

Meaningful Term Extraction and Discriminative Term Selection in Text Categorization via Unknown-Word Methodology

YU-SHENG LAI

AND

CHUNG-HSIEN WU

National Cheng Kung University, Tainan, Taiwan

In this article, an approach based on unknown words is proposed for meaningful term extraction and discriminative term selection in text categorization. For meaningful term extraction, a phrase-like unit (PLU)-based likelihood ratio is proposed to estimate the likelihood that a word sequence is an unknown word. On the other hand, a discriminative measure is proposed for term selection and is combined with the PLU-based likelihood ratio to determine the text category. We conducted several experiments on a news corpus, called MSDN. The MSDN corpus is collected from an online news Website maintained by the Min-Sheng Daily News, Taiwan. The corpus contains 44,675 articles with over 35 million words. The experimental results show that the system using a simple classifier achieved 95.31% accuracy. When using a state-of-the-art classifier, kNN, the average accuracy is 96.40%, outperforming all the other systems evaluated on the same collection, including the traditional term-word by kNN (88.52%); sleeping-experts (82.22%); sparse phrase by four-word sleeping-experts (86.34%); and Boolean combinations of words by RIPPER (87.54%). A proposed purification process can effectively reduce the dimensionality of the feature space from 50,576 terms in the word-based approach to 19,865 terms in the unknown word-based approach. In addition, more than 80% of automatically extracted terms are meaningful. Experiments also show that the proportion of meaningful terms extracted from training data is relative to the classification accuracy in outside testing.

Categories and Subject Descriptors: I.2.7 [Artificial Intelligence] Natural Language Processing – *speech recognition and synthesis*; H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing – *indexing methods; linguistic processing*; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *clustering; selection process*

General Terms: Algorithms, Experimentation, Measurement, Performance

Additional Key Words and Phrases: AC-machine, dimensionality reduction, discriminability, discriminative term selection, inconsistency problem, meaningful term extraction, n-gram, phrase-like unit, sparse data problem, term adaptation, term purification, text categorization, text indexing, unknown word detection, vector space modeling

1. INTRODUCTION

With the development of the Internet, the number of text documents in electronic form grows every day. Properly classifying the dramatically increasing number of text documents into predefined categories is becoming more and more important. Various

Authors' addresses: Y.S. Lai, E000 CCL/ITRI, Bldg. 51, 195-11 Sec. 4, Chung Hsing Rd., Chutung, Hsinchu, Taiwan 310; email: laisy@itri.org.tw; C.H. Wu, Dept. of Computer Science & Information Engineering, National Cheng Kung University, Ta-Hsueh Rd., Tainan, Taiwan; email: chwu@csie.ncku.edu.tw.

Permission to make digital/hard copy of part of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date of appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

© 2002 ACM 1530-0226/02/0300-0034 \$5.00

search engines were developed to allow data search to retrieve user-requested information on the Internet in a short time. In order to save time for online search, search engines must assign the documents to appropriate categories in advance, allowing the document to be referenced quickly during a search. In addition to information retrieval, text categorization can be applied to many applications, such as information filtering [Khan and Card 1997; Murthy and Keerthi 1999], email filtering [Farkas 1995; May 1997], news categorization [Jacobs 1993], and classification and routing of electronic message traffic [Schütze et al. 1995].

Text categorization has been approached in a variety of ways, such as rule-based approaches [Cohen 1995; 1996; Freund et al. 1997; Sasaki and Kita 1998]; knowledge-based methods [Jacobs 1993; Kim and Moldovan 1995]; on the basis of textual similarity [Yang and Pedersen 1997], etc. In our opinion, the best approach to text categorization is to understand the underlying meaning of each document. Unfortunately, such a solution is beyond current computational technology. Within the realm of currently attainable systems, the approach based on textual similarity is a good solution. Textual features, so-called terms [Schäuble 1997], are extracted from documents and used for estimating the textual similarity between documents. Therefore, the extraction of textual features is very important, and often determines system performance.

Three textual features: characters, words, and n -grams are typically employed for textual indexing. Most characters and words are evenly distributed over a wide range of categories. Such terms lack category dependency (certain strings may be words in some domains while not in others [Nie 1995]), and hence are basically useless for representing or determining categories. For n -gram-based indexing, storage requirements are comparatively higher than those for other indexing methods. Moreover, an n -gram is not a meaningful unit in linguistics. N -gram-based indexing tends to cause inconsistencies between training data and testing data when a term appears in the training data but not in the testing data. Solving the inconsistency problem is very important because terms that have high discriminative ability, but are not seen in testing data, have no practical use. Since using more meaningful terms for indexing can help solve this problem, we propose to extract meaningful terms. On the other hand, a term may appear in testing data but not in training data, i.e., the so-called sparse data problem. We believe that this problem can be partially improved by further term adaptation, but leave it to future research. Instead of the textual features mentioned above, we employ unknown words for textual indexing due to their domain dependency.

There are two main methods for detecting unknown words: statistical and rule-based approaches. Statistical approaches gather statistics such as the frequency of term occurrence/co-occurrence. They can be easily adapted to other domains, although they require a large amount of training data, which should sometimes be prepared manually. Chang et al. [1994] presented a statistical approach for recognizing Chinese proper names. The necessary statistical information was drawn from a general corpus and a proper name corpus. A similar approach was proposed in Sun et al. [1994]; both methods were restricted to a particular type, i.e., proper names.

Rule-based approaches detect unknown words by using a dictionary and heuristic rules for forming words, but they require a complete dictionary. It is not possible for a dictionary to contain all the words in Chinese nor to specify all the rules for word formation. A corpus-based learning method was proposed for detecting unknown Chinese

words [Chen and Bai 1998]. It has the average amount of automatic rule learning. One of the steps for detecting unknown words requires extra information, i.e., the part-of-speech. In practice, such information is not always available, nor is it consistently reliable. Liang and Yeh [2000] presented a two-phase mechanism for detecting unknown words. In the first phase, several principles of word formation are used to extract only the possible three-character unknown words and a set of stop words, which are constructed by a meaningless character during word formation and are used to filter the extracted unknown words. In the second phase, a neural network classifier based on the back propagation algorithm is constructed for further recognition of unknown words. Some unknown words may be useful for text categorization, but may be illegal or may be included in the stop word list.

From the results of an experiment on the distribution of category frequencies (to be introduced), it is indeed proven that the unknown words presented are more domain-specific than the traditional words. In order to focus on the important characteristic of unknown words, i.e., *domain dependency*, this article proposes using unknown words as indexing terms for text categorization. Our strategy is to extract unknown words from text documents and select those that are highly domain-specific as textual features. The extraction of unknown words is based on our previous research, which used a statistical approach [Lai and Wu 2000]. The statistical information was gathered from a word-based heptagram, i.e., the seven-order Markov model, since low-order Markov models, such as bigrams, can barely handle words containing more than two characters [Li et al. 1991]. In other words, this language model can help in detecting very long unknown words (in our experiment, 30-character words at most). Due to the domain dependency of the proposed indexing terms, a probabilistic measure, called discriminability, was also proposed, which works well for selecting highly domain-specific terms.

