "bag of words", also known as vector space model, which consists of the unique terms (words or phrases) [3] extracted from the training set. A document is represented as a point of the vector space according to the terms appearing in it. The score assigned to each term usually expresses whether the term appears in a document or how

* Corresponding Author

ISSN: 2005-4270 IJDTA

frequently the term appears [9]. There exist two characteristics about text categorization. One is that the number of the terms in the vector space model can easily reach orders of tens of thousands even for moderate size data sets [5]. The other is that the sparsity of document is very high. These characteristics reduce the performance of the text classification; even some sophisticated algorithms cannot be applied for text classification.

Therefore, dimensionality reduction, aims to reduce the size of the vector space without losing the performance of the classifiers [4], has become the focus of text categorization. Feature-selection is one of dimensionality reduction methods to which most researchers pay attention. The terms occurring in documents collection are ranked according to statistics or information theory, and then the top *k* terms are selected to construct the new vector space. There exist many feature selections, such as Information Gain (IG) [3, 10], Chi-square statistics [3, 10], Expected Cross Entropy [11], improved Gini Index [10, 12-13], Mutual Information [3, 14], Odds Ratio [12, 15], Ambiguity Measure(AM) [12], Darmstadt Indexing Approach association factor(DIA) [4-5, 16], Bi-Test [17], *etc*. Yang and Pedersen [3] indicated that most of feature selection algorithms can reduce the dimensionality of the vector space by a factor of 100 without losing categorization accuracy. The feature selection approaches mentioned above can be grouped into two opposed categories, one is global feature selection approach, and the other is local feature selection approach [4]. The global feature selection approach selects key terms from entire training set, meanwhile, the local feature selection approach selects key terms from the category. Although several local feature selection approaches have been proposed [18-19], the global approach is often considered in the feature selection [20]. Since the new vector space generated by the global feature selection approaches consisted of key terms of entire training set, the representation method of documents using this reduced vector space is denoted by KT-of-TR in this paper. In this context, KT-of-TR has a potential defect. When a document is represented as a vector in new vector space, most or even all terms of some documents will not appear in the reduced vector space. So the value of most or even all of terms in the document vector is zero. Such documents must be misclassified when new reduced vector space is used. Thus the performance of the classifier is association with the representation of the individual document [21].

In this paper, we proposed a new text representation algorithm, named KT-of-DOC. The proposed method selects *k* most informative (key) terms from each document in the corpus based on a feature selection algorithm, and then these terms compose a new vector space. So it can be guaranteed that any of the documents in corpus has at least *k* terms appeared in the vector space; meanwhile, the dimensionality of the vector space is reduced. In this paper, we firstly select key terms from all documents based on six feature selection algorithms, improved Gini index (Gini), Information Gain (IG),Mutual Information (MI), Odds Ratio (OR), Ambiguity Measure (AM) and DIA association factor (DIA), respectively, and then evaluate our algorithm on three benchmark document collections, 20-Newsgroups, Reuters-21578 and WebKB, using two classification algorithms, Support Vector Machines (SVM) and K-Nearest Neighbors (KNN). The experiment results show that the performance of the classifier has been greatly improved when the new text representation method (KT-of-DOC) instead of the traditional text representation method (KT-of-TR) is used. So the representation style of text is very important to improve the performance of the classifier.

The rest of this paper is organized as follows: Section 2 discusses the related work. Section 3 presents the motivation and theoretical foundation of the proposed algorithm. Section 4 presents the experimental setup and the datasets, classifiers and evaluation measures we used. Section 5 describes the computational efforts in the experiments. Discussions are shown in Section 6 and Conclusions are given in Section 7.

2. Related Work

The “bag of words” (vector space model) is commonly used by most of the classifiers, denoted by $D = \{t_1, t_2, ...t_n\}$, n is the number of the terms in vector space, and may be an orders of magnitude of tens of thousands. The element of the vector space model consists of terms (words, phrases or n-grams) which are extracted from the training set. Each document in corpus is represented as a vector according to the vector space model, $d_i = \{t_{1i}, t_{2i}, ...t_{ji}, ...t_{ni}\}$, where $1 \leq j \leq n$. The value of an element $t_{ji}$is assigned according to three methods: 1. binary method,$t_{ji} = 1$, if t_j occurs in the document d_i, otherwise $t_{ji} = 0$; 2. term frequency (tf), $t_{ji}$is assigned the frequency of the term t_j that occurs in the document d_i; 3. the product of term frequency and inverse document frequency(tf×idf), $t_{ji}$is assigned the value of term frequency multiplied by inverse document frequency which is the number of documents in training set that $t_j$occurs.

So far, there exist many sophisticated text representations, which are proposed and evaluated. These text representations focus on the generation of the “”bag of words”. For example, an unique word, the phrases (statistics phrases or syntactic phrases) or n-grams [22] can be considered as a term. Bekkerman,*et al*. [23] adopted the distributed term-clustering based on the information bottleneck method [24], and a document was represented as a vector of word cluster counts corresponding to a cluster mapping(from words to cluster centroids). In this way, not only was the dimensionality of the vector space reduced, but also the relationships of the features were retained. In addition, this method generated extremely compact representations. L. Chen, Zeng and Tokuda [25] proposed the concept of the stereo document representation, they considered that it is not necessary to read entire content of a document in order to assign a class label to it, and only one part of a document may be enough. Therefore, the stereo document representation consists of the information, which is extracted from a document in different ways. It can be said that a document is represented by the different perspectives of it. Xiao-Bing and Zhi-Hua [9] utilized the distributional features to improve the performance of classifier. They believed that the appearance frequency and the distribution of a term in a document are critical for reflecting the theme of a document. Graham [26] discussed the use of Bayesian analysis in spam filtering. The terms of an email were ranked based on the probability, and then the top fifteen terms were selected to represent the email. The text representation method achieved significant improvement in spam filtering. Mladenić, *et al*. [21] investigated the method of the feature selection approach combined with various learning models. The average number of nonzero components in the vector by which documents are represented is used to control the sparsity of the document representation. In their experiment, different sparsity level sare achieved by retaining a number of features with higher score according to the feature selection approach. They concluded that the learning algorithms are more sensitive to sparsity rather than the number of the features, and the sparsity of vectors representing the document was useful for comparing the different feature selection methods. Malik and Kender [27-28] adopted a three-step heuristic feature selection method to ensure that every document in training set is properly covered by the selected features: firstly, determining the number of the selected features (n) according to the number of the training documents and the size of the available features; secondly, selecting n features for the new vector space according to information gain; finally, checking whether or not every document is covered by at least k selected features. If not, the features in the document are ranked in descending order based on *TF**information gain, and then the top k features are added to the new vector space.

The literatures mentioned above show that the theme of a document determinates which category it falls into, and can be supported by only some representative terms instead of entire content of this document. In this paper, we will discuss the issue from two aspects: (1) how to represent a document by the representative terms, (2) how to use the new text representation on various types of classifiers.

3. Algorithm Description

3.1. Problem and Motivation

The curse of dimensionality caused by the “bag of words” (vector space model) is a primary obstacle for text categorization. So feature selection is used to reduce the dimensionality of the vector space model. The significance for categorization of each term that occurs in training set is calculated by various of feature selection algorithms and all terms in the training set are ranked according to the significance of the term. Finally, the reduced vector space consists of the top *k* terms selected from the terms list. Although the dimensionality and sparsity of the document representation has been reduced, another problem has arisen. Because the significance of a term is calculated and ranked based on all terms that occur in the training set. However, only few terms or even non-terms occur in the reduced vector space for some documents. For example, a document has 10 terms. In original vector space, the document is denoted by d_i = {0,1,0,0,1,1,0,1,1,1,1,0,0,1,1,1,0,…,0}, where 1 indicates that the term occurs, and 0 indicates the term does not occur. The dimensionality of $d_i$is equal to the dimensionality of the original vector space. After the dimensionality of the original vector space was reduced to 50, the document may be denoted by d_i= {0,0,0,0,0,0,...,0}, the dimensionality of d_i is equal to 50. It can be seen that the document d_i cannot be represented in the reduced vector space. Therefore, the document will be misclassified.

Human can grasp the topic of a document by glancing at the document and capturing its keywords, instead of using all the words in the document [12]. Namely, a person can decide the category of the document only according to a few key words. So the document represented by key terms can be correctly classified in text categorization.

