Pattern Anal Applic (2005) 8: 199–209
DOI 10.1007/s10044-005-0256-3

ORIGINAL ARTICLE

Fengxi Song · Shuhai Liu · Jingyu Yang

A comparative study on text representation schemes in text categorization

Received: 9 June 2003 / Accepted: 2 December 2004 / Published online: 28 July 2005

Abstract It is well known that the classification effectiveness of the text categorization system is not simply a matter of learning algorithms. Text representation factors are also at work. This paper will consider the ways in which the effectiveness of text classifiers is linked to the five text representation factors: ''stop words removal'', ''word stemming'', ''indexing'', ''weighting'', and ''normalization''. Statistical analyses of experimental results show that performing ''normalization'' can always promote effectiveness of text classifiers significantly. The effects of the other factors are not as great as expected. Contradictory to common sense, a simple binary indexing method can sometimes be helpful for text categorization.

Keywords Text categorization · Text representation · Support vector machines · Multi-way analysis of variance · Pattern recognition

1 Introduction

Text categorization, the assignment of free text documents to one or more predefined categories based on their content, goes back at least to the early 1960s and to Maron's seminal work [1]. With the rapid growth of electronic text documents on the Internet and corporate intranets, as a potential tool for better finding, filtering, and managing these resources, text categorization has gained more and more attention in recent years [2, 3].

While many researchers apply various machine learning algorithms in promoting the effectiveness of text classifiers [4–10], few people systematically compare and statistically analyze the impact of different text representations on the generalization accuracy of text categorization systems. There are still some questions left unanswered:

- Among the possible text representation schemes which are probably the best?
- Among the factors that may affect a text representation which are the most important and should be dealt with seriously?
- Is ''stop words removal'' an indispensable step to represent a text document?
- Does indexing a text document with term frequency always outperform indexing it with binary value?
- Whether ''word stemming'' is harmful or beneficial for text categorization?
- How should we represent text document the best?

All these and other questions are the main concerns of this paper.

By extensive experiments of 32 different text representation schemes on two benchmark dataset Reuters-21578 and 20 Newsgroups, and careful statistical analyses of these experimental results, most of the concerned questions have been experientially answered in this paper.

2 Text representation

Text documents, which typically are strings of characters, are not amenable to being interpreted by a classifier. Because of this, they have to be transformed into succinct representations suitable for the learning algorithm and the classification task. The choice of a representation for text depends on what one regards as the meaningful textual units and the meaningful natural language rules for the combination of these units. Inherited from Information Retrieval, each document is usually represented by a vector of weighted terms, the basic meaningful textual units, in text categorization. Though richer text representations that regard phrases as basic textual units have been attempted by some

F. Song (✉) · S. Liu · J. Yang
Department of Computer Science,
Nanjing University of Science and Technology, China
E-mail: songfengxi@yahoo.com
Tel.: +86-25-4315751
Fax: +86-25-4315510

researchers, many experimental results show that it will yield better categorization effectiveness to use words as the meaningful textual units instead. In text categorization, all text representation approaches that represent text document by a vector of weighted words are named as *Bag of Words*, which corresponds to Salton's *Vector Space Model* (*VSM*) [11].

The process of converting a text document into a vector of weighted words is mainly divided into two phases. In the first phase, the vocabulary used to represent text documents (hereafter it is called text representation dictionary or simply dictionary) is defined. In the second phase, each text document is mapped into its feature vector based on both the text representation dictionary and the context of the document.

The text representation dictionary usually consists of all the words that occur in the training corpus. Before mapping text document into a feature vector there are two things that need be done.

First, high frequency words (so-called stop words) such as pronouns, articles, and conjunctions, etc. need to be taken out of the text representation dictionary, as most scholars believe that retaining these topic-neutral words will be harmful for text classification. But in the view of a few opponents, the words in the stop-list are much less than those in the text representation dictionary. Thus, eliminating them from the dictionary has no significant influence on the effectiveness of a text classifier.

Second, the words in the dictionary that share the same morphological root may be merged. Supporters for *word stemming* argue that it can not only reduce the dimensionality of feature space but also be helpful to promote the effectiveness of a text classifier. But some experimental results showed that stemming sometimes might be harmful to the effectiveness of a text classifier [12].

Thus there exist at least two factors or variables that may influence the text representation dictionary. Factor A is "stop words removal". It has two experimental levels, which correspond to performing or not performing *stop words removal* in generating the text representation dictionary respectively. Factor B is "word stemming". It has also two experimental levels: performing *word stemming* or not performing *word stemming*.

In addition to these two factors the choice of feature selection algorithms and the number of selected features may influence the dictionary too. Since the influences of feature selection algorithms on the effectiveness of text classifiers are not the focus of this paper, *Mutual Information*[1] criterion, which is the most commonly used and often most effective method for selecting features [13], will be used in the following experiments.

[1] In some text categorization literature *Mutual Information* is named as *Information Gain*.

The second phase can be further divided into three steps.