A brief system overview is presented in Section 2. The term extraction and selection method is described in Section 3. The AC-machine for text indexing and the classification function are described in Section 4. Finally, in Section 5, several experiments that evaluate the performance of the proposed methods are presented. Key themes of this article are discussed in Section 6. Generalized conclusions are presented in Section 7.

2. SYSTEM OVERVIEW

A text categorization system can be divided into two phases: a training phase and a classification phase. In the training phase, a set of collected and preclassified text documents are used to construct a classification model. In the classification phase, the system accepts a text document and assigns it to a predefined category by using the trained classification model. A block diagram of the training phase is shown in Figure 1, where a phrase-like unit (PLU) is an unknown word extracted from an n -gram language model and employed for text indexing. In the n -gram construction process, K topics of preclassified text documents are collected to construct K word-based n -gram language models. The number n is assigned a range from 1 to 8, which provides enough occurrence and co-occurrence data for PLU extraction. In the PLU extraction process, PLUs are extracted from each n -gram database individually. However, some of the preliminary terms are of no use in classifying text documents, and are discarded in the further purification process. A total of K sets of terms notated as $T_k, k = 1, 2, \dots, K$ are extracted.

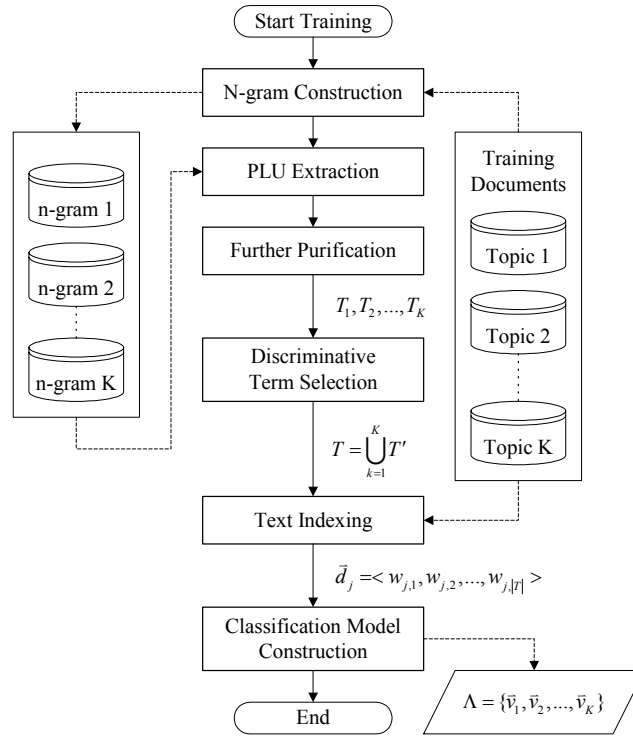


Fig. 1. Training phase in the text categorization system.

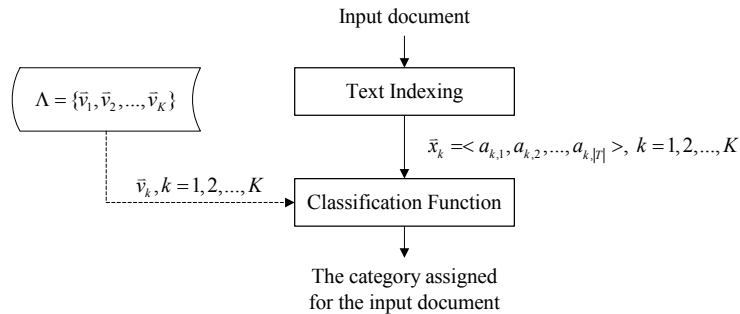


Fig. 2. Classification phase in the text categorization system.

After extracting the PLUs, terms that possess utility are selected when distinguishing text documents by topic. This is the discriminative term selection process. A measure called “discriminability” is defined for measuring the degree of utility in distinguishing categories. Terms with low discriminability are pruned from each term set T_k . The

pruned term sets $T'_k, k = 1, 2, \dots, K$ are combined into a term set normalization method. In the text-indexing process, each text document is represented as a description vector \vec{d}_j composed of a set of term weights $\{w_{j,1}, w_{j,2}, \dots, w_{j,|T|}\}$. In our approach, since the proposed terms are compound words and are undefined in the vocabulary, an AC-machine is employed and modified for text indexing. The AC-machine proposed by Aho et al. is an efficient string pattern-matching algorithm for locating all occurrences of a finite number of keywords in a text string [Aho and Corasick 1975; Aoe 1989; Hayashi and Mochizuki 1999]. All the training text documents are represented as description vectors and constructed into a classification model Λ , consisting of K mean vectors in the construction process. Figure 2 illustrates a block diagram of the classification phase. In the process of text indexing, because the proposed term weighting method is category dependent, each unclassified document is represented in K description vector $\vec{x}_k, k = 1, 2, \dots, K$. The classification function generates a classification score by comparing the description vectors of the unclassified documents and the mean vectors \vec{v}_k in the classification model Λ , and then assigns the unclassified document to the category with the highest classification score.

3. TERM EXTRACTION AND SELECTION

The basic idea of our term extraction process can be found in our previous research [Lai and Wu 2000] that extracts unknown words and phrases, i.e., PLUs, from sentences in a specific domain by using a statistical method. According to our previous study, most of extracted PLUs are extremely domain specific. The current study attempts to improve text categorization by utilizing this significant characteristic. Thus, in this article, PLUs are chosen as the terms for text indexing.

3.1 PLU Extraction

The proposed method for PLU extraction is based on an assumption described in the following. For a frequently occurring word sequence $p = w_1 w_2 \dots w_n$, if there is a word w_i in the word sequence p such that the preceding word sequence $w_1 w_2 \dots w_i$ is always followed by the word sequence $w_{i+1} w_{i+2} \dots w_n$, the word sequence p is probably an unknown word or phrase. That is, if two word sequences always occur together, their concatenation is probably an unknown word or phrase. For example, the unknown word “chen2 shwei2 byan3 (陳水扁),” the name of the president of Taiwan, is undefined in the dictionary. It consists of three one-character words “chen2 (陳),” “shwei2 (水),” and “byan3 (扁).” Every time the last two words “shwei2 (水)” and “byan3 (扁)” occur together, the first word “chen2 (陳)” always occurs in front of them. Under this assumption, a PLU-based likelihood ratio $PLR(p)$ is defined in Eq. (1) to decide if a word sequence is an unknown word or not;

$$PLR(p) = \max_i \frac{tf(p)}{\min\{tf(w_1 w_2 \dots w_i), tf(w_{i+1} w_{i+2} \dots w_n)\}} \quad (1)$$