In Graham [26], an email is represented by only fifteen terms whose probability is the highest. The method of Graham only fits to the Naïve Bayes classifier, and cannot be applied in model-based and instance-based classifiers, such as Support Vector Machines, K-Nearest Neighbors and Rocchio [4]. Mladenić, *et al.* [21] represented a document with certain number of nonzero terms of the document selected by the feature selection, and the other elements of the document vector is replaced by zero. In this paper, we firstly select *k* key terms from every document, and then merge them together as a new vector space model. Finally, all documents in the corpus are re-represented according to the new vector space model. The new vector space model has three advantages: (1) lower dimensionality, (2) lower sparsity, (3) each document can be represented by at least *k* key terms.

3.2. Algorithm Implement

After the document in the corpus is represented under the proposed vector space, two cases may occur with the new document vector. First, the document vector only contains these *k* key terms of the document, and the other elements of the document vector are zero. Second, the document vector contains these *k* key terms of the document and other terms that are not key terms of the document. In fact, the second situation is common. In order to emphasize the effect of the key terms of a document, we use two parametersC_1, C_2 and weighted on the key terms and non-key terms of the document in our algorithm, respectively. The parameter C_1 enhances the effect of the key terms of the document, so $C_1 \geq 1$. The parameter C_2 weaken the effect of the non-key terms of the document, so $0 \leq C_2 \leq 1$.The proposed algorithm includes two stages: the first stage is to generate new vector space; the second stage is to represent the document in the corpus as the new document vector.The pseudo code of the algorithm is detailed as follows.

Algorithm 1:

Input:

K: the number of the key terms that are extracted from each document;

V: the vocabulary which consists of the distinct terms occurring in the training set and ranked utilizing the feature selection algorithm, $V = \{v_1, v_2, \ldots, v_n, v_{n+1}, \ldots\}$, score($v_n$)>score($v_{n+1}$), $1 \leqslant n < |V|$, where $|V|$ is the size of the vocabulary, the score(v_n) is the significance of the term v_n based on one feature selection algorithm.

D: the training set, d_i is the ith document in the training set, $1 \leqslant i \leqslant N$, where N is the total number of documents in the training set.

C_1: the weight of the key term, $C_1 \geqslant 1$

C_2: the weight of the non-key term, $0 \leqslant C_2 \leqslant 1$

Output:

Q: new document set in which each document only retain the key terms; q_i is the ith document in new document set.

W: new vector space model which only consists of the key terms of each document in training set.

S: new training set in which each document is re-represented by new vector space model; $s_i$is the ith document in the new training set, $1 \leqslant i \leqslant N$, where N is the amount of the documents in the training set.

```
Step 1:   for each document $d_i$ in the training set $D$
Step 2:   $m = 0$;
Step 3:     for each term $v_n$ in $V$
Step 4:         for each term $d_{ji}$ in document $d_i$
Step 5:             if $v_n$and$d_{ji}$ are the same terms
Step 6:                 $m = m+1$;
Step 7:                 add $v_n$ to $W$
Step 8:                 add $d_{ji}$ to $q_i$
Step 9:                 goto Step 12
Step 10             end if
Step 11:        end for
Step 12:        if $m == K$
Step 13:            add $q_i$ to $Q$
Step 14:            goto step 1
Step 15:        end if
Step 16:    end for
Step 17: end for
Step 18: for each document $d_i$ in the training set $D$ and $q_i$ in new document set $Q$
Step 19:    for each term$w_j$ in new vector space $W$
Step 20:      if $w_j$ occurs in $d_i$(suppose the $k$th term $d_{ki}$ in $d_i$ is the same term as $w_j$)
Step 21:            if $w_j$ occurs in $q_i$
Step 22:            $s_{ji} = C_1*M(d_{ki})$// $M(d_{ki})$ is the weight value of term $d_{ki}$ in $d_i$ (tf or tf×idf)
Step 23:            else
Step 24:                $s_{ji} = C_2*M(d_{ki})$
Step 25:            end if
Step 26:      else
Step 27:            $s_{ji} = 0$
Step 28:      end if
Step 29:      add $s_{ji}$ to $s_i$
Step 30:    end for
Step 31:    add $s_i$ to $S$
Step 32: end for
```

3.3 Complexity Analysis

The variables used in the following complexity analysis are

V---The vocabulary which consists of the distinct terms occurring in the training set and ranked utilizing the feature selection algorithm.

$|V|$ --- The number of terms in the vocabulary V.

K --- The number of the key terms extracted from every document in training set.

U --- The number of terms in the new vector space.

T_r --- The number of the training documents.

T_e ---The number of the test documents.

L_r ---The average length of the training documents.

L_e ---The average length of the test documents.

The proposed algorithm is divided into two stages: the new vector space is generated in first stage and the documents in training set are re-represented according to the new vector space in the second stage.

- The time complexity of generating new vector space is $O(T_r{\cdot}L_r{\cdot}|V|{\cdot}K)$.
- The re-representation of documents in training set costs $O(T_r{\cdot}L_r{\cdot}U)$.
- The time complexity of extraction of key terms of test documents is $O(T_e{\cdot}L_e{\cdot}|V|{\cdot}K)$.
- The re-representation of test documents costs $O(T_e{\cdot}L_e{\cdot}U)$

The total time complexity of the proposed text representation is $O((T_r{\cdot}L_r{+}T_e{\cdot}L_e){\cdot}|V|{\cdot}K$ $+(T_r{\cdot}L_r{+}T_e{\cdot}L_e){\cdot}U)$. In the traditional text representation, the time complexity of generating the new vector space is $O(1)$, and the time complexity of representation of all documents under the reduced vector space is $O((T_r{\cdot}L_r + T_e{\cdot}L_e){\cdot}U)$. The method of the traditional text representation costs $O(1{+}(T_r{\cdot}L_r + T_e{\cdot}L_e){\cdot}U)$. So the time complexity of new text representation method is greater than that of the traditional one.

4. Experiment Setup

In this paper, we extracted k terms from each document in training set and unite them into a new text representation space, and compared the micro F1 measure and accuracy of classifiers when the number of the key terms extracted from each document is 2, 4, 6, 8, 10, 12, 14, 16, 18 or 20, respectively. Moreover, we compared the performance of three strategies with that of the traditional text representation. The three strategies are denoted by KT-of-DOC-1.0-0.8 (C_1=1.0; C_2=0.8), KT-of-DOC-1.0-1.0 (C_1=1.0; C_2=1.0) and KT-of-DOC-1.2-1.0 (C_1=1.2; C_2=1.0), respectively. The size of new feature vector space generated by our proposed method is different with regard to variable training sets. Thus, it is unreasonable to make a comparison between two or more tests, and the 10-fold cross validation was not applied in our experiments.

4.1 Feature-Selection Algorithms

There exist many sophisticated feature-selection algorithms for text categorization. In this paper, we utilized six classic feature-selection algorithms to extract key terms from each document, respectively.

Improved Gini index

Gini index is a non-purity split method and was widely used in decision tree algorithms. To apply the Gini index directly to feature selection, Shang, *et al.* {, 2007 #8} proposed the improved Gini index method. It measures the purity of feature t_k toward category c_i. The bigger the value of purity is, the better the feature is. The formula of the improved Gini index is defined as follows.

$$Gini(t_k) = \sum_i P(t_k \mid c_i)^2 P(c_i \mid t_k)^2 \tag{1}$$

Where $P(t_k| c_i)$ is the probability that a feature t_k occurs in category c_i; $P(c_i|t_k)$ refers to the conditional probability that a feature t_k belongs to category c_i when feature t_k occurs.

Information Gain

Information Gain [29] is frequently used as a criterion in the field of machine learning [3]. The Information Gain of a given feature t_k with respect to class $c_i$is the reduction in uncertainty about the value of c_i when we know the value of t_k. The larger Information Gain of a feature is, the more important the feature is for categorization. Information Gain of a feature t_k toward a category c_i can be defined as follows.

$$IG(t_k, c_i) = \sum_{c \in \{c_i, \overline{c}_i\}} \sum_{t \in \{t_k, \overline{t}_k\}} P(t,c) \log \frac{P(t,c)}{P(t)P(c)} \tag{2}$$

$$IG(t_k) = \sum_{i=1}^{C} P(c_i) IG(t_k, c_i) \tag{3}$$

Mutual Information

Mutual Information is a concept in information theory. Mutual Information measures arbitrary dependencies between random variables and is suitable for assessing the information content of a feature [30]. Mutual information is used to measure the dependence between a feature t_k and category c_i in the feature selection. A feature possessing higher Mutual Information with a category indicates that this feature contains more information about the category.

$$MI(t_k, c_i) = \log \frac{P(t_k, c_i)}{P(t_k)P(c_i)} \tag{4}$$

$$MI(t_k) = \sum_{i=1}^{C} P(c_i) MI(t_k, c_i) \tag{5}$$

Where $P(t_k,c_i)$ is the probability that feature t_k occurs in document x and x belongs to category c_i;

Odds Ratio

Odds Ratio [31] is known in information retrieval. It calculates the odds of a term t_k occurring in category c_i normalized by the odds of term t_k occurring in the other categories [12]. If the odds ratio value of a given term t_k for categoryc_i is higher, it can be interpreted that the termt_k is more important the categoryc_i. The odds ratio of a term t_k for category c_i is defined as follows.

$$OR(t_k, c_i) = \frac{P(t_k \mid c_i)(1 - P(t_k \mid \overline{c}_i))}{(1 - P(t_k \mid c_i)P(t_k \mid \overline{c}_i))} \tag{6}$$

$$OR(t_k) = \sum_{i=1}^{C} P(c_i) OR(t_k, c_i) \tag{7}$$

where $P(t_k \mid \overline{c}_i)$ refers to the probability that feature t_k occurs in all categories except category c_i.

