JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 29, 99-114 (2013)

A Text Categorization Method using Extended Vector Space Model by Frequent Term Sets*

MAN YUAN[1,2], YUAN XIN OUYANG[1,2,+] AND ZHANG XIONG[1,2]
[1]*School of Computer Science and Technology*
Beihang University
Beijing, 100191 P.R. China
[2]*Research Institute of Beihang University in Shenzhen*
VU Park, High-tech Industrial Estate
Shenzhen, 518057 P.R. China

Text categorization is one of the most important research topics in Natural Language Processing and Information Retrieval due to the ever-increasing electronic documents. This paper presents a new text categorization method using frequent term sets. A novel constraint measure AD-Sup was introduced to extract discriminative features from frequent term sets for classification task. Then text documents are represented in the global feature space which contains both single terms and frequent term sets. To solve the sparse instance problem, a term weighting strategy is then implemented which assigns estimated weights using feature similarity and highly reduces the sparse rate. Through extensive experiments, the optimal proportion of single features and frequent term set features is empirically determined. Classification results on Reuters-21578 and WebKB corpus demonstrate that AD-Sup constraint is effective to extract useful frequent features and the combination strategy is effective to build better feature space and improve the SVM classifier.

Keywords: text categorization, text representation, frequent term sets, Apriori, SVM

1. INTRODUCTION

Text categorization (TC) is one of the main themes in text mining. In the past decades, due to the increasing need to manage and process the explosively growing online and digital text data, text categorization has gained special attentions from researchers. By assigning unlabelled natural language documents into a predefined set of categories, text categorization is beneficial for many applications, such as information retrieval, web pages classification, email filtering, and information management systems. A large number of machine learning and probabilistic based methods have been proposed for text categorization to build classifiers. The most popular methods include Naïve Bayes, decision tree, decision rules, association rules, Rocchio method, neural networks, and support vector machines (SVM) [1].

Text mining or Kowledge discovery in text data mainly focuses on the automation process and the efficient performance. However, computer systems still rely on human users for cognitive functions such as decision-making, planning, and creative thinking in order to avoid expensive failure and guarantee the results useful and relevant. In recent

Received May 31, 2011; accepted March 31, 2012.
Communicated by Francisco J. García-Peñalvo, Ricardo Colomo-Palacios and Jane Yung-Jen Hsu.
* This work was supported by National Nature Science Foundation of China (No. 60972145 and No. 60803072), Fundamental Research Funds for the Central Universities (No. 2009JBM024), and China Postdoctor Research Foundation (20090460197).
+ Corresponding author.

years, the Internet and its extension on mobile network, ubiquitous devices and interfaces make the integration between human and computer much more tightened. The importance of human-computer interaction (HCI) has been well studied in many information disciplines, such as interactive data mining, visual data mining, interactive information retrieval, interactive text mining etc. Interactive Text Mining [2] allows the user and the text mining system to interact with each other in data mining tasks. In this procedure, the participation of users can encourage learning and achieve the most needed results and in turn, the user's feedback can help to improve the system.

One of the key issues in Interactive Knowledge Discovery is the information acquisition for user which aims to provide knowledge that is easy to interpret and understand with natural semantic meanings [3]. However, in most of the TC methods, the representation of documents is based on vector space model (VSM) proposed by Salton in 1975 [1]. VSM is widely used in text mining because of its conceptual simplicity and the appeal of the underlying metaphor of using spatial proximity for semantic proximity. In VSM, the text contents are treated as "bag of words" and every document is represented as a vector of features where each feature corresponds to a unique word from the documents. Term weighting methods are then employed to assign values on each word to express the importance of each feature. Since the "bag of words" method ignores the combinations and orders of terms, an obvious drawback of this strategy is that it is not sufficient to preserve the semantic and syntactic information. To deal with this problem, some researchers attempted to construct complex feature unit like base forms of morphological categories, phrases, word senses [4] and multi-word [5] to substitute the single words by Natural Language Processing approach. Although these NLP based method apparently carry more information than bag of words, they can only gain small or no improvement for TC tasks [6].

Frequent itemsets originate from data mining where frequent itemsets are used to find association rules. A frequent itemset refers to a set of items in which the co-occurrence of these items is more than a threshold value called minimum support. Since the frequent itemsets reflect strong associations between items and includes more underlying semantic and contextual meaning than individual word, it has been developed within the text mining domain in different aspects, including: association classifier for text categorization [7, 8], frequent pattern based text clustering [9-11], frequent pattern for text classification [12] *etc*. These previous works have validated the value of frequent itemsets in text mining, but there is still a lack of specified exploration about the application of frequent itemsets on text categorization.

This paper proposes a new approach for text categorization using frequent term set. The motivation is to improve classification effect and at the mean time, provide a new way to represent text data and explore the latent term associations for better data description on the user end. The main contributions of this work involve: (1) A novel restraint measure AD-Sup is defined to generate discriminative frequent term sets for classification problem. (2) The term weighting process is conducted under a new strategy which assigns estimated weights using feature similarity and highly reduces the sparse rate. (3) Text document is represented with a combination scheme in which the feature space contains both single words and frequent term sets and the ratio of the two were investigated through extensive experiment. (4) The proposed method could run well with popular text categorization algorithm, hence it is also easy to extend to and improve other

classifiers.