where $tf(\cdot)$ denotes term frequency. Note that $tf(p)$ must be smaller than or equal to both $tf(w_1w_2\dots w_i)$ and $tf(w_{i+1}w_{i+2}\dots w_n)$ because both $w_1w_2\dots w_i$ and $w_{i+1}w_{i+2}\dots w_n$ are substrings of p . When $tf(w_1w_2\dots w_i) \leq tf(w_{i+1}w_{i+2}\dots w_n)$ and $tf(p)$ approximates to $tf(w_1w_2\dots w_i)$, then an occurrence of the word sequence $w_1w_2\dots w_i$ is frequently followed by the word sequence $w_{i+1}w_{i+2}\dots w_n$. However, when $tf(w_{i+1}w_{i+2}\dots w_n) \leq tf(w_1w_2\dots w_i)$ and $tf(p)$ approximate $tf(w_{i+1}w_{i+2}\dots w_n)$, then an occurrence of the word sequence $w_{i+1}w_{i+2}\dots w_n$ is frequently preceded by the word sequence $w_1w_2\dots w_i$. Therefore, the “min” function is applied to pick the smaller one between $tf(w_1w_2\dots w_i)$ and $tf(w_{i+1}w_{i+2}\dots w_n)$. On the other hand, the “max” function is used for seeking a position i such that $tf(p)$ approximates $\min\{tf(w_1w_2\dots w_i), tf(w_{i+1}w_{i+2}\dots w_n)\}$.

Without loss of the generality, the PLU-based likelihood ratio can be simplified as

$$PLR(p) = \frac{tf(p)}{\min\{tf(w_1w_2\dots w_{n-1}), tf(w_2w_3\dots w_n)\}} \quad (2)$$

Thus the computational complexity can be effectively reduced from $O(n)$ to $O(1)$ in the worst case. The detailed proof is shown in Lai and Wu [2000]. By using the PLU-based likelihood ratio, a word sequence $p = w_1w_2\dots w_n$ is considered an unknown word if the following conditions hold simultaneously:

- (1) $n > 1$,
- (2) $tf(p) \geq c$, and
- (3) $PLR(p) \geq 1 - \varepsilon$ or $PLR(p) \cdot tf(p) \geq d$

where c is a frequency constraint defined as a lower bound of the frequency of word sequence p ; ε is a fault tolerance of the likelihood ratio $PLR(p)$; and d is a product constraint defined as a lower bound of the product of the likelihood ratio $PLR(p)$ and the frequency $tf(p)$.

The first two conditions are trivial. The third condition consists of two subconditions separated by the operation “or.” The left-hand subcondition means that there is at least one word w_i in the word sequence p such that the leading word sequence $w_1w_2\dots w_i$ always accompanies the word sequence $w_{i+1}w_{i+2}\dots w_n$ when the PLU-based likelihood ratio $PLR(p)$ approximates 1. Then the word sequence p is a PLU. After error analysis, we found that some unknown words consist of subword sequences with very high occurrence, but the unknown words occur less frequently than the subword-sequences. The phenomenon often occurs when $n = 2$. In this case, applying only the left-hand subcondition will lose some meaningful PLUs. Hence the right-hand subcondition is added to solve the problem, and implies that the terms with very high occurrence frequency can be regarded as PLUs, even though their PLU-based likelihood ratios are low.

The definition of the PLU-based likelihood ratio is based on the following assumption: “Unknown words and phrases are always the combination of some words defined in the lexicon and some undefined characters.” Moreover, in text categorization,

important terms such as proper nouns are usually unknown words or phrases. Important information embedded in unknown words includes highly domain-specific characteristics.

However, there is a problem in the direct use of the extraction method for text categorization. In the PLU extraction process, three parameters: the frequency constraint (c), the fault tolerance (ε), and the product constraint (d), should be determined, and are variant from corpus to corpus. As yet it is not easy to determine these parameters automatically. As we observed in our previous experimental results: “The recall rates will be increased as reducing the frequency constraints, increasing the fault tolerances, or reducing the product constraints.” [Lai and Wu 2000]. It means that terms can be extracted in an appropriate amount by tuning the parameters to achieve a high recall rate. We then need to merely select the terms that can distinguish one category from the others.

Even though large quantities of terms can be extracted, we should make sure that the extracted terms are meaningful. Compared to nonmeaningful terms, meaningful terms are more representative for untrained text documents. Therefore, fault tolerance cannot be increased enough to prevent extracting meaningless PLUs.

3.2 Further Purification

Of the extracted PLUs, some are useless or even interfering. To reduce the effect of the interfering terms on system performance, the purification process shown in Algorithm 1 is performed. The interfering terms can be divided into two types: stopping terms and cross-included terms. The stopping terms include dates, times, numbers, quantities, addresses, etc. The discarding of stopping terms is based on the following heuristic rules:

(1) discard the terms with a set of predefined prefixes such as “而 (*er2*),” “並 (*bing4*),” “且 (*chye3*);” (2) discard the terms with a set of predefined postfixes such as “了 (*le0*),” “的 (*de0*);” (3) discard the terms containing a set of predefined stopping words such as “公斤 (kilogram),” “公尺 (meter),” “世紀 (century).”

The cross-included terms, indicating that a group of terms include each other, are generated from the PLU extraction process. In our previous research, since PLUs were extracted directly from sentences instead of n -gram language models, some may overlap. A Phrase Choosing Algorithm [Lai and Wu 2000] was proposed and used for solving the problem. However, since a large number of sentences had to be treated, the approach took much time to extract PLUs directly from sentences. In order to save extraction time, in this article PLUs are extracted indirectly from word n -gram language models instead of sentences.

However, a problem arises from the simplified PLU extraction procedure. Suppose there are two terms in the extracted PLUs, one of which is a substring of the other. It is reasonable that some of their occurrences will overlap. It is improper to extract two different terms from the same location in a document. To solve this problem, the reliability degree defined in Eq. (3) decides which one is more suitable.

$$Rel(t_i, t_j) = Comp(PLR(t_i), PLR(t_j)) + Comp(tf(t_i), tf(t_j)) \quad (3)$$

where $Rel(t_i, t_j)$ denotes the reliability degree for term t_i with respect to term t_j , which is determined by comparing the two term frequencies and the two PLU-based likelihood ratios. For approaches based on textual similarity, the term frequency is often used to measure the term weight. In this article, the PLU-based likelihood ratio is used to

measure the meaningful degree of terms. Equation (3) combines both the term frequency and the PLU-based likelihood ratios. The $Comp(a, b)$ function in Eq. (3) compares the two parameters a and b , and then returns trivalued values as follows:

$$Comp(a, b) = \begin{cases} 1, & \text{if } a > b \\ 0, & \text{if } a = b \\ -1, & \text{if } a < b \end{cases} \quad (4)$$

As mentioned above, the cross-included problem occurs in a group of terms rather than in two terms only. A term may be reliable for some terms, i.e., reliability is greater than or equal to 0, and unreliable for other terms. For this reason, the decision to reserve a term or not can be made according to a cumulative reliability degree as defined in Eq. (5), which is computed by accumulating all the reliabilities from the terms cross-included with it. This idea is based on decision by majority voting.

$$CR(t_i) = \sum_{\forall t_j \neq t_i, t_j A t_i} Rel(t_i, t_j) \quad (5)$$

where $CR(t_i)$ denotes the cumulative reliability degree of term t_i and A is a binary relation that denotes that term t_j includes or is included by term t_i .