Ambiguity Measure

Ambiguity measure was proposed by Mengle & Goharian [12]. They considered that a person could capture the topic of a document by only glancing at the document and capturing its keywords. So the most unambiguous words of a document can easily determine the category into which the document can fall. If the ambiguity measure score of a term is close to 1, the term is more unambiguous for one category. Conversely, if the ambiguity measure score of a term is close to 0, the term is more ambiguous and should be removed. The unambiguous measure of a word is defined as follows:

$$AM(t_k, c_i) = \frac{tf(t_k, c_i)}{tf(t_k)} \tag{8}$$

$$AM(t_k) = \max(AM(t_k, c_i)) \tag{9}$$

Where *tf(t$_k$,c$_i$)* is the term frequency of a term t_k in category c_i and *tf(t$_k$)* is the term frequency of a term $t_k$in the entire training set.

DIA association factor

DIA association factor [4, 16] is an important tool in automatic indexing. It is an estimate of the probability for the category c_i to be assigned to a document if this document contains the term t_k. The DIA association factor determines the significance of the occurrence of the term t_k with respect to the category c_i. The DIA association factor is defined by

$$DIA(t_k, c_i) = P(c_i \mid t_k) \tag{10}$$

$$DIA(t_k) = \sum_{i=1}^{C} P(c_i) DIA(t_k, c_i) \tag{11}$$

Where *P(c$_i$|t$_k$)* refers to the conditional probability of the feature t_k belonging to category c_i when the feature t_k occurs.

4.2. Data Sets

In order to evaluate the performance of the proposed method, three benchmark datasets – Reuters-21578, WebKB and 20-Newsgroups – were selected in this paper. During the preprocessing, all words were converted to lower case, punctuation marks were removed, stop lists were used, and no stemming was used. Term frequency of a term was used in text representation.

20-Newsgroups

The 20-Newsgroups were collected by Ken Lang [32] and has become one of the standard corpora for text categorization. It contains 19997 newsgroup postings, and all documents were assigned evenly to 20 different UseNet groups. In this paper, we randomly extract 90% documents from every class as training set; the rest is used as test set. There are 17998 documents in training set; and 1999 documents in test set. After removing the header of the document, the number of terms in the vocabulary achieves 105135.

Reuters-21578

The Reuters-21578 contains 21578 stories from the Reuters newswire [23]. All stories are non-uniformly divided into 135 categories. We used the Mod Apte split. In this paper, we only consider the top 10 categories, in which there are 7193 stories in training set, and 2878 stories in test set. After preprocessing, the resulting vocabulary contains 20133 terms.

WebKB

The WebKB is a collection of web pages from four different college web sites [12]. The 8282 web pages are non-uniformly assigned to 7 categories. In this paper, we select 4 categories, “course”, “faculty”, “project” and “student”, as our corpus. We randomly extract 90% documents from every class as training set; the rest is used as test set. As a result, there are 3780 web pages in training set, and 419 web pages in test set. During the preprocessing, HTML tags are removed. Finally, the vocabulary contains 40739 terms.

4.3 Classifiers

In this section, we briefly describe the K-Nearest Neighbors (KNN) and Support Vector

Machines (SVM) used in our study.

Support Vector Machines (SVM)

The Support Vector Machines is a higher efficient classifier in text categorization. In this study, we use LIBSVM toolkit [33], and choose linear kernel support vector machine.

K-Nearest Neighbors (KNN)

KNN [8] is a simple machine learning algorithm that classifies objects depending on the major category labels attached to the *k* training documents similar to the test object. KNN is a type of instance-based classifier, or lazy learner, since the decision is made until all the objects in the training set are scanned [4]. We used *k*=29 in this experiment. The cosine distance was used as the measure of the similarity of the objects.

4.4. Performance Measures

In order to evaluate the effectiveness of the proposed method, we measured the performance of text categorization in terms of the mirco-F1 and Accuracy [4]. The micro-F1 and Accuracy are determined by the classical informational retrieval parameters, “precision” and “recall”. Recall is the ratio of the number of the messages that are correctly identified as the positive category to the total number of the messages that actually belong to the positive category. Precision is the ratio of the number of messages that are correctly identified as the positive category to the total number of messages that are identified as the positive category.

$$P_i = \frac{TP_i}{TP_i + FP_i} \qquad R_i = \frac{TP_i}{TP_i + FN_i}$$

$$P_{micro} = \frac{TP}{TP+FP} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)} \qquad R_{micro} = \frac{TP}{TP+FN} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)}$$

$$F1_{micro} = \frac{2P_{micro}R_{micro}}{P_{micro} + R_{micro}} \qquad Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

where TP_i is the number of the documents that are correctly classified to category c_i;FP_i is the number of the documents that are misclassified to the category c_i; TN_i is the number of the documents that are correctly classified to other categories excluding the category c_i;FN_i is the number of the documents which belong to category c_i and are misclassified to other categories. $|C|$ is the amount of categories.

5. Results

5.1 Results of Algorithm for SVM

In terms of different feature-selection methods, the size of new vector space consisting of key terms is also different. Table 1 shows the number of the features in new vector space when the number of the key terms extracted from the every document is 10 or 20, respectively. The results in Table 1 show that the size of the new vector space is higher when Odds Ratio and Ambiguity Measure are used on three datasets, respectively.

Table 2-4 show the micro F1 measure of SVM using KT-of-TR and KT-of-DOC combined with each feature selection algorithm on 20-Newsgroups, Reuters-21578 and WebKB, respectively. It can be seen from Table 2-3 that the micro F1 measure of KT-of-DOC based on various feature selection algorithms is entirely higher than that of KT-of-TR. When KT-of-DOC combined with GINI, MI, OR, AM and DIA is used on WebKB, the performance of SVM is superior to that based on KT-of-TR. The performance of micro F1 measure of SVM using KT-of-DOC combined with IG on WebKB is superior to that of the KT-of-TR when the number of the key terms is 20. Table

4 indicates that the increasement of the micro F1 measure of SVM is great when the KT-of-DOC combined with AM and OR.

Table 1. The Number of the Features in New Vector Space when the Level of Key Terms Extracted from Every Document Increases Gradually

Datasets	20-Newsgroups		Reuters-21578		WebKB	
The number of the key terms	10	20	10	20	10	20
GINI	3575	10459	1603	4389	296	1023
IG	3556	10463	1592	4361	296	1017
MI	3915	11109	2354	4921	637	1903
OR	45885	64558	9379	15901	22049	31735
AM	59244	74151	13308	16475	20742	27932
DIA	5692	11875	2136	4953	1082	2059

The curve of accuracy of SVM using KT-of-DOC and KT-of-TR combined with six feature selections on 20-Newgsroups, Reuters-21578 and WebKB are shown in Figure1 - 3, respectively. It can be seen that the accuracy of SVM using KT-of-DOC combined with all feature selection algorithms on 20-Newsgroups is superior to that using KT-of-TR when the number of key terms extracted from documents is greater than 4. Moreover, the increase in accuracy of SVM using KT-of-DOC combined with Odds Ratio and Ambiguity Measure on 20-Newsgroups is relatively larger. The accuracy of SVM using KT-of-DOC combined with Odds Ratio is decreases gradually as the size of new vector space increases.