The rest of this paper is organized as follows. Section 2 gives a brief review of previous works on text categorization and frequent patterns in text mining. Section 3 describes a novel method of frequent term sets based text categorization in details. Extensive experimental evaluation on real text data are discussed in section 4. Section 5 concludes the paper and presents some directions for future work.

2. RELATED WORKS

One of the crucial issues in Interactive Knowledge Discovery is the information acquisition for users. The human cooperation characteristic requires that interactive system must provide knowledge that is easy to interpret and understand with natural semantic information [3]. However, in text categorization field, most of the standard classifiers rely on the vector space model (VSM) in which each document is represented as a vector in the feature (term) space [1]. In traditional methods, these features are selected from so called "bag of words" in document and each feature is a single word without considering the combination or order of words. An obvious defect of this scheme is that it ignores the original semantics of text content which exist in the complex syntax and grammar forms. This will lead to loss of information and homonymy problem.

There have been attempts to construct complex features at various levels, like phrases, word senses [4], multi-word [5], and temporal sequences [13] to substitute the single words by Natural Language Processing (NLP) or statistic approach. [6] gave a comparative study on phrases, word senses and syntactic relations, and the result showed although these NLP based methods apparently carry more information than bag of words, they can only gain small or no improvement for TC tasks.

Another kind of approaches is the employment of frequent itemsets which originate from data mining where frequent itemsets are used to find association rules. A frequent itemset refers to a set of items in which the co-occurrence of these items is more than a threshold value called minimum support. Because frequent itemsets reflect strong associations between items, it is naturally expected to contain more underlying semantic and contextual meaning than individual word for text mining. Under this motivation, frequent itemsets have been adopted in the text mining domain in various ways. [7] firstly introduced association rule mining technique into classification problem in which the classifier is built on a subset of association rules called "class association rules" or "classification rules" and frequent itemsets form the basis for discovering these association rules. [8] then developed this approach in a more specific way and applied it in classifying text documents. [14] discussed the problem of mining association rules form textual document. In [15] there are more reports on the advances of associative classification technique.

Besides association rules, the frequent patterns also have been explored and proved to be helpful to obtain competitive performance for text categorization and clustering. Frequent patterns in text mining issues [16] can be frequent sequences or frequent itemsets, the difference lies in if the sequential orders of words are considered. [12] analyzed the frequent patterns for text classification problem and proposed a strategy to set minimum support (*min-sup*) by establishing a connection with feature selection approach. H.

Ahonen-Myka [17] proposed the first algorithm to find maximal frequent sequence for text document. A maximal frequent sequence (MFS) is a sequence that is not contained or subsequence in other frequent sequence. Consequently, the collection of MFS's can be a compact representation for the original term set. In [9], E. Hernández-Reyes applied MFS in text clustering where each MFS of words were used for text representation and each MFS correspond to a feature of text document in vector space model (VSM). Then *k*-means algorithm is employed to group document into clusters. Other text clustering methods based on frequent patterns involve CFWS [11], FTC [18], and MC [10]. Instead of using frequent patterns for text representation, these methods adopt frequent sequence or itemsets in the clustering phase.

3. FREQUENT TERM BASED TEXT CATEGORIZATION

This section describes a new approach for text categorization using frequent term sets. The proposed method includes four steps: (1) Data pre-processing and feature selection; (2) Frequent term set extraction; (3) Text representation using frequent term sets; (4) classifier model learning and classifying.

3.1 Data Pre-processing and Feature Selection

The Data pre-processing and feature selection step is an essential procedure for most of text processing issues because of the high dimensionality of the natural language text which makes the text data quite noisy and sparse in vector space. Text data pre-processing mostly involves stemming and stop-word removal. Feature selection is one of the dimension reduction approaches that can significantly decrease the computational cost of text categorization and, at the same time, preserve or even increase the classification performance. In this paper, Information Gain (IG) [1] is implemented for feature selection. IG has been proved to be one of the best feature selection methods for text categorization [19]. It measures the decrease in entropy between the feature is present or absent. Let $\{C_i\}_{i=1}^m$ denote the set of categories, the IG of term t is defined as:

$$\begin{aligned} IG(t) = & -\sum_{i=1}^{m} P(c_i)\log P(c_i) \\ & + P(t)\sum_{i=1}^{m} P(c_i \mid t)\log P(c_i \mid t) \\ & + P(\overline{t})\sum_{i=1}^{m} P(c_i \mid \overline{t})\log P(c_i \mid \overline{t}). \end{aligned} \tag{1}$$

Before extracting frequent term set, the original terms are selected according to its IG score. In this study, the magnitude of dimensions is reduced from 10^5 to 10^2. An important reason that makes feature selection an essential procedure is that the number of frequent term sets extracted from single terms grows exponentially and so the base number of single terms must be restricted to a reasonable scale.