A special case should be taken into account. When there are n terms, t_1, t_2, \dots, t_n , with the same term frequency, $PLR(t_i) \leq PLR(t_{i+1})$ for $i = 1, 2, \dots, n-1$, and t_i is a substring of t_{i+1} for $i = 1, 2, \dots, n-1$, it means that they occur at the same positions in all documents that contain them, and that the longer ones are more meaningful than the shorter ones. According to the longest match principle [Liu and Liang 1989], the longest term, i.e., t_n , is the most suitable to reserve. However, if applying only the cumulative reliability degree, the longer $\lfloor n/2 \rfloor$ terms, i.e., $t_{\lfloor \frac{n}{2} \rfloor + 1}, t_{\lfloor \frac{n}{2} \rfloor + 2}, \dots, t_n$, will be reserved because they have positive cumulative reliability degrees. Hence, a deletion procedure for this kind of cross-included terms should be performed before the deletion using the cumulative reliability degree. The purification process is summarized as follows:

Algorithm 1. The Purification Process

- Step 1. Discard the stopping terms such as dates, times, numbers, quantities, addresses, etc.
- Step 2. Find all groups in which each group consists of terms $\{t_1, t_2, \dots, t_n\}$ and satisfies the following conditions:
 - (1) $t_i \subseteq t_{i+1}$, (\subseteq denotes “a substring of”),
 - (2) $PLR(t_i) \leq PLR(t_{i+1})$, and
 - (3) $tf(t_i) = tf(t_j) \quad \forall 1 \leq i, j \leq n$.
- Step 3. Cluster the remaining terms into several groups such that each group is isomorphic to a transitive closure of a binary relation A on S , where S is a finite set with a group of terms $\{t_1, t_2, \dots, t_n\}$ and a binary relation A on S is a

subset of $S \times S$. If $t_i \subseteq t_j$ or $t_j \subseteq t_i$, we say that t_i is A -related to t_j and use the notation $t_i A t_j$. The transitive closures can be found using the depth-first search algorithm [Baase 1989]. The cumulative reliability degree $CR(t_i)$ defined in Eq. (5) is computed for each term t_i in transitive closures. Discard the terms with a negative cumulative reliability degree for each group.

3.3 Discriminative Term Selection

An important consideration for term selection is to pick the most representative terms that can be used to distinguish text documents with topics. Each candidate term is given a weight during the term selection process, and these weights are used for deciding whether the term is ultimately selected or not. For approaches based on textual similarity, the selected terms usually dominate system performance. A discriminative term selection method is needed. Here the term “discriminative” implicitly indicates the utility in distinguishing categories.

For text categorization, the simplest task is classifying text documents into two categories. In the case of two categories, it is reasonable that a term has two weights representing, respectively, the two categories. The probability of the term in each category can be used as the measure of term weighting. For the two occurrence probabilities, if one is much larger than the other, it means the term has high discriminative ability in distinguishing the two categories. If, on the contrary, the two occurrence probabilities are similar, it means the term has low discriminative ability in distinguishing the two categories. Thus, the discriminative ability of a term can be defined as the ratio of the two occurrence probabilities.

By expanding the number of categories, a term weight, called discriminability, is defined on its ability to distinguish among documents with topics. For a term t representing category g , discriminability $W(t, g)$ can be defined as in Eq. (6):

$$W(t, g) = \frac{p(g|t)}{\max_{c \neq g} p(c|t)}. \quad (6)$$

Since discriminability is the ratio of two conditional probabilities $p(g|t)$ and $\max_{c \neq g} p(c|t)$, and $p(g|t) = \frac{p(t \in g)}{p(t)}$ and $\max_{c \neq g} p(c|t) = \frac{\max_{c \neq g} p(t \in c)}{p(t)}$, Eq. (6) can be simplified as follows:

$$W(t, g) = \frac{p(t \in g)}{\max_{c \neq g} p(t \in c)} \quad (7)$$

where $p(t \in c)$ denotes the occurrence probability of term t in category c , which can be computed as Eq. (8):

$$p(t \in c) = \frac{tf(t \in c) + 1}{\sum_w tf(w \in c) + 1}. \quad (8)$$

As shown in Eq. (7), the discriminability of term t for category g is defined as the ratio of the occurrence probability $p(t \in g)$ to the maximum of the occurrence probabili-

ties $p(t \in c)$ for all categories $c \neq g$. It implies that the discriminative ability of a term is defined as the ratio of the occurrence probability in the represented category to the maximum of the others. In other words, if a term's occurrence probability in one category is much higher than the other occurrence probabilities, we say that the term has high discriminability in distinguishing the category from the others.

For each category, therefore, we should select only the terms with higher discriminability. Further, discriminability should be greater than a threshold that must be greater than or equal to 1. A term with discriminability less than 1 means that the category is not the category represented with the highest occurrence probability. In other words, there is at least one other category such that the term's occurrence probability in the category is higher than that in the represented category. Such terms are too weak to stand for the represented category and should be discarded. Finally, the selected terms from all categories are joined together for text indexing.

4. INDEXING AND CLASSIFICATION

In character-based and word-based approaches, the terms are defined in a vocabulary. In a bigram-based approach, all terms have the same number of characters or words; for example, there are two characters or words in a bigram. In our approach, however, the terms are unknown words that are undefined in the vocabulary and are of different lengths. Our problem is how to index a large number of text documents by using such terms. The AC-machine is employed to solve this problem.

4.1 The Indexing Machine

The AC-machine is an efficient string pattern-matching algorithm for locating all occurrences of a finite number of keywords in a text string. In this article, a string is simply a finite sequence of Chinese characters and symbols such as a text document. A selected term t is also a string and T is a finite set of the selected terms. For each term in T , our problem is to locate and count the term frequency in an arbitrary text document.

We define an indexing machine as $M = (S, I, g, f, s_0, O)$, where S is a finite set of states; I is a finite set of input symbols; g , called a goto function, is a transition function from $S \times I$ to $S \cup \{fail\}$ in which *fail* indicates a failure transition; f is a transition function from S to S , called a failure function, which works when the corresponding goto function does not accept an input symbol, i.e., went to *fail*; s_0 is the initial state in S , numbered 1 in this article; and O is a transition function from S to a subset of T , called an output function.