Table 2. The Comparison of Micro F1 Measure of Support Vector Machines between KT-of-TR and KT-of-DOC when Six Feature-selection Algorithms are Applied on 20-Newsgroups, Respectively (%)

Feature selection	GINI		IG		MI		OR		AM		DIA	
number of key terms	10	20	10	20	10	20	10	20	10	20	10	20
KT-of-TR	73.78	76.09	73.27	76.14	66.09	73.84	75.37	75.54	55.10	74.63	66.40	74.51
KT-of-DOC-1.0-0.8	**76.28**	**77.99**	**76.48**	77.61	74.13	**77.01**	**79.63**	**78.77**	**79.67**	**78.94**	**76.86**	**78.30**
KT-of-DOC-1.0-1.0	75.82	77.64	75.89	**77.91**	**74.56**	76.86	78.30	77.71	78.12	77.92	76.08	77.56
KT-of-DOC-1.2-1.0	75.47	76.76	75.75	77.10	74.27	75.96	78.42	77.49	78.51	77.40	75.94	77.40

Table 3. The Comparison of Micro F1 Measure of Support Vector Machines between KT-of-TR and KT-of-DOC when Six Feature-selection Algorithms are Applied on Reuters-21578, Respectively (%)

Feature selection	GINI		IG		MI		OR		AM		DIA	
number of key terms	10	20	10	20	10	20	10	20	10	20	10	20
KT-of-TR	67.79	67.83	67.36	67.76	65.97	66.07	51.55	57.06	64.25	66.82	57.24	64.53
KT-of-DOC-1.0-0.8	68.32	68.07	67.51	68.05	**67.99**	67.68	**68.30**	**68.08**	**68.44**	**68.09**	**68.01**	**68.45**
KT-of-DOC-1.0-1.0	68.13	**68.41**	**68.25**	**68.21**	67.96	**67.73**	67.68	67.86	68.01	**68.09**	67.67	67.96
KT-of-DOC-1.2-1.0	**68.47**	67.98	67.73	67.99	68.24	67.55	67.96	67.96	68.10	68.06	67.79	68.19

Table 4. The Comparison of Micro F1 Measure of Support Vector Machines between KT-of-TR and KT-of-DOC when Six Feature-selection Algorithms are Applied on WebKB, Respectively (%)

Feature selection	GINI		IG		MI		OR		AM		DIA	
number of key terms	10	20	10	20	10	20	10	20	10	20	10	20
KT-of-TR	79.44	76.17	**78.93**	77.89	46.28	60.60	39.95	52.52	38.82	70.06	68.66	70.32
KT-of-DOC-1.0-0.8	78.35	79.96	76.79	**79.64**	63.25	78.96	**78.11**	**76.65**	77.81	**78.55**	**73.06**	78.68

KT-of-DOC-1.0-1.0	**79.66**	78.53	78.00	79.06	**64.26**	78.30	77.00	75.47	77.61	78.43	70.21	76.61
KT-of-DOC-1.2-1.0	79.20	**80.34**	77.73	78.61	62.81	**79.08**	77.87	76.48	77.63	78.39	70.90	**78.69**

Figure2. shows that the accuracy of SVM using KT-of-DOC combined with all feature selection algorithms on Reuters-21578 is superior to that using KT-of-TR. Figure2. indicates that the differences of the accuracy between using KT-of-DOC and KT-of-TR is decreased gradually with the increment of the number of the key terms. When the KT-of-DOC combined with GINI and IG is used on WebKB, the performance curve of SVM is not optimal. The accuracy of three KT-of-DOC-based strategies combined with GINI on WebKB is superior to that of KT-of-TR when the number of key terms is greater than 12. It can be seen from Figure 3 (b) that the curve of the three KT-of-DOC-based strategies combined with IG is higher than that of KT-of-TR when the number of the key terms is 2, 16, 18 and 20. In the other hand, Figure 1 - 3 imply that the curve of the KT-of-DOC-1.0-0.8 is higher than others.

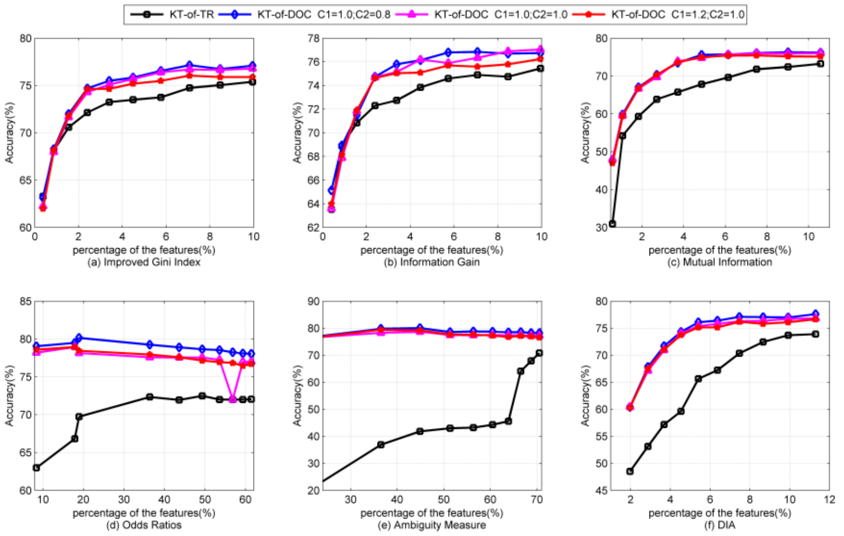


Figure 1. Comparison of Accuracy of the SVM Based on KT-of-TR with KT-of-DOC Combined with Six Feature Selections on 20-Newsgroups, Respectively. X-axis Denotes the Percentage of Selected Features when the Different Level Key Terms are Selected

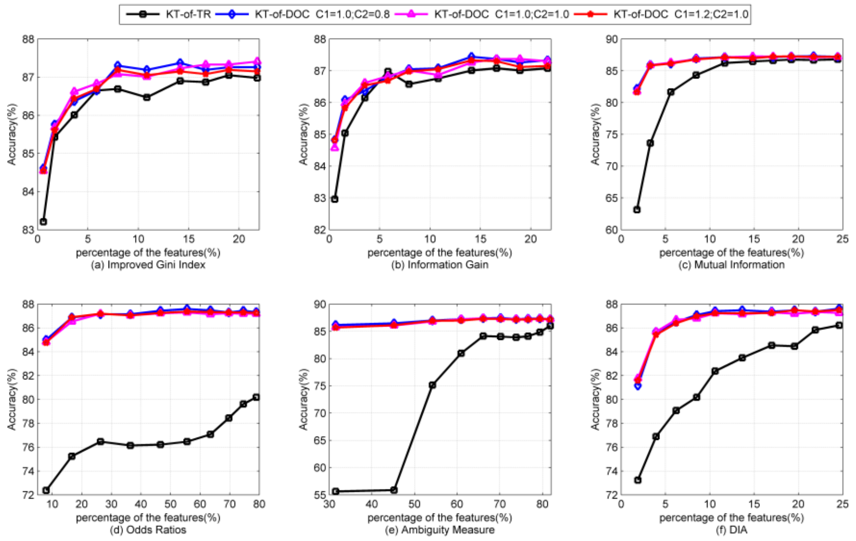


Figure 2. Comparison of Accuracy of the SVM Based on KT-of-TR with KT-of-DOC Combined with Six Feature Selections on Reuters-21578, Respectively. X-axis Denotes the Percentage of Selected Features when the Different Level Key Terms are Selected

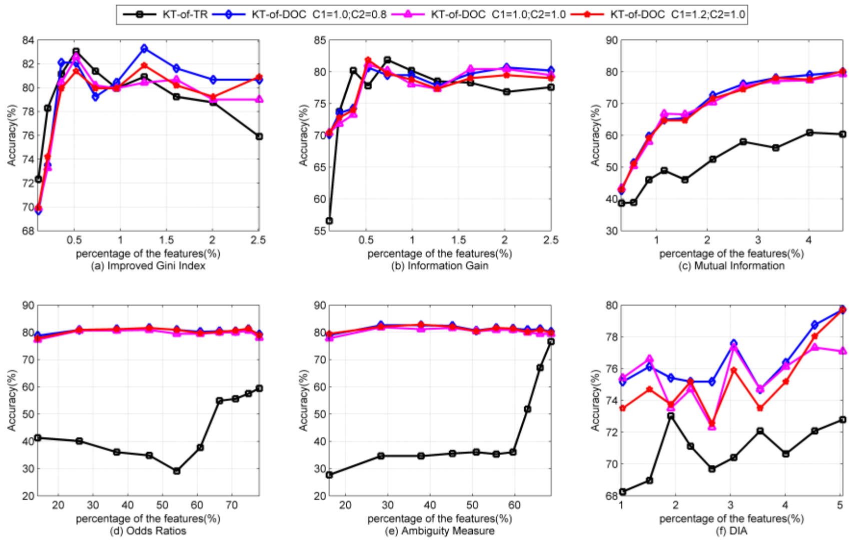


Figure 3. Comparison of Accuracy of the SVM Based on KT-of-TR with KT-of-DOC Combined with Six Feature Selections on WebKB, Respectively. X-axis Denotes the Percentage of Selected Features when the Different Level Key Terms are Selected