In step 1, a text document is transformed into an index vector.

$$\mathrm{tf} = (\mathrm{tf}_1, \mathrm{tf}_2, \ldots, \mathrm{tf}_n) \tag{1}$$

Here, n is the number of words in the text representation dictionary, and tf_i is the index value of the ith word in the dictionary. While many authors use the *term frequency* of a word as its index value, a few scholars prefer to binary values. In the latter case tf_i is 1, if the ith word is present in the document, and 0, if it is absent.

In step 2, a weight vector, $w = (w_1, w_2, \ldots, w_n)$, is calculated. The item w_i is the relative importance of the ith word in the text representation dictionary. There are mainly two kinds of weighting schemes. The first one is the popular *inverse document frequency* (*idf*) scheme, in which w_i is defined as

$$w_i = \log \frac{N}{\mathrm{df}_i} \tag{2}$$

Here, N denotes the number of documents in the training corpus, and df_i denotes the *document frequency* of the ith word.

Another weighting scheme is *uniform weight*, in which each w_i is equal to 1.

In step 3, an index vector is combined with a weight vector to form a feature vector.The feature vector is defined as

$$fv = (w_1 \cdot \mathrm{tf}_1, w_2 \cdot \mathrm{tf}_2, \ldots, w_n \cdot \mathrm{tf}_n) \tag{3}$$

To abstract from different document lengths, many researchers suggest normalizing each document feature vector to unit length by the following formula:

$$fv = \frac{(w_1 \cdot \mathrm{tf}_1, w_2 \cdot \mathrm{tf}_2, \ldots, w_n \cdot \mathrm{tf}_n)}{\sqrt{\sum_{i=1}^{n} (w_i \cdot \mathrm{tf}_i)^2}} \tag{4}$$

Thus there exist another three factors that may influence the text representation. All of them have two experimental levels. Factor C is "indexing". Level 1 means to index text documents with the *term frequency* and level 2 means to index them with binary value. Factor D is "weighting". Level 1 means to weight words with their *inverse document frequencies*, and level 2 means to weight them with the value 1. Factor E is "normalization". It determines whether to perform *normalization* to feature vectors before inputting them into a learning algorithm.

Table 1 is the summary of discussion in this section.

For simplicity, digit "1" and "2" are used to stand for level 1 and level 2, and the combination (1,1,1,1,1) to stand for the text representation scheme corresponding to performing *stop words removal*, applying *word stemming*, indexing with *term frequency*, scaling with *inverse document frequency*, and performing *normalization*, etc.

Table 1 The names and levels of factors that may influence text representation

	Factor A (stop words removal)	Factor B (word stemming)	Factor C (indexing)	Factor D (weighting)	Factor E (normalization)
Level 1	Performing stop words removal	Performing word stemming	Indexing with term frequency	Scaling with inverse document frequency	Performing normalization
Level 2	Not performing stop words removal	Not performing word stemming	Indexing with binary value	Uniform weighting	Not performing normalization

3 Experimental setting

3.1 Classifier

Compared to state-of-the-art methods, Support Vector Machines have showed superb effectiveness on text categorization [4–7]. Moreover text classification problems are generally linearly separable. The Linear Support Vector Machines are used as the basic classification algorithm throughout the experiments in this paper.

There are many SVM packages available on the Internet such as T. Joachims's SVM^{light}, and J.C. Platt's *SMO*, etc. The *LinearSVC* in *OSU SVM Classifier Matlab Toolbox* developed by Ma [14] has been used in the experiments in this paper. For simplicity, instead of fine tuning the only parameter C in linear SVM on a validation set, which is the punishment for misclassification, we let it take the default value 1.

3.2 Test collections and effectiveness metrics

The empirical validation is done on two test collections. The first one is the Reuters-21578 dataset [15], which was compiled by David Lewis and originally collected by the Carnegie group from the Reuters newswire in 1987. The "ModApte" split is used leading to a corpus of 9,603 training documents and 3,299 test documents. Of the 135 potential topic categories only those 90, which have at least one training sample and one test sample, are used. The text enclosed in the tag "<TEXT" and "/TEXT" in each training document is used for classification. Words occurring in the titles are not discriminated from those occurring in the bodies. The size of the text representation dictionary is 27,942. After removing function words from a list of 329 stop words (commonly used in text categorization on Reuters dataset) and performing Porter's stemmer [16] the dictionary's size is reduced to 20,419.

Since Reuters poses a multi-label problem, it is broken into 90 binary classification tasks by one-vs-rest approach [17]. One linear SVM classifier is trained for each category. Micro-average precision-recall *break-even-point* (BEP) [2] is used to measure the effectiveness of a classifier.

The second test collection is the 20 Newsgroups collection, which was first collected as a text corpus by Lang [18]. It contains 19,997 email documents evenly distributed across 20 classes. The first 800 documents in each newsgroup are used as training samples and the rest of the documents are used as test samples. By skipping all headers and UU-encoded blocks only the body of a document is used for classification. The size of the text representation dictionary is 98,225. After removing function words from a list of 526 stop words (commonly used in text categorization on 20 Newsgroups dataset) and performing Porter's stemmer the dictionary's size is reduced to 75,967. Since 20 Newsgroups is a multi-class, single-label problem, *Directed Acyclic Graph SVM* [17] and multi-class classification accuracy is used.