An example of the indexing machine is shown in Figure 3. When the string “ban4 dz4 ju4 you2 cheng2 (半自助套裝遊程)” is fed into the indexing machine, the indexing machine outputs the three terms “ban4 dz4 ju4 (半自助),” “ban4 dz4 ju4 tau4 jwang1 you2 cheng2 (半自助套裝遊程),” and “tau4 jwang1 you2 cheng2 (套裝遊程)” for the following reason: the machine goes through two states with nonempty outputs, i.e., state 4 outputs the term “ban4 dz4 ju4 (半自助),” and state 8 outputs the two terms “ban4 dz4 ju4 tau4 jwang1 you2 cheng2 (半自助套裝遊程)” and “tau4 jwang1 you2 cheng2 (套裝

遊程)” in which the term “tau4 jwang1 you2 cheng2 (套裝遊程)” is the output of state 16.

4.2 Indexing Method and Classification Function

A fairly large number of classification models have been proposed for text categorization, such as the Naïve Bayes Classifier [Joachims 1997; Li and Jain 1998; Tsay and Wang 1999], Subspace Model [Li and Jain 1998; Oja 1983], Neural Network [Farkas 1995; Farkas 1996; Lam and Lee 1999], Fuzzy Logic [Benkhalifa and Bensaid 1999; Jo 1999], and other combination models [Larkey and Croft 1996; Li and Jain 1998]. However, these models require time-consuming computation; hence, to improve computational performance, the commonly used vector space model (VSM) [Schäuble 1997] is employed.

The VSM for text categorization consists of a large number of categorization methods, each of which contains an indexing method and a classification function. The following describes the indexing method and classification function we employ in this article.

4.2.1 Indexing Method

The essential idea of VSM is to represent a document as a description vector in which each component of the vector is a weighted indexing feature. In our approach, the indexing features are unknown words located and counted by the indexing machine. After locating and counting the unknown words hidden in a document, we should give each unknown word a weight. The weighting process can be divided into two parts: one for training documents and one for unclassified documents.

Term weighting for training documents. Since the training documents are collected for model training, each should be preassigned to the category architecture, i.e., a predefined class. Suppose the training documents are classified into K categories and there are N_k documents in the k -th category ($1 \leq k \leq K$). Each training document $D_{k,j}$, the j -th document ($1 \leq j \leq N_k$) in the k -th category, can be represented by a description vector $\vec{d}_{k,j}$ as follows:

$$\vec{d}_{k,j} = \langle c_{k,j,1}, c_{k,j,2}, \dots, c_{k,j,i}, \dots, c_{k,j,n} \rangle \quad (9)$$

where n is the total number of the selected terms and $c_{k,j,i}$ denotes the term weight of a term t_i in the document $D_{k,j}$. Since the PLU-based likelihood ratio is used for meaningful term extraction and discriminability is estimated for document classification, we combine these two estimates with the term frequency of term t_i in document $D_{k,j}$ to form the term weight $c_{k,j,i}$, as follows:

$$c_{k,j,i} = tf(t_i, D_{k,j}) \cdot PLR(t_i) \cdot S(W(t_i, g(D_{k,j}))) \quad (10)$$

where $tf(t_i, D_{k,j})$ is the frequency at which the term t_i occurs in the document $D_{k,j}$; $PLR(t_i)$ is the PLU-based likelihood ratio of the term t_i ; $g(D_{k,j})$ denotes the category

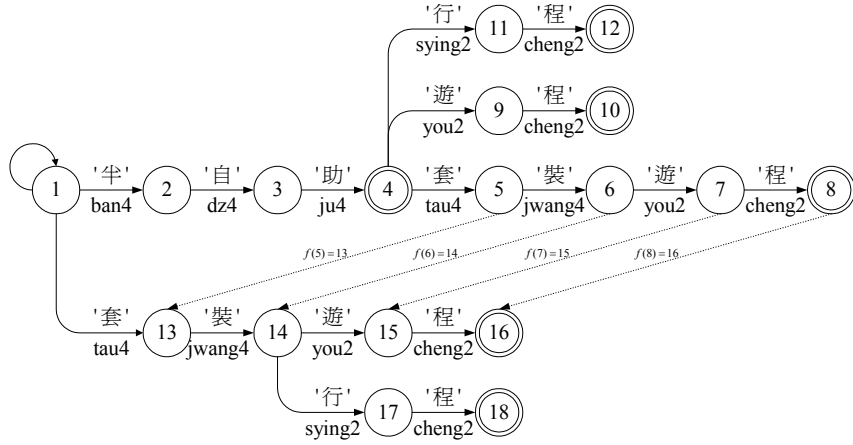


Fig. 3. An example of the indexing machine.

containing the document $D_{k,j}$; $W(\cdot)$ is the discriminability defined in Eq. (6); and $S(w)$ is a smooth 0-1 function, defined as follows:

$$S(w) = \frac{1}{1 + \frac{1}{w - \alpha}} \quad (11)$$

where α is a constant, and is regarded as the discriminability threshold. This smooth function can convert discriminability into an interval between 0 and 1. It also avoids a bias problem, as follows. By experiment, we found that some of the terms have discriminability that is too high, making these terms too strong compared to others. This is not fair, and so we created a smooth function to solve the problem. In other words, the smooth function can reduce the proportions of a higher discriminability to a lower one.

Term weighting for unclassified documents. To implement the definition in Eq. (10), the term weight a_i can be computed by multiplying three terms: the term frequency, the PLU-based likelihood ratio, and discriminability. The third term is especially category dependent. Since we do not know the category of an unclassified document, each unclassified document should be represented as multiple description vectors in accordance with the possible categories. That is, to classify a new document X into K categories ($G_k, k = 1, 2, \dots, K$), the unclassified document should be represented as K description vectors ($\vec{x}_k, k = 1, 2, \dots, K$), as follows:

$$\vec{x}_k = \langle a_{k,1}, a_{k,2}, \dots, a_{k,i}, \dots, a_{k,n} \rangle \quad (12)$$

where the term weight $a_{k,i}$ is modified from $c_{k,j,i}$ defined in Eq. (10), and is defined as:

$$a_{k,i} = tf(t_i, X) \cdot PLR(t_i) \cdot S(W(t_i, G_k)) \quad (13)$$

4.2.2 Classification Function

By the indexing method, each training document can be represented as a description vector and each unclassified document can also be represented as a set of description vectors that correspond to the possible categories. A classification function is used for measuring how an unclassified document belongs to a predefined category. In our approach, we employ the cosine function as the classification function. Therefore, we combine the description vectors of each category into a mean vector. The details are described below.