5.2 Results of Algorithm for KNN

The micro F1 measure of KNN using KT-of-TR and three KT-of-DOC-based strategies on 20-Newsgroups, Reuters-21578 and WebKB are shown in the Table 5 - 7, respectively. It can be seen from Table 5 that the micro F1 measure of three KT-of-DOC-based strategies combined with GINI and IG on 20-Newsgroups is inferior to that of KT-of-TR and the micro F1 measure of KT-of-DOC-1.0-1.0 combined with GINI and IG is the highest of the three KT-of-DOC-based strategies. When KT-of-DOC combined with MI and DIA is applied on 20-Newsgroups, the micro F1 measure of KNN is superior to that of KT-of-TR. It is indicated in Table 6 that when KT-of-DOC combined with GINI is used on Reuters-21578, its micro F1 performance is superior to that of KT-of-TR when the number of the key terms is 20. When the number of key terms is 20, the micro F1 measure of KNN using three KT-of-DOC-based strategies combined with IG on Reuters-21578 is superior to that of KT-of-TR. The micro F1 measure of KNN using three KT-of-DOC-based strategies combined with MI, OR, AM and DIA on Reuters-21578 is almost better than that of KT-of-TR. It can be seen from Table 7 that the micro F1 performance of KNN using KT-of-DOC combined with GINI on WebKB is superior to that of KT-of-TR when the number of key terms is 20. When KT-of-DOC combined with IG is used on WebKB, its micro F1 performance is superior to that of KT-of-TR. The micro F1 performance of three KT-of-DOC-based strategies combined with MI, OR, AM and DIA used on WebKB is superior to that of KT-of-TR.

Figure 4 - 6 show the curves of accuracy of KNN using KT-of-DOC and KT-of-TR combined with six feature selection algorithms on 20-Newsgroups, Reuters-21578 and WebKB, respectively. It can be seen from Figure 4(a) and Figure 4(b) that the curves of three KT-of-DOC-based strategies are lower than that of KT-of-TR. Figure 4(c) and Figure 4(f) indicate that the curves of three KT-of-DOC-based three strategies combined with MI and DIA are higher than that of KT-of-TR, but the performance increases with a little range. The curve of accuracy of KNN using KT-of-TR combined with OR on 20-Newsgroups suddenly rises and coincides with the curve of KT-of-DOC-1.0-0.8 and KT-of-DOC-1.2-1.0 after the number of key terms is more than 4.

Table 5. The Comparison of Micro F1 Measure of KNN between KT-of-TR and KT-of-DOC when Six Feature-selection Algorithms are Applied on 20-Newsgroups, Respectively (%)

Feature selection	GINI		IG		MI		OR		AM		DIA	
number of key terms	10	20	10	20	10	20	10	20	10	20	10	20
KT-of-TR	**71.23**	**72.66**	**70.96**	**73.26**	62.98	69.98	75.85	**76.60**	72.95	**75.14**	60.53	72.08
KT-of-DOC-1.0-0.8	69.49	71.46	68.34	70.96	67.03	70.64	75.77	76.59	**75.84**	74.90	69.77	71.97
KT-of-DOC-1.0-1.0	70.22	71.94	70.94	72.34	**68.53**	**71.98**	73.78	74.37	73.14	73.27	**70.65**	**72.63**
KT-of-DOC-1.2-1.0	69.45	71.90	68.87	71.42	67.47	71.11	**75.87**	76.42	75.83	74.80	70.02	72.26

Table 6. The Comparison of Micro F1 Measure of KNN between KT-of-TR and KT-of-DOC when Six Feature-selection Algorithms are Applied on Reuters-21578, Respectively (%)

Feature selection	GINI		IG		MI		OR		AM		DIA	
number of key terms	10	20	10	20	10	20	10	20	10	20	10	20
KT-of-TR	**66.65**	65.82	**66.21**	65.88	63.95	64.02	45.98	54.02	59.86	**66.77**	54.76	60.20
KT-of-DOC-1.0-0.8	65.87	66.14	65.55	66.09	64.69	64.58	**65.26**	65.09	**66.19**	65.20	65.23	65.10
KT-of-DOC-1.0-1.0	65.96	**66.30**	66.02	**66.48**	**65.42**	**65.69**	64.62	**65.27**	65.01	65.56	**65.63**	**65.82**
KT-of-DOC-1.2-1.0	65.88	65.99	65.78	66.27	64.90	64.93	65.19	65.27	65.96	65.28	64.98	65.10

Table 7. The Comparison of Micro F1 Measure of KNN between KT-of-TR and KT-of-DOC when Six Feature-selection Algorithms are Applied on WebKB, Respectively (%)

Feature selection	GINI		IG		MI		OR		AM		DIA	
number of key terms	10	20	10	20	10	20	10	20	10	20	10	20
KT-of-TR	**64.97**	59.88	61.23	57.94	42.09	44.92	34.63	54.89	32.30	**62.71**	54.62	55.60
KT-of-DOC-1.0-0.8	60.67	**60.90**	59.66	60.40	**55.64**	57.98	**61.03**	**60.59**	60.96	59.82	54.80	61.38
KT-of-DOC-1.0-1.0	63.07	59.17	**62.16**	59.80	55.55	**60.14**	59.79	59.17	60.32	61.34	**55.30**	**62.61**
KT-of-DOC-1.2-1.0	61.59	59.91	61.11	**61.97**	55.26	58.50	60.59	60.45	**61.01**	59.55	54.60	62.50

It can be seen from Figure 5. that the curve of accuracy of KNN using KT-of-DOC on Reuters-21578 almost reaches the highest point when the number of key terms is 4 and then decreases gradually. The curve of accuracy of KNN using KT-of-DOC-1.0-1.0 combined with GINI, IG and MI on WebKB is higher than that of the other two KT-of-DOC-based strategies. When the KT-of-DOC combined with OR, AM and DIA is used on Reuters-21578, the increment of accuracy of KNN is relatively larger. Figure 6 (a) indicates that the curve of accuracy of KT-of-DOC-1.0-1.0 combined with GINI is higher than that of KT-of-TR except for the number of key terms is 4,6,10 or 12. Moreover, when the number of key terms is 4, the curve of accuracy of KT-of-DOC-1.0-1.0 combined with GINI reaches the peak, 68.02%. Although the curve of accuracy of three KT-of-DOC-based strategies combined with IG on WebKB is not optimal, the accuracy of KT-of-DOC-1.0-1.0 combined with IG is superior to that of KT-of-TR when the number of key terms is greater than 6, and its curve reaches the highest point, 66.59%, when the number of key terms is 10. It can be seen from Figure 6. that the curves of accuracy of three KT-of-DOC-based strategies combined with MI, OR, AM and DIA on WebKB are higher than that of KT-of-TR.

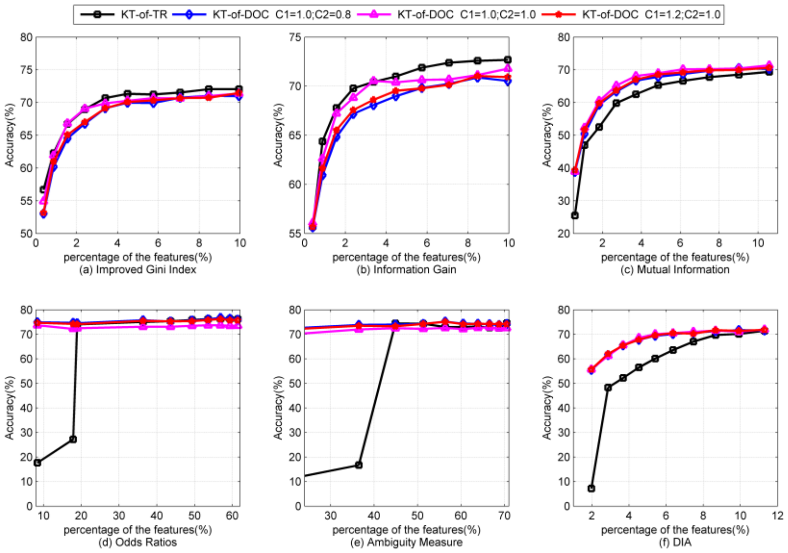


Figure 4. Comparison of Accuracy of the KNN Based on KT-of-TR with KT-of-DOC Combined with Six Feature Selections on 20-Newsgroups, Respectively. X-axis Denotes the Percentage of Selected Features when the Different Level Key Terms are Selected