3.3 Feature selection

To promote the effectiveness of a classification system, a local feature selection method, *Mutual Information*, following the setup in [6] is used for Reuters-21578 dataset. All words are ranked according to their mutual information with a given category. To select a subset of r features for a category, the r words with the highest mutual information with the category are chosen. All other words will be ignored.

Unlike in Reuters, a global feature selection scheme is used in 20 Newsgroups. To select the top r features based on *Mutual Information* criteria, the mutual information of a feature with each category is calculated at first. The total goodness of the feature is the sum of all the category-dependent score. Then the r features with the highest goodness are chosen.

4 Statistical analysis of experimental results

Previous factorial experiments done by the authors show that there are strong interactions among factors for text representation. In order to discover the best text representation among all the possible candidates, completely randomized designs are realized in this paper. It means that a total of 32 experiments have to be carried out for any given feature subset on a dataset. On Reuters the top 1,000, 2,000, 3,000, 4,000, 5,000, and 6,000 features are selected, respectively. On 20 Newsgroups the top 5,000, 7,500, 10,000, 20,000, and 30,000 features are selected, respectively.

All the experimental results on Reuters and Newsgroups are listed in the Appendix. All 32 experimental results on each feature set constitute one observation. Thus, we have six observations on Reuters and five

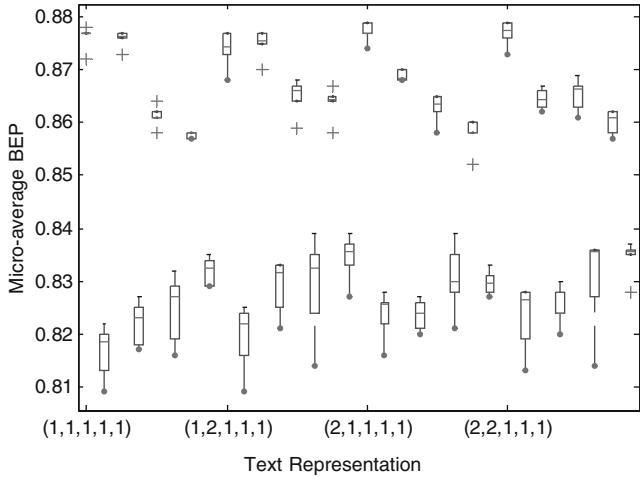

Fig. 1 Box plot of the six observations on Reuters

observations on Newsgroups. Furthermore, all observations on the same datasets are approximately independently identical.

The box plot (implemented by using the function BOXPLOT in Matlab) of the six observations on Reuters is visualized in Fig. 1. It can be found that the following six text representation schemes are the best ones: (1,1,1,1,1), (1,1,1,2,1), (1,2,1,1,1), (1,2,1,2,1), (2,1,1,1,1), and (2,2,1,1,1).

The common ground of the six best representation schemes is that the factors C and E both take the level 1, i.e., indexing text documents with term frequency and normalizing lengths of text documents with 2-norm.

The multi-way analysis of variance (implemented by using the function ANOVAN in Matlab) on the six observations is illustrated in Fig. 2.

The sixth column contains the p-value for the null hypotheses on the main effects and interactions. For example, the first element in that column (0.7668) is the p-value for the null hypothesis, H_{0A}, that samples at all levels of factor A are drawn from the same population. Small p-values lead to rejecting the associated null hypotheses.

Since the p-values of factors E, C, and B are very small (less than 0.01), the main effects of ''normalization'', ''indexing'', and ''word stemming'' are all statistically significant. For similar reasons the interactions of factors C (indexing) and E (normalization), factors D (weighting) and E, factors A (stop words removal) and D, and factors A and E are all statistically significant.

Let $r(i,j,k,l,m)$ be the experimental result (micro-average BEP or multi-class classification accuracy)

Fig. 2 Analysis of variance of the six observations on Reuters

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
A	0	1	0	0.09	0.7668
B	0.00017	1	0.00017	10.01	0.0019
C	0.0002	1	0.0002	11.61	0.0008
D	0	1	0	0.19	0.6627
E	0.07983	1	0.07983	4681.78	0
A*B	0.00005	1	0.00005	3.12	0.0794
A*C	0.00001	1	0.00001	0.79	0.3741
A*D	0.00047	1	0.00047	27.31	0
A*E	0.00015	1	0.00015	8.52	0.004
B*C	0.00011	1	0.00011	6.42	0.0122
B*D	0.00001	1	0.00001	0.67	0.4126
B*E	0.00005	1	0.00005	2.64	0.106
C*D	0	1	0	0.16	0.6882
C*E	0.00465	1	0.00465	272.78	0
D*E	0.00083	1	0.00083	48.63	0
A*B*C	0	1	0	0.11	0.7403
A*B*D	0	1	0	0.16	0.6882
A*B*E	0	1	0	0.26	0.613
A*C*D	0.00008	1	0.00008	4.47	0.036
A*C*E	0.00005	1	0.00005	2.87	0.092
A*D*E	0	1	0	0.26	0.613
B*C*D	0	1	0	0.04	0.8478
B*C*E	0.00009	1	0.00009	5.24	0.0234
B*D*E	0.00002	1	0.00002	0.92	0.3379
C*D*E	0.00001	1	0.00001	0.42	0.5188
A*B*C*D	0.00003	1	0.00003	1.63	0.2039
A*B*C*E	0.00002	1	0.00002	1.21	0.2725
A*B*D*E	0.00003	1	0.00003	1.63	0.2039
A*C*D*E	0.00002	1	0.00002	1.37	0.2433
B*C*D*E	0.00001	1	0.00001	0.73	0.3931
A*B*C*D*E	0.00001	1	0.00001	0.42	0.5188
Error	0.00273	160	0.00002		
Total	0.08963	191			