According to the indexing method, each training document $D_{k,j}$ in a category G_k of size N_k can be represented as a description vector $\vec{d}_{k,j}$. By combining the description vectors belonging to the same category, a mean vector $\vec{v}_k = \langle b_{k,1}, b_{k,2}, \dots, b_{k,i}, \dots, b_{k,n} \rangle$ can be formed in which the component $b_{k,i}$ is defined as follows:

$$b_{k,i} = \frac{1}{N_k} \sum_{j=1}^{N_k} \frac{c_{k,j,i}}{\|\vec{d}_{k,j}\|} \quad (14)$$

where $c_{k,j,i}$ is the i -th component of the vector $\vec{d}_{k,j}$, i.e. the term weight of the term t_i in the document $D_{k,j}$, and $\|\vec{d}_{k,j}\|$ denotes the norm of the description vector $\vec{d}_{k,j}$.

To measure how an unclassified document X belongs to a category G_k , a classification function $f_{G_k}(X; \Lambda)$ parameterized on Λ for each category G_k is defined as:

$$f_{G_k}(X; \Lambda) = \rho_{\cos}(\vec{x}_k, \vec{v}_k) = \frac{\sum_{i=1}^n a_{k,i} b_{k,i}}{\sqrt{\sum_{i=1}^n a_{k,i}^2} \sqrt{\sum_{i=1}^n b_{k,i}^2}} \quad (15)$$

where \vec{x}_k is the description vector of the document X for the category G_k and \vec{v}_k is the mean vector of the category G_k .

When there are many indexing terms where weights $a_{k,i}$ and $b_{k,i}$ are large and the document norm $\sqrt{\sum_{i=1}^n a_{k,i}^2}$ and the category norm $\sqrt{\sum_{i=1}^n b_{k,i}^2}$ are small, the cosine classification function returns a large value. But when most of the indexing terms in the unclassified document are different from the indexing terms in the training document, the cosine classification function returns a small value.

The task of text classification is based on use of the classification function in implementing the decision rule, which is often generically stated as follows:

$$C(X) = G_h \text{ if } h = \arg \max_k f_{G_k}(X; \Lambda) \quad (16)$$

where $C(X)$ denotes the classification decision on a document X and G_h denotes the h -th category.

5. EXPERIMENTAL RESULT

5.1 CORPUS

In our experiments, due to the lack of the benchmark corpus for Chinese text categorization, we collected a corpus for training and testing from the Website of the Min-Sheng Daily News (MSDN). The collection duration was from December 1997 to July 1999, during which the collection from December 1997 to April 1999 was for training, and that from May 1999 to July 1999 was for testing. There were a total of 44,675 text documents, consisting of over 35 million words in the corpus. Each text document was assigned a category by the MSDN staff. All text documents were classified into a hierarchical structure of two layers, illustrated in Figure 4.

5.2 Performance Evaluation

For binary classification, classifiers are typically evaluated by using a contingency table, as shown in Table I [Manning and Schütze 1999], in which a , b , c , and d are the numbers of a YES/NO tag being correctly/incorrectly assigned. For example, a is the number of objects in the category of interest that were correctly assigned to the category. The classification accuracy is defined as

$$\frac{a+d}{a+b+c+d}$$

the ratio of correctly classified objects.

In our experiments, we also evaluated performance by making a 2×2 contingency table for each category c_i separately (evaluating c_i vs. $\neg c_i$). There are then two ways to proceed: the macro-average and micro-average. The first computes an evaluation measure such as accuracy for each contingency table separately and then averages the evaluation measure over categories to get an overall measure. The second way first makes a single contingency table for all the data by summing the scores in each cell for all categories and then computes the evaluation measure for this large table. In what follows in this section, since the control category $\neg c_i$ consists of text documents from the category c_i , the macro-average accuracy is equal to the micro-average accuracy, and is viewed as the final evaluation measure.

5.2.1 Baseline Performance

A VSM baseline system, which uses the words defined in a dictionary to index text documents, was implemented for comparison with the proposed approach. The dictionary was originally provided by the Academia Sinica, Taiwan, and consists of 84,559 words with 13,071 one-character words, 48,391 two-character words, 11,567 three-character words, 10,444 four-character words, and the rest five- to ten-character words.

Each text document was first segmented into a group of words by a Chinese word segmentation process based on the dictionary. The purification process was performed on the extracted words in the same way as proposed in this article. A total of 50,576 words were used in the indexing process. That is, there was a total of 50,576 terms used for

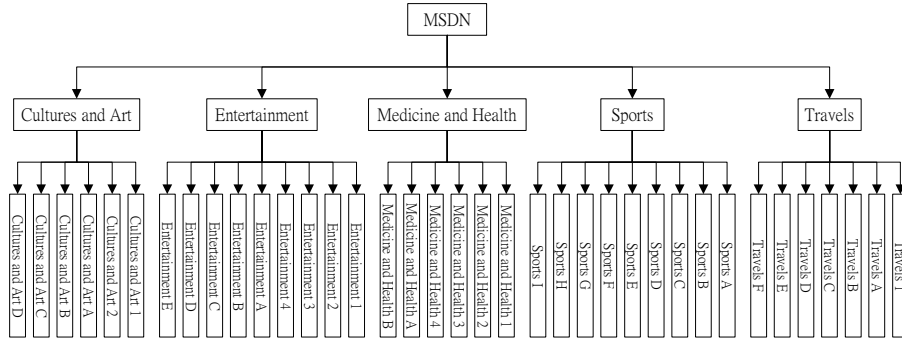


Fig. 4. The hierarchically classified MSDN.

Table I. Contingency Table for Evaluating a Binary Classifier

	<i>YES correct</i>	<i>NO correct</i>
YES was assigned	<i>a</i>	<i>b</i>
NO was assigned	<i>c</i>	<i>d</i>

Table II. Experimental Results for Word-based Approach with Term Weight TF×IDF

		<i>Average Accuracy</i>		<i>Grand Average</i>	
		Inside Test	Outside Test	Inside Test	Outside Test
1 st layer		83.74%	80.12%	-	
2 nd layer	Culture & art	77.69%	72.58%	78.63%	74.63%
	Entertainment	75.56%	73.14%		
	Medicine & health	78.02%	75.63%		
	Sports	81.56%	76.51%		
	Travel	80.33%	75.29%		

classification. Then the TF×IDF measure was used for term weighting. For the training set, the mean vector, which is formed by the components defined in Eq. (14), was computed for each category. For the classification function, the cosine function was employed to measure the similarity between an unclassified document and each category mean vector.

The experimental results listed in Table II show that the average accuracy of the classification in the first layer achieved 83.74% for inside testing and 80.12% for outside testing. We also conducted an experiment on a combination of word-based indexing and the proposed discriminability measure. The experiment was performed on the first layer

Table III. Experimental Results for Word-based Approach with Term Weight Discriminability

	<i>Inside Test</i>	<i>Outside Test</i>
Culture & art	99.64%	99.16%
Entertainment	68.77%	66.75%
Medicine & health	82.33%	82.47%
Sports	24.28%	28.56%
Travel	78.47%	79.53%
Average accuracy	70.70%	71.29%

classification. Table III shows the experimental results. Average accuracy achieves performance of 70.70% for inside testing and 71.29% for outside testing. This demonstrates that the proposed measure, discriminability, is not suitable for word-based indexing.