3.2 Frequent Term Set Extraction

Efficient mining of frequent itemsets is a fundamental problem for mining association rules. The original description of association rule mining [20] is as follows: Let I =

$\{i_1, i_2, \ldots, i_n\}$ be a set of items and $T = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions in which each transaction contains a subset of the items in I. If itemset $X \subseteq I$, the number of transactions in T that contain X is $Count(X)$ and the total number of transactions is n, then the support of X is $Sup(X) = Count(X)/n$. An itemset X is called frequent if its support is greater than or equal to a given percentage s which is so called the minimum support (*min-sup*).

In text mining issues, each document d in $D = \{d_1, d_2, \ldots, d_n\}$ is treated as a transaction and the set of terms $T = \{t_1, t_2, \ldots, t_m\}$ contained in D corresponds to the items set I. A term set S in T is frequent if $Sup(S) \geq min\text{-}sup$. The *min-sup* constraint of term set is a key measure for frequent sets extraction because it determines the scale and quality of the selected frequent sets. This deserves more consideration for text document because comparing with the classical market blanket analysis, the amount of terms in a document is usually much larger than the items in a transaction. The large number of terms will sometimes lead to the exponential growth of frequent term set candidates and make the result unreachable.

Definition (*AD-Sup*): When applied to text mining problem, the concept of support count corresponds to Document Frequency (DF). It can be deduced that the support count of a term set is the minimum DF of all the terms in set. However, in classification task, support count cannot be simply substituted by DF because DF only measures the occurrences and this is not sufficient to differentiate the discriminative effect of the frequent term sets. Considering the distribution discrepancy of term sets in each document class, we propose a new restraint measure for text categorization referred as Average Deviation Support (*AD-Sup*). Assume the documents set have n classes $\{class_1, \ldots, class_i, \ldots, class_n\}$ and let FS denote the term set and t is the term in FS, *AD-Sup* can be formulated as:

$$AD\text{-}Sup(FS) = \frac{\sqrt{\sum_{i=1}^{n} \{Sup(FS)_i - Ave(Sup(FS))\}^2}}{Ave(Sup(FS))}, \tag{2}$$

$$Ave(Sup(FS)) = \frac{\sum_{i=1}^{n} Sup(FS)_i}{n}, \tag{3}$$

$$Sup(FS)_i = \min\{df(t)_1 \ldots df(t)_m\}. \tag{4}$$

The expression of *AD-Sup* (1) can be deemed as a modified support deviation, where $Sup(FS)_i$ means the local support of FS in class i and $Ave(Sup(FS))$ denotes the average value of $Sup(FS)$ in all the classes. Additionally, analysis on real data shows that when a term set has a large average support, even if it's distributed quite evenly, sometimes its standard support deviation may still surpass that of the term sets which occur comparatively less but are more distinctive in different classes. However, when a term set has very close support value in different classes, it would not be a valuable feature for classification. Hence in the *AD-Sup* equation, the standard deviation is divided by the average support to represent the deviation rate instead of the absolute deviation value.

The frequent term extraction is implemented using Apriori strategy [15]. Apriori is

one of the best known methods for association rule mining and as a breadth-first-search algorithm, it generates itemsets in a level-wise manner, where each candidate k-itemsets in the kth iteration is generated from frequent $(k-1)$-itemsets. The candidates with support $\geq$ *min-sup* will be added into the frequent set and this iteration continues until the candidate set is empty. Apriori algorithm is chosen as the extraction framework for the character that in Apriori, transactions are not stored in memory and the algorithm works by scanning the database iteratively. This strategy makes Apriori very suitable for the large count of transactions and items in text documents. After obtaining all the frequent term sets (*FS*), *AD-Sup* restraint is used to select the frequent features. The selected *FS* will involve more term sets that are not only frequent but distributed unevenly in different classes.

3.3 Text Representation and Categorization

In vector space model, text representation mainly focuses on how to build the feature vectors and how to assign weights to the elements of the vector. On the first part, we consider the procedure of extracting frequent term sets in a new perspective: the input of this iteration are the selected single words, when modeling these term sets in a unified formulation, these single words could be perceived as unit sets or 1-term sets and in this case the global feature space is composed of term sets with element numbers from 1 to n. The following problem is how to combine single terms and frequent term sets together. There are two extreme distributions: the first one is simply using the single terms and no frequent term sets, which is the traditional method and the second one is only using the frequent term sets without single terms, which has been discussed in previous literatures. In this paper, single terms and frequent term sets are taken as two indispensible parts for the feature space, and the proportions of the two are determined empirically.

On the term weighting part, one of the key issues is the sparse data problem. Sparse instance vector is an inherent problem for text categorization which means for each document, the corresponding document vector contains only few entries that are not zero. This problem will become more severe when documents are represented by frequent term sets. Because when a document is referred to contain a frequent term set, it indicates that the document contains all the terms in this term set and their co-occurring probabilistic is usually much smaller. Table 1 shows the average sparse rate (the percentage of empty weights in all features) of the documents in Reuters-21578 corpus when represented with different number of frequent term set features.

Table 1. Sparse rate on Reuters-21578.