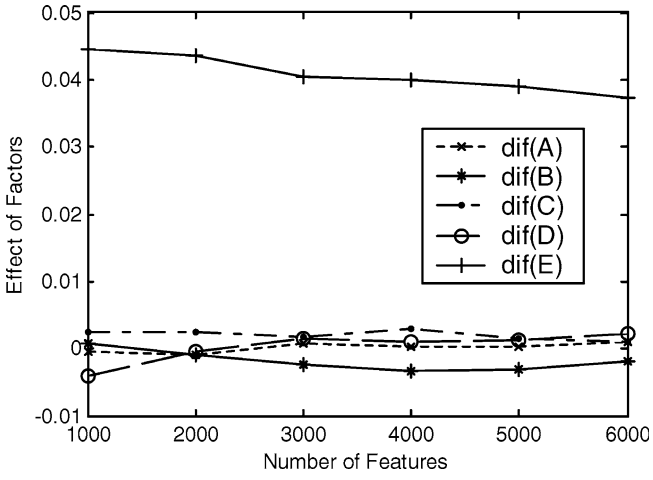

Fig. 3 Effects of the five factors on Reuters

under the text representation scheme (i,j,k,l,m). Then, the average experimental result for level i of factor A, $r_A(i)$, can be calculated as

$$r_A(i) = \frac{1}{16} \sum_{j,k,l,m=1}^{2} r(i,j,k,l,m). \quad (5)$$

And the effect of factor A, dif(A), can be calculated as

$$\mathrm{dif}(A) = r_A(1) - r_A(2). \quad (6)$$

The definitions of dif(B), dif(C), dif(D), and dif(E) are similar.

Figure 3 illustrates the effects of various factors on the effectiveness of text representation schemes for Reuters.

From Fig. 3, we find that, in view of classification accuracy, performing normalization promotes the effectiveness of text classifiers the most; indexing text documents with term frequency can also promote the effectiveness of text classifiers; performing word stemming is a bit harmful for text categorization.

The box plot of the five observations on 20 Newsgroups is visualized in Fig. 4. It can be found that the following eight text representation schemes are the best ones: (1,1,1,1,1), (1,1,2,1,1), (1,2,1,1,1), (1,2,2,1,1), (2,1,1,1,1), (2,1,2,1,1), (2,2,1,1,1), and (2,2,2,1,1).

The common ground of the eight best representation schemes is that the Factors D and E both take the level 1, i.e., weighting indexed scores with inverse document frequency and normalizing lengths of text documents with 2-norm.

The multi-way analysis of variance on the five observations is illustrated in Fig. 5.

From Fig. 5 we find that the main effects of factors E, C, A, D, and B are all statistically significant. And the interactions of factors D (weighting) and E (normalization), factors A (stop words removal) and D, factors C (indexing) and E, factors A and C, and factors C and D, are also statistically significant.

Figure 6 illustrates the effects of various factors on the effectiveness of text representation schemes for 20 Newsgroups.

From Fig. 6, we find that, in view of classification accuracy, performing normalization promotes the effectiveness of text classifier the most again; indexing text documents with term frequency is harmful for text categorization; removal of stop words, weighting indexed scores with inverse document frequency, and performing word stemming can also promote the effectiveness of text classifiers.