5.2.2 Parameter Testing

For each category, the number of representative terms is variable. Hence it is not intrinsically apparent whether we should normalize the term weights in each category or constrain the number of terms selected from each category. Further, as mentioned above, discriminability should be greater than a threshold and the threshold should be greater than 1. So we conducted this experiment by changing the following parameter settings:

1. For a set of selected terms t_1, \dots, t_n , suppose the terms occur in a document identically and suppose that the PLU-based likelihood ratios of these terms are all the same. Then the probability that the document belongs to each category should be equal. To maintain this equivalence, we normalize the discriminability of a term t_i to a category G_k as follows:

$$\hat{S}(W(t_i, G_k)) = \frac{S(W(t_i, G_k))}{\sum_{j=1}^n S(W(t_j, G_k))} \quad (17)$$

In other words, the term $S(\cdot)$ in Eqs. (10) and (13) is replaced by $\hat{S}(\cdot)$ after normalizing discriminability.

2. An alternative method for maintaining this equivalence is to equalize the number of the representation terms in each category. We simply let the number of the terms selected from each category be equal to the smallest one.
3. To examine the discriminability effect on performance, we experimented with changing the discriminability threshold from 0 to 3 in steps of 1.

Table IV lists the results of the experiments on accuracy. The notation *Norm* and *UNorm* denote the normalizing and unnormalizing of discriminabilities; *CN* and *UCN* denote constraining and unconstraining the number of selected terms.

Figures 5 and 6 compare the accuracy for discriminability thresholds from 0 to 3 for different parameter settings. The 1.0 accuracy for the discriminability threshold outperforms the others, except for the parameter setting *UNorm+UCN*. In the case of the parameter setting *UNorm+UCN*, although the 2.0 accuracy for the discriminability

Table IV. Comparing Accuracy for Different Parameter Settings: $Norm / UNorm$ denote the normalizing and unnormalizing discriminabilities; CN / UCN denote constraining and unconstraining the number of selected terms and discriminability thresholds (0–3)

		<i>Discriminability Thresholds</i>			
		0	1	2	3
Inside	UNorm + UCN	95.97%	96.30%	96.32%	95.78%
	Norm + UCN	95.90%	96.34%	96.33%	95.79%
	UNorm + CN	96.18%	96.26%	95.80%	94.72%
	Norm + CN	96.21%	96.29%	95.83%	94.72%
Outside	UNorm + UCN	94.87%	95.30%	95.32%	94.78%
	Norm + UCN	94.85%	95.31%	95.30%	94.78%
	UNorm + CN	95.12%	95.30%	94.66%	93.52%
	Norm + CN	95.08%	95.27%	94.64%	93.52%

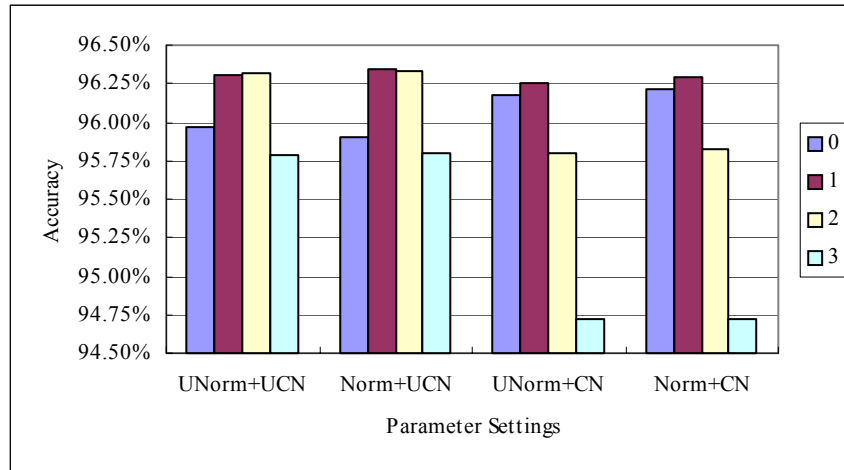


Fig. 5. Unknown word-based inside test on the first layer for comparing the accuracies of discriminability thresholds 0-3 for different parameter settings.

threshold, outperforms the others, it still confirms the viewpoint that: “the higher the discriminability, the stronger the discriminative ability.”

Figures 7 and 8 compare the accuracy of the parameter settings $UNorm + UCN$, $Norm + UCN$, $UNorm + CN$, and $Norm + CN$ for different discriminability thresholds. The performance shows only minor improvement after normalizing discriminability. This is because the terms selected by the weighting discriminability already possess high discriminability. In other words, not many documents contain terms that are spread over

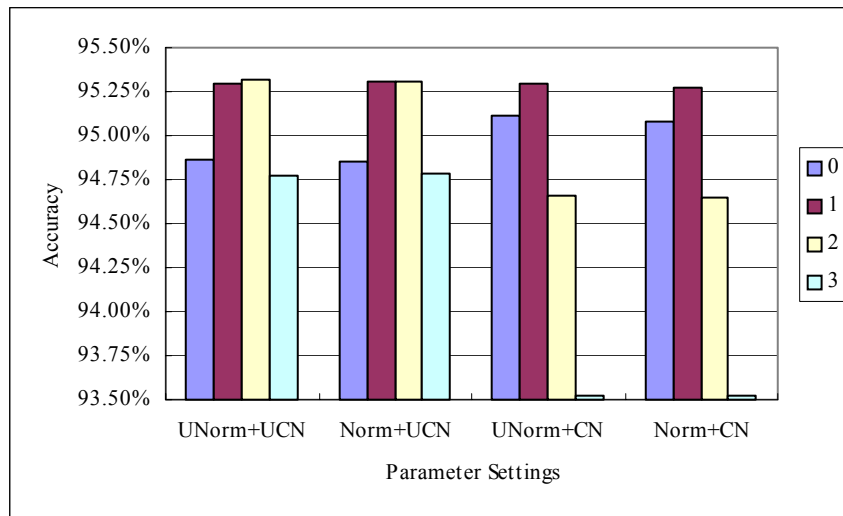


Fig. 6. Unknown word-based outside test on the first layer for comparing the accuracies of discriminability thresholds 0-3 for different parameter settings.