Feature Number	600	800	1000	1200	1400	1600
Sparse Rate	96.8%	96.5%	96.4%	96.2%	96.0%	95.9%

To solve the sparse instance problem, we propose a weight estimation method using feature similarity. The key notion is that when a document contains part of a frequent term set, it has a similar feature. Accordingly, the weight of the similar feature is obtained by the cosine similarity with the original one. In this way, the similar feature is not completely ignored but the weights still differ from each other according to their similar-

ity. Let FS be the original feature, FS' be the similar feature, and W is feature weight, then the similarity is computed by the cosine similarity:

$$FeatureSimilarity(FS', FS) = \frac{\sum_{i=1}^{n} W_i' \times W_i}{\sqrt{\sum_{i=1}^{n} (W_i')^2} \times \sqrt{\sum_{i=1}^{n} (W_i)^2}}. \quad (5)$$

And the weight of FS' will be the average term frequency (TF) in FS' multiplying the feature similarity of FS' and FS as:

$$W(FS') = \frac{(\sum_{i=1}^{n} TF_i')}{n} \times FeatureSimilarity(FS', FS). \quad (6)$$

In the above two equations, the original weight of FS is always set to be (1, ..., 1) and the weight of similar feature FS' is set by term frequency. For example, assume document d_1 contains terms as $\{a, a, b, d, e\}$ and there is a frequent term set as (a, b, c). The original weight is (1,1,1) and the corresponding similar feature in d_1 which actually appears is (2,1,0). Then the weight of (a, b, c) in d_1 will be 0.775.

The categorization step is to build classifiers by learning on the training data and then evaluate on the test data. Since the proposed approach is based on vector space model, popular classifiers can work well on it. The classification implementation and results will be discussed in the following section.

4. EXPERIMENTAL RESULTS

This section presents the details of implementation, data sets, experiment configuration and results. SVM is a well developed classifier that has achieved top performance in the previous comparative studies [1]. Based on the proposed method, SVM is trained as the classifier model to compare with several benchmark algorithms.

4.1 Performance Measures

There are two basic criterions that are widely used in document categorization field: precision and recall. Precision is the proportion of returned documents that are correct targets, while recall is the proportion of correct target documents returned. For each category, let the judgment results be as shown in Table 2:

Table 2. The representation of categorization result.

Category i		Expert judgment	
		True	False
Classifier judgment	True	TP_i	FP_i
	False	FN_i	TN_i

Formally, the definitions of precision and recall are:

$$recall_i = \frac{TP_i}{TP_i + FP_i}, \tag{7}$$

$$precision_i = \frac{TP_i}{TP_i + FN_i}. \tag{8}$$

However, sometimes neither precision nor recall makes sense in isolation from each other as it is well known from the IR practice that higher levels of precision may be obtained at the price of low values of recall. Thus we utilize the F_1 measure which is widely used in text categorization [21, 22]. The formulation of F_1 is defined as:

$$F_1 = \frac{2 \times precision_i \times recall_i}{precision_i + recall_i}. \tag{9}$$

4.2 Data Sets

Reuters-21578: The Reuters-21578 data set is a widely used corpus for text categorization research. The distribution of documents in Reuters-21578 is quite skewed: some of the classes contain more than 2000 documents, while some classes contain less than 5 even no document. Hence most of researchers use a subset of the Reuters-21578 corpus. This paper uses the sub-collection R8 set which includes the top 8 categories with the largest number of documents in the training set. Finally, there are 5,485 documents in the training set and 2,189 documents in the test set.

WebKB: The documents in the WebKB are web pages collected by the World Wide Knowledge Base project of the CMU text learning group. The documents of WebKB are manually classified into seven different classes: *student*, *faculty*, *staff*, *department*, *course*, *project*, and *other*. Because the *department* and *staff* have very few contents and the class *other* has very different documents. Most of researches are conducted on the top 4 classes: *project*, *course*, *faculty* and *student*. Then the remaining corpus for training includes 2784 documents.

In this paper, both the above datasets are pre-processed by stemming and stop-word elimination. The evaluation is firstly conducted through 10-folds cross validation on the training set, which equally splits the training corpus into 10 folds and, each time, uses nine folds as the training set and the left fold as the test set. Then we also evaluated the classification results on different number of training instances.

4.3 Results and Discussion

4.3.1 Frequent term sets extraction

The frequent term sets extraction starts from the selected single terms by IG. The first scan generates frequent term sets with two terms and these double term sets are used as the input term sets for the candidate-generating algorithm. Then the iteration starts

until no candidate is selected to be frequent term set. During the extraction procedure, two parameters are very important to obtain high quality frequent features for classification: the number of single terms and *min-sup*. Single terms are the fundamental source for frequent term sets; therefore, a proper number of single terms must be set to control the quantity of input single terms. Because too few terms are not sufficient for extracting enough frequent features, while too many terms will bring low weight terms and redundant information, as well as a too large collection of frequent term sets. Similarly, *min-sup* is the threshold for the iterating extraction steps to guarantee all the selected term sets are frequent enough to be statistically meaningful.