Fig. 4 Box plot of the five observations on 20 Newsgroups

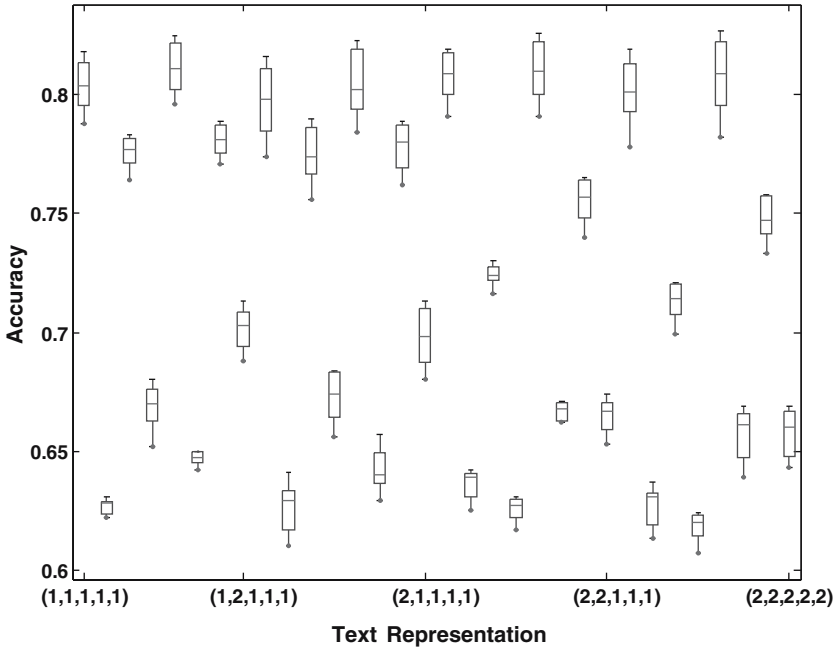

Fig. 5 Analysis of variance of the five observations on 20 Newsgroups

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
A	0.01318	1	0.01318	111.07	0
B	0.00109	1	0.00109	9.2	0.0029
C	0.01731	1	0.01731	145.87	0
D	0.00726	1	0.00726	61.22	0
E	0.66281	1	0.66281	5586.96	0
A*B	0.00025	1	0.00025	2.11	0.149
A*C	0.00145	1	0.00145	12.24	0.0006
A*D	0.0248	1	0.0248	209.05	0
A*E	0.00013	1	0.00013	1.09	0.2979
B*C	0.00001	1	0.00001	0.05	0.8167
B*D	0.00002	1	0.00002	0.13	0.7173
B*E	0.00001	1	0.00001	0.06	0.8055
C*D	0.00123	1	0.00123	10.39	0.0016
C*E	0.00289	1	0.00289	24.36	0
D*E	0.05134	1	0.05134	432.73	0
A*B*C	0.00007	1	0.00007	0.55	0.4604
A*B*D	0.00007	1	0.00007	0.61	0.4345
A*B*E	0.00006	1	0.00006	0.49	0.4872
A*C*D	0.00057	1	0.00057	4.8	0.0302
A*C*E	0.00001	1	0.00001	0.06	0.8055
A*D*E	0.00024	1	0.00024	2.02	0.1573
B*C*D	0	1	0	0.04	0.8393
B*C*E	0.00005	1	0.00005	0.45	0.5055
B*D*E	0.00002	1	0.00002	0.15	0.6957
C*D*E	0.00002	1	0.00002	0.19	0.6639
A*B*C*D	0.00001	1	0.00001	0.06	0.8055
A*B*C*E	0	1	0	0	0.9653
A*B*D*E	0.00005	1	0.00005	0.45	0.5055
A*C*D*E	0.00073	1	0.00073	6.16	0.0143
B*C*D*E	0	1	0	0	1
A*B*C*D*E	0	1	0	0.01	0.9423
Error	0.01519	128	0.00012		
Total	0.80085	159			

5 Discussions

An interesting fact is that performing normalization can always promote the effectiveness of text classifiers greatly. The possible reason is that the classifier used in this paper is a linear support vector machine. It is well known that a linear support vector machine tries to determine the optimal separating hyperplane by maximizing the Euclidean distance (the margin) between the two opposite sample sets. Normalizing the feature vector with 2-norm can eliminate the noise introduced by the length of the text document, and thus be of great help in locating the optimal separating hyperplane properly. As a matter of fact, our experimental results demonstrate that the effectiveness promotion is marginal when feature vectors are normalized with 1-norm or ∞-norm.

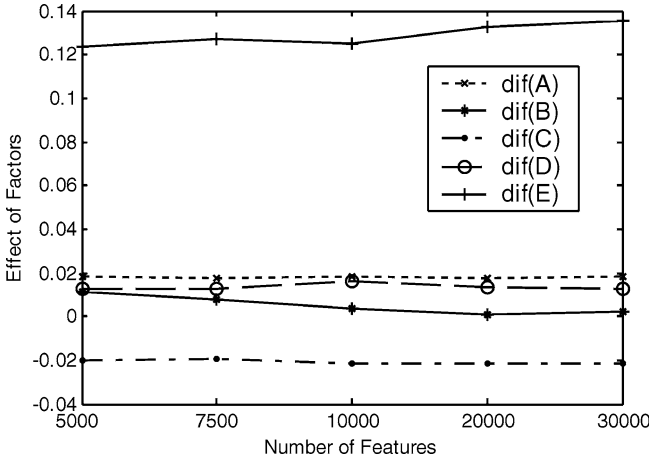

Fig. 6 Effects of the five factors on 20 Newsgroups

The effect of normalization on 20 Newsgroups is much larger that that on Reuters. The probable reason is that the variation in length of the text document in 20 Newsgroups is much larger than that in Reuters. As a matter of fact the coefficient of variation for the text length in training samples in 20 Newsgroups is 241.01% whereas the coefficient of variation in Reuters is 100.70%. The coefficient of variation of data x is the ratio of standard deviation of x to the mean of x.