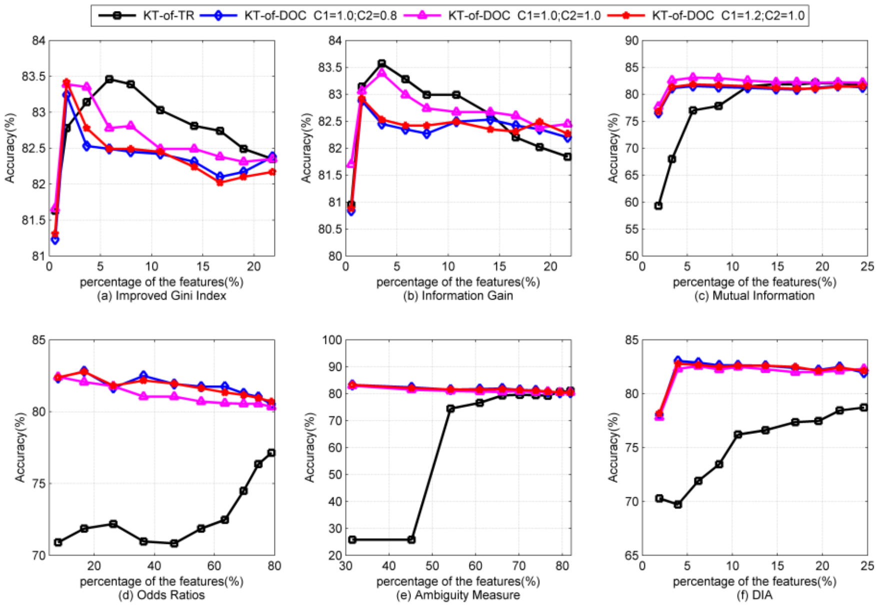


Figure 5. Comparison of Accuracy of the KNN Based on KT-of-TR with KT-of-DOC Combined with Six Feature Selections on Reuters-21578, Respectively. X-axis Denotes the Percentage of Selected Features when the Different Level Key Terms are Selected

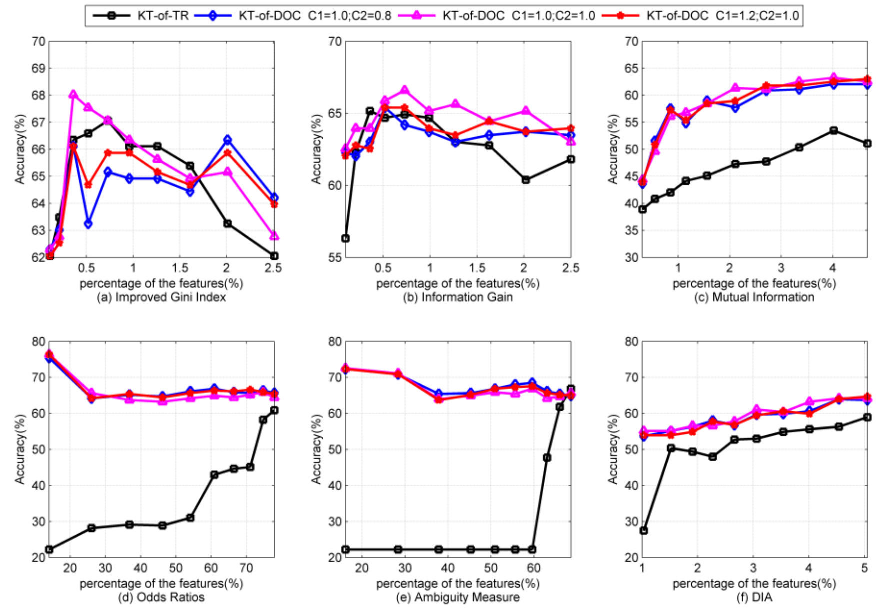


Figure 6. Comparison of Accuracy of the KNN Based on KT-of-TR with KT-of-DOC Combined with Six Feature Selections on WebKB, Respectively. X-axis Denotes the Percentage of Selected Features when the Different Level Key Terms are Selected

6. Discussions

In order to compare the performance of the proposed new representation method with the traditional approaches, Friedman and Iman & Davenport [34] test are used in the statistical analysis. The null hypothesis of them is that all the algorithms are equivalent and so the ranks of all algorithms should be equal [35]. If the null hypothesis of Friedman and Iman & Davenport test is rejected, the post test (Holm test) [36] can be used to detect the significant differences among all the methods. In Holm test, all the hypotheses are ordered by their significances (p_i), and then the corresponding p_i and α/(k-i) are compared (k is the number of the algorithms tested; i is the rank of the hypothesis ordered by their significance; α is the confidence level). Holm test starts with the most significant p-value. If p_1 is below α/(k-1), the corresponding hypothesis is rejected, and then the p_2 and α/(k-2) are compared. If the corresponding hypothesis is rejected, the next hypothesis is tested until a certain null hypothesis cannot be rejected. If the null hypothesis is rejected, there is a significant difference between two corresponding algorithms. In our experiments, we statistically compared three KT-of-DOC-based strategies with KT-of-TR using the classification accuracy. Table 8 and Table 9 show the Holm [35] test table for α = 0.05 when SVM and KNN classifier are used, respectively. It can be seen from Table 8 that the accuracy of three KT-of-DOC-based strategies significantly outperform that of KT-of-TR, and the accuracy of KT-of-DOC-1.0-0.8 is superior to that of KT-of-DOC-1.0-1.0 and KT-of-DOC-1.2-1.0. Table 9 indicates that when K-Nearest-Neighbor classifier is used, the accuracy of KT-of-DOC-1.0-1.0 significantly outperforms that of KT-of-TR.

In this paper, the performance of KNN and SVM using KT-of-DOC on three benchmark collections is almost superior to that using KT-of-TR. We think there are at least two principle factors that bring the results mentioned above; (I) the documents in training set and test set are represented by enough terms in new text representation method, however, it cannot be guaranteed in the traditional text representation, (II) the terms used to represent a document are best features of the document and carry more category information.

Table 8. Holm Test Table for α = 0.05 when Support Vector Machines is Used

i	*algorithms*	*z=(R_0-R_i)/SE*	*p-value*	*Holm*
6	KT-of-TR vs. KT-of-DOC-1.0-0.8	6.5841	4.58E-11	0.0083
5	KT-of-DOC-1.0-0.8 vs. KT-of-DOC-1.2-1.0	3.7439	1.81E-04	0.0100
4	KT-of-TR vs. KT-of-DOC-1.0-1.0	3.4857	4.91E-04	0.0125
3	KT-of-DOC-1.0-0.8 vs. KT-of-DOC-1.0-1.0	3.0984	0.0019	0.0167
2	KT-of-TR vs. KT-of-DOC-1.2-1.0	2.8402	0.0045	0.0250
1	KT-of-DOC-1.0-1.0 vs. KT-of-DOC-1.2-1.0	0.6455	0.5186	0.0500

Table 9. Holm Test Table for α= 0.05 when K-Nearest Neighbor is Used

i	*algorithms*	*z=(R_0-R_i)/SE*	*p-value*	*Holm*
6	KT-of-TR vs. KT-of-DOC-1.0-1.0	3.0984	0.0019	0.0083
5	KT-of-TR vs. KT-of-DOC-1.2-1.0	2.3238	0.0201	0.0100
4	KT-of-TR vs. KT-of-DOC-1.0-0.8	1.8074	0.0707	0.0125
3	KT-of-DOC-1.0-0.8 vs. KT-of-DOC-1.0-1.0	1.2909	0.1967	0.0167
2	KT-of-DOC-1.0-1.0 vs. KT-of-DOC-1.2-1.0	0.7746	0.4386	0.0250
1	KT-of-DOC-1.0-0.8 vs. KT-of-DOC-1.2-1.0	0.5164	0.6056	0.0500

When the KT-of-DOC combined with Odds Ratio (OR) and Ambiguity Measure (AM) is used, it can be seen that the number of the features in new vector space is enormous even if only a few of key terms were extracted from every document. The main reason is that the Odds Ratio algorithm only selects the positive features and neglects the negative features [15], so the probability of key terms extracted from each document overlapping is low.