After investigating different values of parameters, they are finally set to be the values in Table 3. Table 4 shows the first round of extraction results on the two datasets by *min-sup*, including the number of double or 2-term sets, the number of (> 2) term sets, the total number of frequent term sets and the extracting time.

Table 3. Parameters configuration.

Data set	Number of documents	Number of terms	Number of single terms	*min-sup*
Reuters-21578	5485	14560	300	200
WebKB	2784	7282	300	200

Table 4. Number of extracted frequent term sets.

Data set	Number of double-term sets	Number of (>2) term sets	Total Number of frequent term sets	Extracting Time (second)
Reuters-21578	751	7448	8199	707.39
WebKB	1043	6990	8033	660.94

Table 5. Single term and frequent term sets in Reuters-21578.

Top 10 terms	IG value	Top 10 frequent term sets	Support count	Top 10 refined frequent term sets	*AD-Sup* restraint
ct	0.460	mln/reuter	2602	prime/reuter	0.517
net	0.336	dlr/reuter	2344	japan/last	0.459
shr	0.313	year/reuter	2048	japan/japanes	0.456
qtr	0.250	inc/reuter	2030	unit/april	0.454
trade	0.204	ct/reuter	1999	trade/japanes	0.451
rev	0.195	mln/dlr	1791	reuter/secretari	0.448
oil	0.174	net/reuter	1786	market/econom	0.446
inc	0.166	mln/reuter	1743	corp/nation	0.443
rate	0.163	compani/reuter	1710	trade/minist	0.441
loss	0.149	corp/reuter	1563	trade/japan	0.439

Tables 5 and 6 report more details of selected terms and extraction results. Note that the original documents have been processed by stemming algorithm to bring variant forms of words together. Simultaneously, it also changes the form of the words and makes the stemmed terms appear to be different from the real or right ones. Comparing the top 10 list of frequent term sets ranked by support count and refined sets by *AD-Sup*

Table 6. Single term and frequent term sets in WebKB.

Top 10 terms	IG value	Top 10 frequent term sets	Support Count	Top 10 refined frequent term sets	AD-Sup restraint
professor	0.318	scienc/comput	1898	offic/comput/scienc/engin	0.559
instructor	0.265	univers/comput	1648	offic/scienc/engin	0.558
assign	0.264	univers/scienc	1546	scienc/engin/mail	0.558
research	0.226	univers/comput/scienc	1536	univers/comput/scienc /depart/home/engin	0.557
syllabu	0.217	comput/depart	1355	scienc/depart/home /engin	0.557
hour	0.192	research/comput	1342	research/comput/depart /inform/engin	0.556
homework	0.183	scienc/depart	1314	scienc/depart/page/home /engin	0.556
exam	0.177	scienc/comput/depart	1312	scienc/depart/page /engin	0.556
grade	0.171	comput/page	1228	research/comput/scienc /depart/inform/engin	0.556
class	0.149	univers/depart	1210	research/scienc/depart /inform/engin	0.556

restraint, most of the term sets are different except for one on WebKB. On Reuters-21578, in the top 10 frequent term sets by *min-sup*, there are 9 of them containing word "reuter". Apparently, they are selected due to their high occurrences, but "reuter" is a common mark appearing in almost every document without discriminating value for judging a document's category. Most of these term sets are eliminated by AD-Sup restraint. On WebKB, the problem of original frequent term sets is similar but the result is a little different. AD-Sup selects more long sets which have 3 or more terms.

4.3.2 Combination of single terms and frequent term sets

After obtaining the frequent term sets, the global feature space $\mathcal{F}$ was constructed using both single terms and frequent term sets (ST+FS). Single terms are perceived as unit sets or 1-term sets here. So the global feature space $\mathcal{F}$ is composed of term sets with element numbers ranging from 1 to n. In the categorization phase, SVM was employed as the classification model by SMO [23] algorithm using the poly kernel.

In order to find the optimal proportion of single terms and frequent term sets, all the possible combinations as enumerated in Tables 7 and 8 were evaluated where d denotes the number of single terms and D means the number of frequent term sets. Let FS_D be the frequent term sets dimensions in $\mathcal{F}$, then FS_D can be obtained by $FS\_D = d + D$. For example, when $d = 600$ and $D = 400$, then $FS\_D = 1000$. Let classifier learning from $\mathcal{F}$ be denoted as FT-SVM, therefore FT-SVM is built on 600 single term features and 400 frequent term sets, while SVM learning from the single feature space (ST) contains 1000 single terms. Tables 7 and 8 are the details of the performance comparison between the above two different feature spaces. On either dataset, the observed classification results include 64 different feature numbers from 800 to 3600.