Another concern is the effect of indexing. On Reuters the contribution of term frequency is marginal (average 0.0020 micro-average BEP) whereas on 20 Newsgroups indexing text documents with term frequency is definitely harmful. Though term frequency (tf) is the prevailing indexing approach in text categorization, one might think that a richer indexing approach such as $\log(\mathrm{tf}+1)$ is probably more effective. To check the effectiveness of $\log(\mathrm{tf}+1)$ we performed two more experiments. All experimental setups are the same as

Table 2 The micro-average BEPs of different text representations for reuters with the top 3,000 features

bep(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.879	0.864	0.880	0.865	0.882	0.864	0.881	0.866
(1,2)	0.838	0.828	0.842	0.834	0.842	0.833	0.844	0.838
(2,1)	0.882	0.858	0.881	0.865	0.878	0.860	0.875	0.860
(2,2)	0.853	0.837	0.850	0.842	0.850	0.836	0.847	0.840

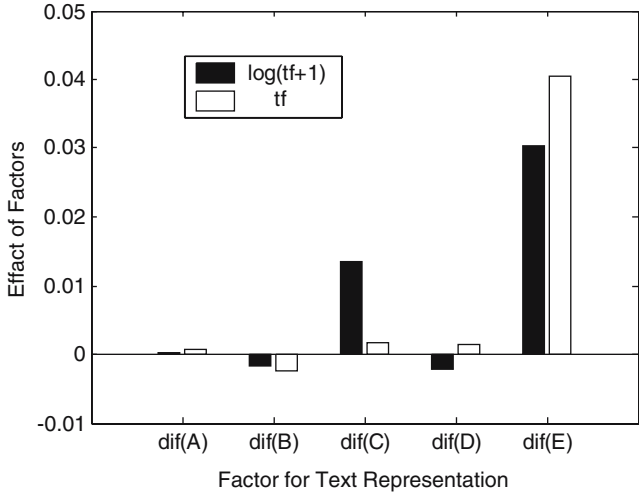

Fig. 7 Comparison of effects of five text representation factors on Reuters when the level 1 of factor C is tf and log(tf + 1), respectively

before except that the level 1 of factor C (indexing) is no longer tf (term frequency) but log(tf + 1).

The 32 experimental results on Reuters with the top 3,000 features are listed in Table 2.

The effects of the five factors are illustrated in Fig. 7.

From Fig. 7 we find that log(tf + 1) does do better than tf (term frequency) when used in indexing text documents. It not only contributes more to the effectiveness of the text classifiers by itself, but also lessens the impact of other factors.

We can find a similar situation on the 20 Newsgroups dataset.

The 32 experimental results on 20 Newsgroups with the top 5,000 features are listed in Table 3.

The effects of the five factors are illustrated in Fig. 8.

From Fig. 8 we find that while lessening the impacts of other factors, indexing text documents with log(tf + 1) is almost as effective as indexing with binary values.

6 Conclusions

Based on the statistical analyses of experimental results and discussions in the previous section, the following conclusions can be drawn with some confidence.

– There are strong interactions between text representation factors. And the best text representation schemes are corpus-dependent.

For Reuters, indexing text documents with term frequency and normalizing feature vectors before submitting them to learning algorithms are two necessary conditions for text classifiers to achieve the best effectiveness. In addition to those, weighting the indexed scores with inverse document frequency, or weighting with a constant value but removing stop words at the same time is also required.

For 20 Newsgroups, weighting the indexed scores with inverse document frequency and normalizing the feature vectors before submitting to learning algorithms are two sufficient and necessary conditions for text classifiers to achieve the best effectiveness.

– Among the five factors that may affect a text representation "normalization" is the most important. Normalizing feature vectors before submitting them to learning algorithms can always promote effectiveness of text classifiers greatly. In contrast, the impacts of other factors such as "stop words removal", "word stemming", "indexing" and "weighting" are small.
– Removing stop words from the vocabulary is not harmful if it is not helpful, no matter in view of

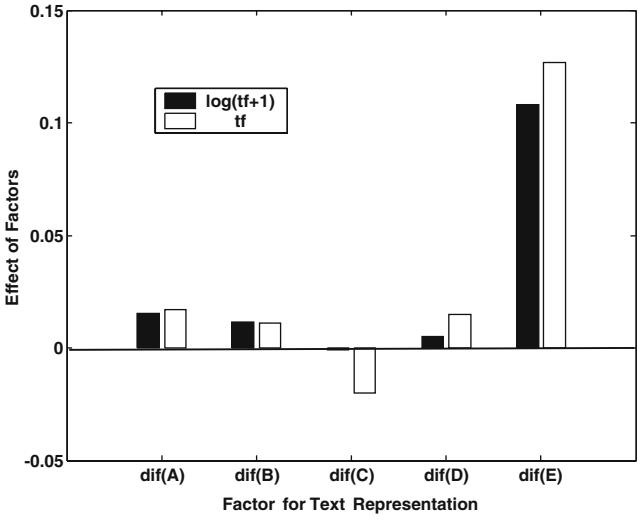

Fig. 8 Comparison of effects of five text representation factors on 20 Newsgroups when the level 1 of factor C is tf and log(tf + 1), respectively

Table 3 The multi-label accuracies of different text representations for 20 newsgroups with the top 5,000 features

bep(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.800	0.796	0.790	0.784	0.801	0.791	0.788	0.782
(1,2)	0.647	0.655	0.635	0.641	0.664	0.673	0.644	0.648
(2,1)	0.777	0.771	0.767	0.762	0.747	0.740	0.733	0.733
(2,2)	0.699	0.703	0.698	0.696	0.657	0.669	0.644	0.660

classification effectiveness or in view of classification efficiency.