We analyzed the features in the feature vector space generated by KT-of-DOC and KT-of-TR based on six classic feature-selection algorithms, respectively. Some features are commonly selected by both KT-of-DOC and KT-of-TR. Figure 7. shows the proportion of the common features in the feature vector space generated by KT-of-DOC and KT-of-TR based on six feature selection algorithms on 20-Newsgroups. It is worth noting that most of features (about 65%) are commonly selected by KT-of-DOC and KT-of-TR based on various feature selection algorithms except for DIA association factor. In the other hand, as the number of key terms increases, the proportion of common features will be increased gradually up to 100%. We utilized the features only selected by KT-of-DOC or KT-of-TR to construct the vector space into which the documents in the corpus were mapped. Figure 8 indicates the comparison of accuracy between the vector spaces which consist of the features selected only by the KT-of-DOC or KT-of-TR on 20-Newsgroups. It can be seen from Figure 8 (c) – (f) that the performance of the vector space that consists of the features only selected by KT-of-DOC is superior to that by KT-of-TR. However, when Gini and IG are used, the performance of the features only in KT-of-DOC is inferior to that in KT-of-TR. This result is contrary to that the vector space consists of all the features selected by KT-of-DOC or KT-of-TR. We think there are two reasons that can explain this phenomenon: (1) Gini and IG are two of the best feature selection algorithms, so the features selected by them contain more category information. (2) Although the performance of the features selected only by KT-of-TR outperforms that by KT-of-DOC, the features selected by KT-of-DOC can capture the topic of the document and the relationship among the key terms.

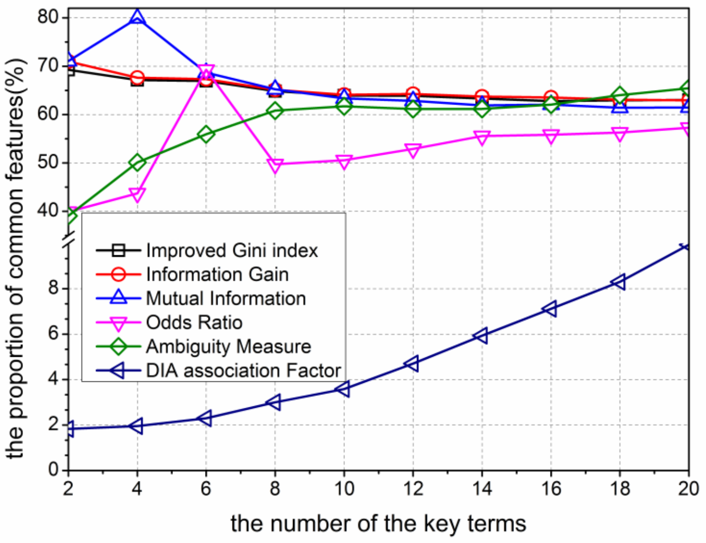
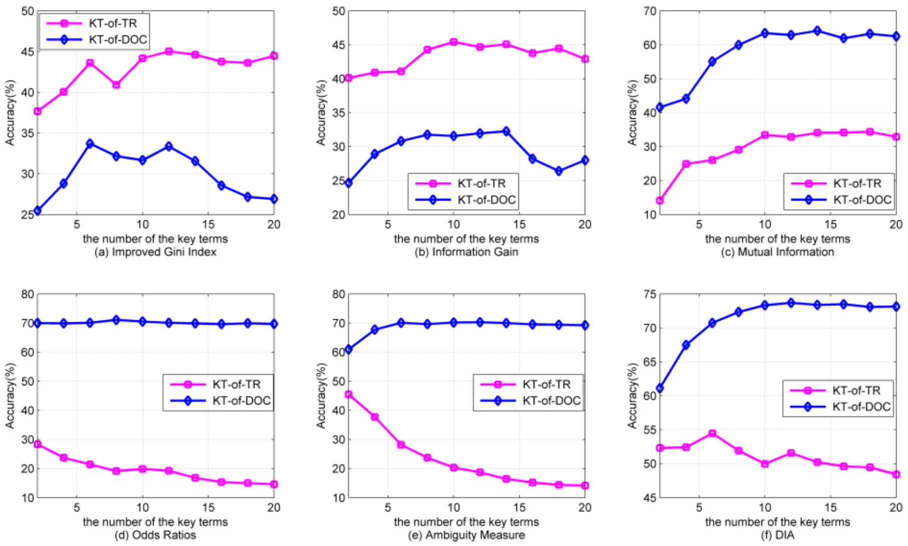


Figure 7. The Proportion of the Common Features in the Feature Vector Space Generated by KT-of-DOC and KT-of-TR Based on Six Feature Selection Algorithms on 20-Newsgroups, Respectively



(a) Improved Gini Index

(b) Information Gain

(c) Mutual Information

(d) Odds Ratios

(e) Ambiguity Measure

(f) DIA

Figure 8.The Comparison Results in Terms of Accuracy between the Vector Spaces which Consist of the Features Selected Only by KT-of-DOC or KT-of-TR on 20-Newsgroups

There is a characteristic shared by Malik's and our method that every document in the training set is represented by at least *k* features. However, the two approaches have some essential differences which are listed as follows: (1) The size of new vector space in the Malik's method is firstly predefined based on the number of documents in the training set and the size of the vocabulary; however, the size of new vector space in our method is dynamically determined according to the number of key terms of documents in the training set; (2) The new vector spaces constructed by our method and Malik's method are different, namely the elements of new vector space constructed by our method are different from that by Malik's; (3) Though both Malik's and our method keep that every document is represented by at least *k* features, there is a difference between "*k* features" in two methods. It is enough for Malik's representation method that each document contains *k* features; however, the document represented by our method may contain more than *k* features, but only k features are considered as the representative features; (4) Malik's method only emphasizes on reducing the sparsity of the document representation, such as guaranteeing each document has at least *k* features in the document vector. However, our method not only reduces the sparisity of the documents representation, but also strengthens the contribution of key terms of the document and weakens that of non-key terms of the document; (5) In Malik's method, in order to ensure local coverage, the features in documents, which are not properly covered by the selected features, are sorted according to TF*Information Gain. The features selected by this method are different from that by our method. Figure 9 shows the comparison results between Malik's and our method in term of accuracy using SVM and KNN on 20-newsgroups, Reuters and WebKB, respectively. It can be seen from Figure 9 that the performance of KT-of-DOC is superior to that of Malik's method when the SVM is used on Reuters; the performance of KT-of-DOC outperforms that of Malik's method when KNN is used on 20-newsgroups and WebKB.

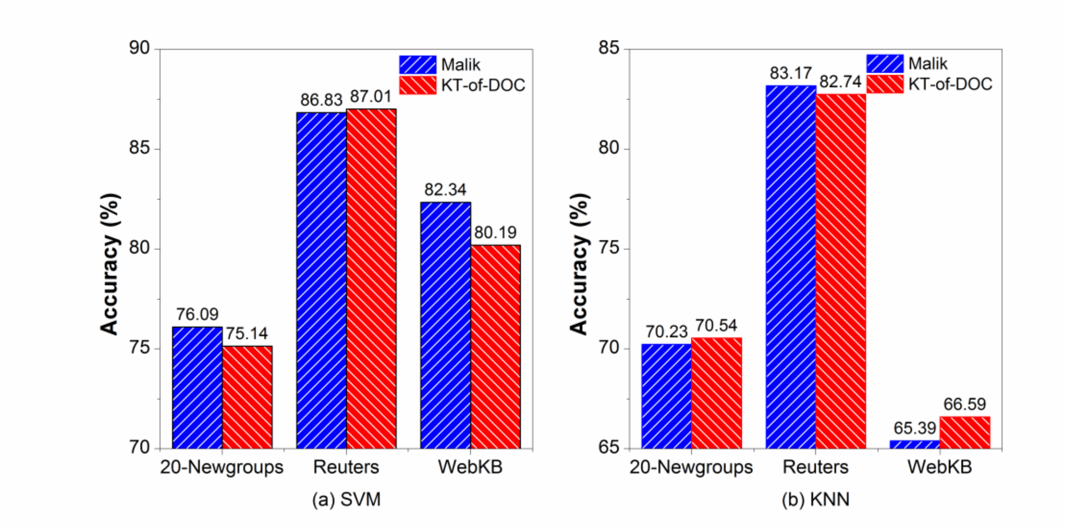


Figure 9.Comparison of KT-of-DOC with Malik's Method in Terms of Accuracy Using Support Vector Machines and K-Nearest Neighbors on 20-newsgroups, Reuters and WebKB, Respectively

In order to control the sparsity of the vector space, Mladenić, *et al*. [21] only used a fixed number of terms selected from a document according to the feature selection algorithm to represent one document, and they did not consider the effect of non-key terms. When C_1 =1.0 and C_2=0.0, the proposed text presentation algorithm is consistent with Mladenić's method.