As shown in Tables 7 and 8, on both of the datasets, FT-SVM obtains better F_1 value than SVM. On Reuters-21578 the best F_1 value is obtained as 95.1% when $d = 800$, $D =$

Table 7. F_1 value on Reuters-21578 using FT-SVM vs. SVM (%).

d \ *D*	200	400	600	800	1000	1200	1400	1600
600	94.5 93.4	94.6 93.6	94.8 93.6	94.9 93.6	94.9 93.6	94.9 93.5	94.9 93.3	94.9 93.3
800	94.3 93.6	94.8 93.6	94.8 93.6	95.0 93.6	95.1 93.5	94.8 93.3	94.9 93.3	95.0 93.3
1000	94.7 93.6	94.8 93.6	95.0 93.6	94.9 93.5	94.8 93.3	94.8 93.3	94.8 93.3	94.9 93.4
1200	94.4 93.6	94.7 93.6	94.9 93.5	94.7 93.3	94.8 93.3	94.9 93.3	94.9 93.4	94.9 93.4
1400	94.5 93.6	94.6 93.5	94.8 93.3	94.8 93.3	94.9 93.3	94.9 93.4	95.0 93.4	95.0 93.2
1600	94.5 93.5	94.6 93.3	94.8 93.3	94.8 93.3	94.9 93.4	94.9 93.4	94.9 93.2	95.0 93.2
1800	94.5 93.3	94.6 93.3	94.7 93.3	95.0 93.4	94.9 93.4	95.0 93.2	95.0 93.2	95.0 93.2
2000	94.3 93.3	94.4 93.3	94.6 93.4	95.0 93.4	94.9 93.2	94.8 93.2	95.0 93.2	95.0 93.1

Table 8. F_1 value on WebKB using FT-SVM vs. SVM (%).

d \ *D*	200	400	600	800	1000	1200	1400	1600
600	88.3 88.8	88.7 89.1	89.4 89.5	89.4 89.4	89.4 89.0	89.4 89.0	89.6 88.5	89.3 89.1
800	89.1 89.1	89.5 89.5	89.8 89.4	89.9 89.0	89.9 89.0	89.8 88.5	89.8 89.1	89.9 88.4
1000	89.6 89.5	90.1 89.4	90.5 89.0	90.7 89.0	90.8 88.5	90.6 89.1	90.4 88.4	90.3 88.2
1200	89.4 89.4	89.9 89.0	90.5 89.0	90.5 88.5	90.6 89.1	90.6 88.4	90.6 88.2	90.3 87.7
1400	89.8 89.0	90.1 89.0	90.4 88.5	90.5 89.1	90.6 88.4	90.5 88.2	90.4 87.7	90.6 88.1
1600	89.3 89.0	89.2 88.5	89.5 89.1	89.6 88.4	89.8 88.2	90.0 87.7	90.1 88.1	90.0 87.8
1800	89.1 88.5	89.1 89.1	89.2 88.4	89.5 88.2	89.6 87.7	89.8 88.1	90.1 87.8	89.7 87.6
2000	88.6 89.1	88.7 88.4	89.2 88.2	89.3 87.7	89.4 88.1	89.5 87.8	89.7 87.6	89.8 87.6

1000 and respectively, the best F_1value of original classifiers is obtained as 93.6% when *FS_D* = 1000. On WebKB the best F_1 value is obtained as 90.8% when *d* = 1000, *D* = 1000 and respectively, the best F_1 value of original classifiers is obtained as 89.5% when *FS_D* = 1200. On other dimension numbers, the FT-classifier outperforms the original version in most of cases. On Reuters-21578, the percentage of results that FT-SVM outperforms SVM is 100% (64 in 64), and on WebKB, the percentage is 90.6% (58 in 64). Table 9 shows the comparison of time costs on ST and ST+FS with the same feature number from 1000 to 2600. It spends more time on training and testing with ST+FS than ST because when features involves frequent term sets, the document vec-

Table 9. Time cost (second) when d = 800 on Reuters-21578 and d = 1000 on WebKB.

FS_D	Reuters-21578 train		Reuters-21578 test		WebKB train		WebKB test	
	ST	ST+FS	ST	ST+FS	ST	ST+FS	ST	ST+FS
1000	7.342	7.585	6.845	7.035	N/A	N/A	N/A	N/A
1200	8.172	8.547	6.926	7.965	4.062	4.719	3.421	4.062
1400	8.232	10.016	7.525	9.167	4.234	6.125	3.685	5.062
1600	8.531	11.181	7.785	10.289	4.797	7.344	4.067	6.217
1800	8.797	12.094	8.232	10.903	5.047	6.578	4.317	6.370
2000	9.120	14.062	8.267	11.520	5.250	8.563	4.626	6.578
2200	9.391	14.279	8.471	12.501	5.687	8.922	4.815	7.028
2400	9.734	14.235	8.6875	13.079	5.672	9.250	4.982	7.801
2600	N/A	N/A	N/A	N/A	5.719	10.782	5.348	8.684

tors have more non-zero elements which increases the computation time. This trend becomes more obvious as the proportion of frequent features raises but it always stays in a reasonable range. It can be concluded that in the feature space that is constructed solely by single terms, the classification performance will reach a limit as the number of feature grows, and after that, using more features will not improve the classification but contrarily bring more noise data and make the results worse. However, joining appropriate proportion of frequent term sets can extend the feature space with more discriminating features and help to obtain better results. It also proves frequent term sets refined by AD-Sup restraint are better features than surplus single words for classification.