- Though it is a very simple binary-value-indexing scheme it is sometimes a good choice such as in 20 Newsgroups.
- In view of the classification effectiveness there is no definite conclusion about "word stemming". On Reuters it is harmful whereas on 20 Newsgroups it is helpful. Considering the fact that the effect of "word stemming" is very small and it can reduce dimensionality of text feature space greatly, we suggest using word stemming in text categorization to promote the efficiency of the text classification system.
- When representing a text document, it will possibly be the best choice to perform stop word removal and word stemming, to index with a function of term frequency (such as $\log(\mathrm{tf}+1)$), to scale with inverse document frequency, and to normalize the feature vector with 2-norm before submitting it to a learning algorithm.

Acknowledgements The authors would like to thank the anonymous reviewers for their precious suggestions.

Appendix

Table 4 The micro-average break-even points of different text representations for reuters-21578 when the top 1,000 features are selected

bep(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.872	0.809	0.873	0.817	0.858	0.816	0.858	0.829
(1,2)	0.868	0.809	0.870	0.821	0.859	0.814	0.858	0.833
(2,1)	0.874	0.816	0.869	0.820	0.858	0.821	0.852	0.831
(2,2)	0.873	0.813	0.864	0.820	0.861	0.814	0.857	0.828

Table 5 The micro-average break-even points of different text representations for reuters-21578 when the top 2,000 features are selected

bep(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.877	0.813	0.876	0.818	0.861	0.819	0.857	0.829
(1,2)	0.873	0.816	0.875	0.825	0.864	0.824	0.864	0.827
(2,1)	0.877	0.822	0.870	0.821	0.862	0.828	0.858	0.829
(2,2)	0.876	0.819	0.862	0.824	0.863	0.827	0.858	0.835

Table 6 The micro-average break-even points of different text representations for Reuters-21578 when the top 3,000 features are selected

bep(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.878	0.818	0.877	0.823	0.864	0.827	0.858	0.832
(1,2)	0.877	0.821	0.875	0.832	0.865	0.832	0.865	0.835
(2,1)	0.877	0.825	0.868	0.824	0.864	0.830	0.860	0.828
(2,2)	0.879	0.826	0.863	0.824	0.866	0.836	0.860	0.837

Table 7 The micro-average break-even points of different text representations for reuters-21578 when the top 4,000 features are selected

bep(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.877	0.819	0.876	0.825	0.862	0.827	0.857	0.833
(1,2)	0.877	0.823	0.877	0.831	0.867	0.833	0.865	0.837
(2,1)	0.879	0.826	0.870	0.826	0.865	0.830	0.858	0.827
(2,2)	0.879	0.827	0.865	0.830	0.867	0.835	0.862	0.836

Table 8 The micro-average break-even points of different text representations for reuters-21578 when the top 5,000 features are selected

bep(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.877	0.820	0.877	0.823	0.861	0.829	0.858	0.834
(1,2)	0.874	0.824	0.876	0.833	0.868	0.835	0.867	0.836
(2,1)	0.879	0.826	0.868	0.824	0.863	0.835	0.860	0.830
(2,2)	0.878	0.828	0.867	0.828	0.867	0.836	0.862	0.836

Table 9 The micro-average break-even points of different text representations for reuters-21578 when the top 6,000 features are selected

bep(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.877	0.822	0.877	0.827	0.861	0.832	0.858	0.835
(1,2)	0.875	0.825	0.877	0.833	0.867	0.839	0.864	0.839
(2,1)	0.877	0.828	0.868	0.827	0.865	0.839	0.858	0.833
(2,2)	0.877	0.828	0.866	0.824	0.869	0.836	0.862	0.835

Table 10 The multi-label accuracy of different text representations for 20 newsgroups when the top 5,000 features are selected

acc(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.788	0.628	0.764	0.652	0.796	0.647	0.771	0.688
(1,2)	0.774	0.610	0.756	0.656	0.784	0.629	0.762	0.680
(2,1)	0.791	0.625	0.716	0.617	0.791	0.662	0.740	0.653
(2,2)	0.778	0.613	0.699	0.607	0.782	0.639	0.733	0.643

Table 11 The multi-label accuracy of different text representations for 20 newsgroups when the top 7,500 features are selected

acc(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.798	0.631	0.774	0.666	0.804	0.650	0.777	0.696
(1,2)	0.788	0.619	0.770	0.667	0.797	0.640	0.772	0.690
(2,1)	0.803	0.633	0.724	0.624	0.803	0.668	0.751	0.661
(2,2)	0.798	0.621	0.710	0.620	0.800	0.650	0.744	0.649