7. Conclusion

In text categorization, feature selection is commonly used to reduce the dimensionality of the vector space and improve the performance of the classifier. After feature selecting, each raw document in training set and test set is re-represented as a vector in new reduced vector space. In fact, the essential of this method is that the document is only represented by the most informative terms in the training set instead of all terms. However, the most informative terms do not always appear in every document, so it is inevitable that only few or even no informative terms occur in some documents. In order to guarantee that these documents can be correctly classified, we proposed a new text representation algorithm, KT-of-DOC. The main idea of KT-of-DOC is that every document is represented by a certain amount of key terms, which are most effective for categorization and extracted from the document itself. In our experiments, we extracted key terms from every document based on six feature selection algorithms, Improved Gini Index, Information Gain, Mutual Information, Odds Ratio, Ambiguity Measure and DIA association factor, respectively, and selected three benchmark collections, 20-Newsgroups, Reuters-21578 and WebKB as our datasets. Two classifiers (Support Vector Machines and K-Nearest Neighbors) are used to compare the performance of the proposed text representation. The experiments show that the proposed text representation can significantly improve the performance of classifiers.

In this paper, we use feature selection algorithm to extract key terms from a document. So efficient method that extracts key terms from a document is our future work.

Acknowledgment

This research is supported by the project development plan of science and technology of Jilin Province under Grant no. 20140204071GX.

References

[1] F. Song, S. Liu, and J. Yang, “A comparative study on text representation schemes in text categorization”, Pattern Analysis & Applications, vol. 8, no. 1, **(2005)**, pp. 199-209.

[2] M. Radovanović and M. Ivanović, “Interactions Between Document Representation and Feature Selection in Text Categorization”, Database and Expert Systems Applications,Springer Berlin / Heidelberg, **(2006)**.

[3] Y. Yang and J. O. Pedersen, “A Comparative Study on Feature Selection in Text Categorization”, Proceedings of ICML-97, Nashville, TN, **(1997)**.

[4] F. Sebastiani, “Machine Learning in Automated Text Categorization”, ACM Computing Surveys, vol. 34, no. 1, **(2002)**, pp. 1-47.

[5] D. Fragoudis, D. Meretakis and S. Likothanassis, “Best terms: an efficient feature-selection algorithm for text categorization”, Knowledge and Information Systems, vol. 8, no. 1, **(2005)**, pp. 16-33.

[6] A. McCallum and K. Nigam, “A Comparison of Event Models for Naive Bayes Text Classification”, AAAI-98 Workshop on Learning for Text Categorization, **(1998)**.

[7] H. Drucker, D. Wu and V. N. Vapnik, “Support Vector Machines for Spam Categorization”, IEEE Transactions on Neural Networks, vol. 10, **(1999)**, pp. 1048-1054.

[8] T. Cover and P. Hart, “Nearest neighbor pattern classification,” Information Theory, IEEE Transactions on, vol. 13, no. 1, **(1967)**, pp. 21-27.

[9] X. X. Bing and Z. Z. Hua, “Distributional Features for Text Categorization”, IEEE Transactions onKnowledge and Data Engineering, vol. 21, no. 3, **(2009)**, pp. 428-442.

[10] H. Ogura, H. Amano and M. Kondo, “Feature selection with a measure of deviations from Poisson in text categorization”, Expert Systems with Applications, vol. 36, **(2009)**, pp. 6826-6832.

[11] D. Koller and M. Sahami, “Hierarchically classifying documents using very few words,” Proceedings of the Fourteenth International Conference on Machine Learning(ML-97), Nashville,Tennessee, **(1997)**.

[12] S. S. R. Mengle and N. Goharian, “Ambiguity Measure Feature-Selection Algorithm”, Journal of the American Society for Information Science and Technology, vol. 60, no. 5, **(2009)**, pp. 1037-1050.

[13] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu and Z. Wang, “A novel feature selection algorithm for text categorization”, Expert Systems with Applications, vol. 33, no. 1, **(2007)**, pp. 1-5.

[14] H. Peng, F. Long and C. Ding, “Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, andMin-Redundancy”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, **(2005)**, pp. 1226-1238.

[15] J. Chen, H. Huang, S. Tian and Y. Qu, “Feature selection for text classification with Naive Bayes”, Expert Systems with Applications, vol. 36, **(2009)**, pp. 5432-5435.

[16] N. Fuhr, S. Hartmann, G. Lustig, M. Schwantner, K. Tzeras, T. H. Darmstadt, F. Informatik and G. Knorz, “AIR/X - a Rule-Based Multistage Indexing System for Large Subject Fields”, Proceedings of RIAO-91, 3rd International Conference "Recherche d'Information Assistee par Ordinateur", Barcelona,Spain, **(1991)**.

[17] J. Yang, Y. Liu, Z. Liu, X. Zhu and X. Zhang, “A new feature selection algorithm based on binomial hypothesis testing for spam filtering”, Knowledge-Based Systems, vol. 24, no. 6, **(2011)**, pp. 904-914.

[18] F. Domingos, “Control-Sensitive Feature Selection for Lazy Learners”, Artificial Intelligence Review, vol. 11, no. 1, **(1997)**, pp. 227-253.

[19] W. W. Cohen and Y. Singer, “Context-sensitive learning methods for text categorization”, ACM Transaction Information System, vol. 17, no. 2, **(1999)**, pp. 141-173.

[20] A. Kolcz, “Local sparsity control for naive Bayes with extreme misclassification costs”, Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, Chicago, Illinois, USA, **(2005)**.

[21] D. Mladenić, J. Brank, M. Grobelnik and N. M. Frayling, “Feature selection using linear classifier weights: interaction with classification models”, Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, Sheffield, UK, **(2004)**.

[22] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval”, Information Processing & Management, vol. 24, no. 5, **(1988)**, pp. 513-523.

[23] R. Bekkerman, R. El-Yaniv, N. Tishby and Y. Winter, “Distributional word clusters vs. words for text categorization”, Journal Mach. Learn. Res., vol. 3, **(2003)**, pp. 1183-1208.

[24] N. Tishby, F. C. Pereira and W. Bialek, “The information bottleneck method”, Proceeding of the 37-th Allerton Conference on Communication and Computation, **(1999)**.

[25] L. Chen, J. Zeng and N. Tokuda, “A “stereo” document representation for textual information retrieval,”, Journal of the American Society for Information Science and Technology, vol. 57, no. 6, **(2006)**, pp. 768-774.

[26] P. Graham., “A Plan for Spam”, http://paulgraham.com/spam.html.

[27] H. H. Malik and J. R. Kender, “Classification by Pattern-Based Hierarchical Clustering”, ECML/PKDD, **(2008)**.

[28] H. H. Malik and J. R. Kender, “Classifying High-Dimensional Text and Web Data using Very Short Patterns”, Proceeding of the IEEE International Conference on Data Mining, Los Alamitos, **(2008)**.

[29] J. R. Quinlan, “Induction of Decision Trees”, Machine Learning, vol. 1, no. 1, **(1986)**, pp. 81-106.

[30] R. Battiti, “Using Mutual Information for Selecting Featuresin Supervised Neural Net Learning”, IEEE Transactions on Neural Networks, vol. 5, no. 4, **(1994)**, pp. 537-550.

[31] D. Mladenic and M. Grobelnik, “Feature selection for classification based on text hierarchy”, Conference on Automated Learning and Discovery (CONALD-98), Pittsburgh, PA, **(1998)**.

[32] K. Lang, “NewsWeeder: Learning to filter netnews”, Proceedings of ICML-95, 12th International conference on machine learning, **(1995)**.

[33] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines”, http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[34] R. L. Iman and J. M. Davenport, “Approximations of the critical region of the Friedman statistic”, Communications in Statistics, vol. 18, **(1980)**, pp. 571-579.

[35] J. Demšar, “Statistical Comparisons of Classifiers over Multiple Data Sets”, Journal Mach. Learn. Res., vol. 7, **(2006)**, pp. 1-30.

[36] S. García, A. Fernández, J. Luengo and F. Herrera, “A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability”, Soft Computing - A Fusion of Foundations, Methodologies and Applications, vol. 13, no. 10, **(2009)**, pp. 959-977.

Authors

Jie Ming Yang, received his MSc and PhD degree in computer applied technology from Northeast DianLi University, China in 2008 and Computer Science and Technology from Jilin University, China in 2013, respectively. His research interests are in machine learning and data mining.

Zhi-Ying Liu, received her MSc degree in computer applied technology from Northeast DianLi University, China in 2005. His research interests are in information retrieval and personalized recommendation.

Zhao-Yang Qu, received his MSc and PhD degree in computer science from Dalian University of Technology, China in 1988 and North China Electric Power University, China in 2012, respectively. His research interests are in artificial intelligence, machine learning and data mining.