4.3.3 Classification results analysis

To demonstrate the effectiveness of the combination strategy and term weighting method, we compared the FT-SVM with standard classifiers SVM and C4.5, which were trained on single term features in traditional way. Two frequent set based classifiers were also selected as benchmark algorithms: Classification Based on Associations (CBA) [24] and Frequent Pattern based Classification [12]. Because frequent pattern based classification selects the frequent sets in a feature-selection view and then trains SVM classifier with frequent features, it is denoted as FS-SVM in Fig. 1. On both datasets, all classifiers were trained on certain percentages of training documents and used the remainder for test. The training percentage ranges from 10% to 90% with 10% increment.

Fig. 1 shows the comparisons of classification results. For both datasets, training on solely frequent term sets features (FS-SVM) will not improve SVM classifier but contrarily decrease the F_1 value. The main reason is that many of frequent term sets have overlapping terms with others and that make the whole sets include too much redundant information. On the other hand, feature space combining single terms and frequent term sets together obtains best performance on both datasets. FT-SVM also shows superior performance to the other two frequent set based classifiers CBA and FS-SVM, and on both of the two datasets, FT-SVM obtains the highest F_1 value. The result proves that the combination strategy to construct feature space can achieve better feature space than single terms and frequent term sets.

The term weighting method by feature similarity was also examined by comparing

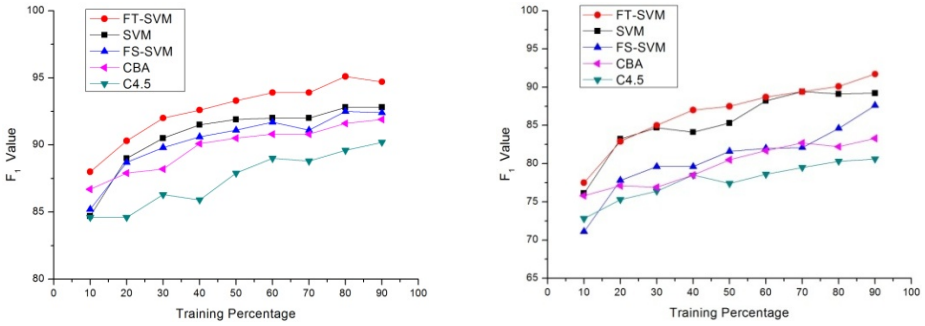

Fig. 1. Classification results on Reuters-21578 (a) and WebKB (b).

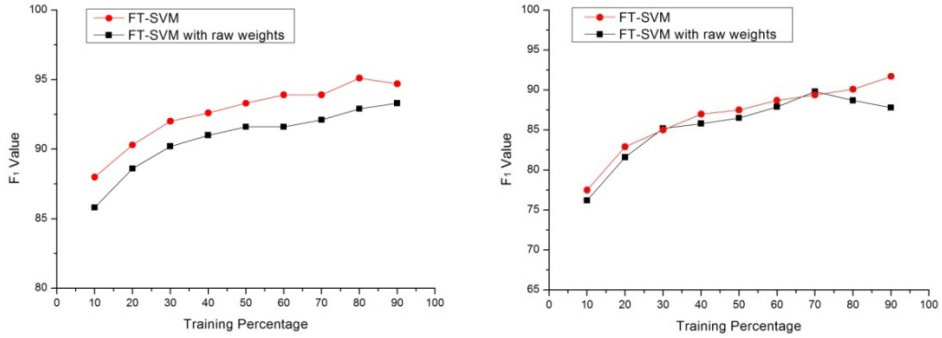

Fig. 2. Term weighting analysis on Reuters-21578 (a) and WebKB (b).

with the "raw weights" method in which the weight of a term set will be zero if it doesn't appear in a document entirety. In feature similarity based term weighting strategy, the weight of term set is assigned by feature similarity to avoid too many empty values on feature vector. If document has part of the term set, it still has weight on this feature. As is shown in Fig. 2, on Reuters-21578, the weighting method of FT-SVM improves the classification result on every training percentage, and on Web-KB, it also surpasses the original one result in most cases. The result indicates reducing sparse rate and estimating feature weight by similarity is essential to make frequent term sets applicable to construct feature space and improves the classification effect.

5. CONCLUSION

This paper has presented a novel approach for text categorization using frequent term sets. The motivation is to improve classification effect and at the mean time, provide a new way to represent text data and discover the latent term associations for better data description on the user end. To extract useful frequent term set features, a new restraint measure AD-Sup was introduced which considers more on the term sets distribution in different classes. The extraction is conducted with the Apriori strategy and

the result shows the term sets which are frequent but with general meanings are effectively removed. The feature space is constructed using both the original single words and the frequent term sets. To solve the sparse instance problem, a new term weighting method is employed to estimate the weights of frequent term sets by feature similarity. Extensive combinations of single words and frequent term sets are evaluated to achieve the optimal proportions of them. Based on the new feature space, we trained SVM classifier on the Reuters-21578 and WebKB corpus and compared with both standard classifiers and other classifiers based on frequent itemsets and association rules. The results indicate that the frequent term sets generated by AD-Sup restraint are better features than surplus single terms for classification and the combination strategy is effective to build better feature space and improve the SVM classifier.