Table 12 The multi-label accuracy of different text representations for 20 newsgroups when the top 10,000 features are selected

acc(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.804	0.628	0.777	0.670	0.811	0.650	0.781	0.703
(1,2)	0.798	0.641	0.774	0.674	0.802	0.657	0.780	0.698
(2,1)	0.809	0.639	0.724	0.627	0.810	0.671	0.757	0.667
(2,2)	0.801	0.631	0.714	0.617	0.809	0.669	0.747	0.660

Table 13 The multi-label accuracy of different text representations for 20 newsgroups when the top 20,000 features are selected

acc(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.812	0.622	0.781	0.675	0.821	0.642	0.787	0.707
(1,2)	0.809	0.631	0.785	0.683	0.818	0.647	0.787	0.709
(2,1)	0.817	0.642	0.727	0.629	0.821	0.663	0.764	0.669
(2,2)	0.811	0.637	0.720	0.623	0.821	0.665	0.757	0.666

Table 14 The multi-label accuracy of different text representations for 20 newsgroups when the top 30,000 features are selected

acc(*i*,*j*,*k*,*l*,*m*)	Factors C, D, and E							
	(1,1,1)	(1,1,2)	(1,2,1)	(1,2,2)	(2,1,1)	(2,1,2)	(2,2,1)	(2,2,2)
Factor A and B								
(1,1)	0.818	0.624	0.783	0.680	0.825	0.646	0.789	0.713
(1,2)	0.816	0.629	0.790	0.684	0.823	0.639	0.789	0.713
(2,1)	0.819	0.640	0.730	0.631	0.826	0.670	0.765	0.674
(2,2)	0.819	0.631	0.721	0.624	0.827	0.661	0.758	0.669

7 Originality and contribution

It is well known that the effectiveness of the text categorization system is not simply a matter of learning algorithms. Text representation factors are also at work. Though a lot of text representation modes other than the *bag of words*, such as *statistical phrase-based* representation and *ngram-based* representation, have been examined previously without much success, variants of the *bag of words* and their effectiveness have not been studied systematically as known by the authors. There are still some questions left behind without answers:

- Among the possible variants of the *bag of words* schemes which ones are probably the best?
- Among the factors that may affect a text representation which ones are the most important and should be dealt with seriously?
- Is "stop words removal" an indispensable step to represent a text document?
- Does indexing a text document with term frequency always outperform indexing it with binary value?
- Whether "word stemming" is harmful or beneficial for text categorization?
- How should we represent text document the best?

By extensive experiments on two benchmark dataset Reuters-21578 and 20 Newsgroups and thorough statistical analyses of those results all of the above questions have been answered in this paper with some confidence.

The main contribution of this paper is that it clarifies some blurred cognition on text representation such that text representation can be more effective and efficient.

References

1. Maron M (1961) Automatic indexing: an experimental inquiry. J Assoc Comput Mach 8(3):404–417
2. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47
3. Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans PAMI 22(1):4–37
4. Yang Y (1999) An evaluation of statistical approaches to text categorization. Inf Retrieval 1(2):69–90
5. Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. Proceedings of the 10th European Conference on Machine Learning (ECML). Springer, Berlin Heidelberg New York
6. Dumais S, Platt J, Heckerman D, Sahami M (1998) Inductive learning algorithms and representations for text categorization. Proceedings of the CIKM-98, Seventh ACM International Conference on Information and Knowledge Management, pp 148–155

7. Yang Y, Liu X (1999) A re-evaluation of text categorization methods. Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, pp 42–49
8. Zhang T, Oles FJ (2001) Text categorization based on regularized linear classification methods. Inf Retrieval 4:5–31
9. Chakrabarti S, Roy S, Soundalgekar MV, Bombay I (2002) Fast and accuracy text classification via multiple linear discriminant projections. Proceedings of the 28th VLDB Conference, Hong Kong, China
10. Petridis V, Kaburlasos VG, Fragkou P, Kehagias A (2001) Text classification using the -FLNMAP neural network. Proceedings of the 2001 International Joint Conference on Neural Networks (IJCNN2001)
11. Salton G, Wong A, Yang C (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620
12. Baker LD, McCallum AK (1998) Distributional clustering of words for text categorisation. Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval, Melbourne, Australia, pp 96–103
13. Yang Y Pedersen JO (1997) A comparative study on feature selection in text categorization. In: Machine learning, Proceedings of the 14th International Conference (ICML'97), pp 412–420
14. Ma J, Zhao Y Ahalt S OSU SVM Classifier Matlab Toolbox (ver 3.00). Available at: http://www.eng.ohio-state.edu/~maj/osu_svm/
15. Porter MF (1980) An algorithm for suffix striping, *Program*, vol 14, no. 3, pp 130–137
16. Lewis, Reuters-21578, Distribution 1.0. Available at: http://www.research.att.com/~lewis/reuters21578.html
17. Hsu C, Lin C (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13(2)
18. Lang K (1995) Newsweeder: learning to filter netnews. Proceeding of the Twelfth International Conference on Machine Learning, pp 331–339
19. Schutze H, Hull DA, Pedersen JO (1995) A comparison of classifiers and document representations for the routing problem. Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval, pp 229–23