ACKNOWLEDGMENTS

We are grateful to Shenzhen Key Laboratory of Data Vitalization (Smart City) for supporting this research.

REFERENCES

1. F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, Vol. 34, 2002, pp. 1-47.
2. R. Liu, "Interactive high-quality text classification," *Information Processing and Management*, Vol. 44, 2008, pp. 1062-1075.
3. D. Roth and K. Small, "Interactive feature space construction using semantic information," in *Proceedings of the 13th Conference on Computational Natural Language Learning*, 2009, pp. 66-74.
4. A. Kehagias, V. Petridis, V. G. Kaburlasos, and P. Fragkou, "A comparison of word- and sense-based text categorization using several classification algorithms," *Journal of Intelligent Information Systems*, Vol. 21, 2000, pp. 227-247.
5. W. Zhang, T. Yoshida, and X. Tang, "Text classification based on multi-word with support vector machine," *Knowledge-Based Systems*, Vol. 21, 2008, pp. 879-886.
6. A. Moschitti and R. Basili, "Complex linguistic features for text classification: A comprehensive study," *Lecture Notes in Computer Science*, Vol. 2997, 2004, pp. 181-196.
7. B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, 1998, pp. 80-86.
8. M. Antonie and O. R. Zaïane, "Text document categorization by term association," in *Proceedings of IEEE International Conference on Data Mining*, 2002, pp. 19-26.
9. E. Hernández-Reyes, R. A. García-Hernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "Document clustering based on maximal frequent sequences," *Lecture Notes in Computer Science*, Vol. 4139, 2006, pp. 257-267.
10. W. Zhang, T. Yoshida, X. Tang, and Q. Wang, "Text clustering using frequent itemsets," *Knowledge-Based Systems*, Vol. 23, 2010, pp. 379-388.
11. Y. Li, S. M. Chung, and J. D. Holt, "Text document clustering based on frequent word meaning sequences," *Data & Knowledge Engineering*, Vol. 64, 2008, pp. 381-

404.
12. H. Cheng, X. Yan, J. Han, and C. Hsu., "Discriminative frequent pattern analysis for effective classification," in *Proceedings of the 23rd International Conference on Data Engineering*, 2007, pp. 716-725.
13. X. Luo and A. N. Z. Heywood, "Analyzing the temporal sequences for text categorization," in *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, 2004, pp. 498-505.
14. H. Mahgoub, "Mining association rules from unstructured documents," in *Proceedings of the 3rd International Conference on Knowledge Mining*, 2006, pp. 167-172.
15. F. Thabtah, "A review of associative classification mining," *The Knowledge Engineering Review*, Vol. 22, 2007, pp. 37-65.
16. J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions," *Data Mining and Knowledge Discovery*, Vol. 15, 2007, pp. 55-86.
17. H. Ahonen-Myka, "Discovery of frequent word sequences in text," *Pattern Detection and Discovery*, LNAI, Vol. 2447, 2002, pp. 180-189.
18. F. Beil, M. Ester, and X. W. Xu, "Frequent term-based text clustering," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 436-442.
19. G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Journal of Machine Learning Research*, Vol. 3, 2003, pp. 1289-1305.
20. R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, 1994, pp. 487-499.
21. F. J. García-Peñalvo, R. Colomo-Palacios, P. Soto-Acosta, I. Martínez-Conesa, and E. Serradell-López, "SemSEDoc: Utilización de tecnologías semánticas en el aprovechamiento de los repositorios documentales de los proyectos de desarrollo de software," *Information Research*, Vol. 16, 2011, paper 504.
22. A. García-Crespo, R. Colomo-Palacios, J. M. Gómez-Berbís, and B. Ruiz-Mezcua, "SEMO: A framework for customer social networks analysis based on semantics," *Journal of Information Technology*, Vol. 25, 2010, pp. 178-188.
23. J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Technical Report MSR-TR-98-14, Microsoft Research, 1998.
24. W. Li, J. Han, and J. Pei, "CMAR: Accurate and efficient classification based on multiple class-association rules," in *Proceeding of International Conference on Data Mining*, 2001, pp. 369-376.

Man Yuan (袁满) received the B.S. degree from Beihang University, Beijing, China, in 2007. He is currently pursuing his Ph.D. degree in Computer Science and Engineering of Beihang University. His research interests include data mining, text processing and smart city.

Yuanxin Ouyang (欧阳元新) received the B.S., M.S. and Ph.D. degree from Beihang University, Beijing, China. She is now an Associate Professor in School of Computer Science and Engineering, in Beihang University. Her research interests include data mining, recommendation systems and social networks.

Zhang Xiong (熊璋) received the B.S. degree in Harbin Engineering University, Harbin, China; and received the M.S. degree in Beihang University, Beijing, China, in 1984. He is currently a Professor and Ph.D. supervisor in Computer Science and Engineering in Beihang University. His research interests include multimedia, computer control, information system and smart city.