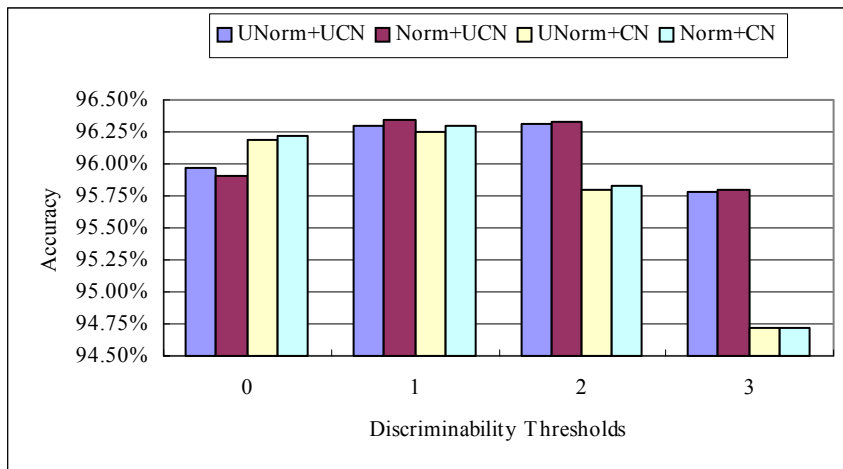


Fig. 7. Unknown word-based inside test on the first layer for comparing the accuracies of parameter settings UNorm+UCN, Norm+UCN, UNorm+CN, and Norm+CN for different Discriminability thresholds.

several categories widely and equally. The number constraints on the selected terms show no improvement in accuracy. These results show that extracted terms with high discriminability play irreplaceable roles in text categorization. Thus, the advantages of maintaining equivalence among categories are enough not to discard them.

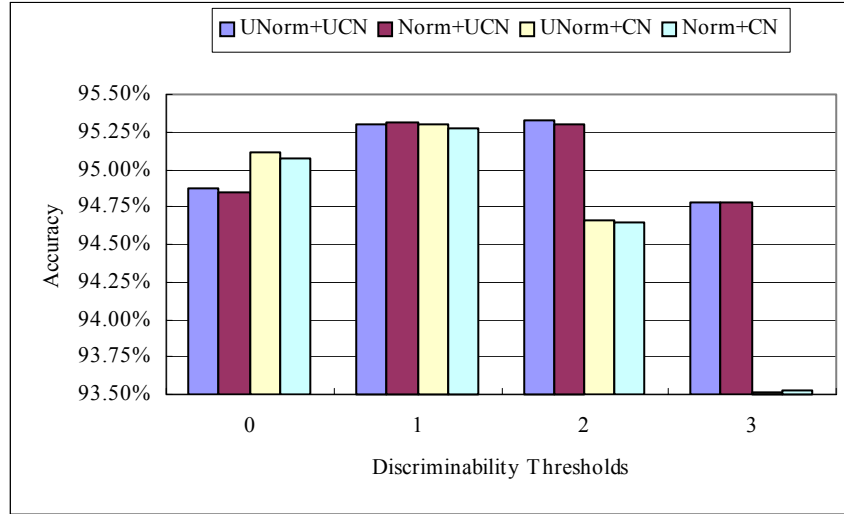


Fig. 8. Unknown word-based outside test on the first layer for comparing the accuracy of parameter settings UNorm+UCN, Norm+UCN, UNorm+CN, and Norm+CN for different Discriminability thresholds.

Table V. Experimental Results for Stopping-Words for Unpurified, Purified, and Fully Purified Levels of Purification

		<i>Unpurified</i>	<i>Stopping Words Purified Only</i>	<i>Fully Purified</i>
Number of Terms Needed		84,644	69,505	19,865
Inside Test	Culture & art	93.81%	94.29%	96.04%
	Entertainment	88.52%	89.50%	95.49%
	Medicine & health	94.73%	95.38%	98.17%
	Sports	89.90%	91.84%	95.94%
	Travel	98.79%	98.45%	96.07%
	Average accuracy	93.15%	93.89%	96.34%
Outside Test	Culture & art	91.35%	92.17%	94.07%
	Entertainment	86.59%	88.30%	95.13%
	Medicine & health	94.12%	94.60%	97.89%
	Sports	89.01%	90.96%	94.52%
	Travel	98.39%	97.71%	94.92%
	Average accuracy	91.89%	92.75%	95.31%

5.2.3 Experiments on the Purification Process

To evaluate the performance of the purification process, we conducted the following experiment. Strictly speaking, the last two steps of the purification process, i.e., steps 2 and 3, can be viewed as equivalent because both steps are used for eliminating overlapping and useless terms. Hence we compare the accuracy of three different levels of purification: unpurified, stopping words purified, and fully purified.

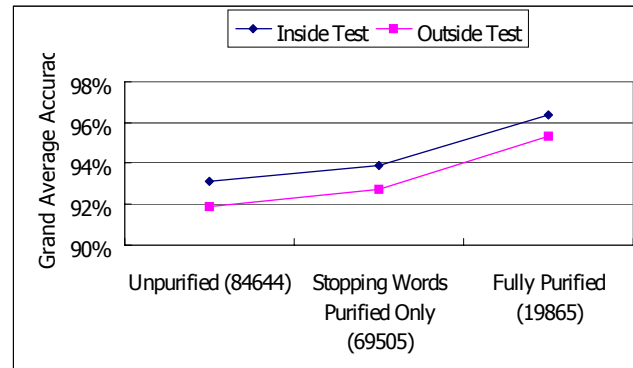


Fig. 9. Average accuracies comparing three different levels of purification: (1) unpurified, (2) only stopping words purified, and (3) fully purified levels.

This experiment was performed on the first layer of classification. The parameter combination *Norm+UCN* and a discriminability threshold of 1 were adopted. The details are listed in Table V. Comparing the number of needed terms between the unpurified level and the fully purified level, 64,779 terms are saved. That is, the dimensionality of the feature space can be reduced from 84,644 to 19,865. Figure 9 compares the average accuracies for the three different levels of purification. The overall purification process has around 3.42% improvement for inside testing and 3.72% improvement for outside testing.

5.2.4 Combined Approach

This experiment was performed on the second layer of classification. Figure 10 illustrates the average classification accuracy in the second layer. The average accuracy achieves 85.65% for inside testing and 85.07% for outside testing. The results show that for second layer performance, the unknown-word-based approach is better than the word-based approach, 78.63% for inside testing and 74.63% for outside testing (see Table II).

Compared to the first layer of unknown-word-based performance (96.34% accuracy for inside testing and 95.31% accuracy for outside testing), second layer performance is much worse. After checking the description vectors for some of the documents that were wrongly assigned, we found that the primary reason was that, for the unknown word-based approach, the number of representative terms of each category in the second layer is much fewer than that of an arbitrary category in the first layer. Even some of the text documents do not contain any representative terms in the second layer classification; that is, their description vectors are zero vectors. For this reason, we conducted the following experiment, which combines the unknown word-based approach with the word-based approach and compares TF×IDF weighting performance with discriminability weighting performance.

The experimental results are listed in Table VI. Clearly, the discriminability performance is superior to that of TF×IDF for both inside and outside tests. Compared to the unknown-word-based approach, the combined approach achieves better performance for second layer classification. This shows that there is a trade-off between the dimen-

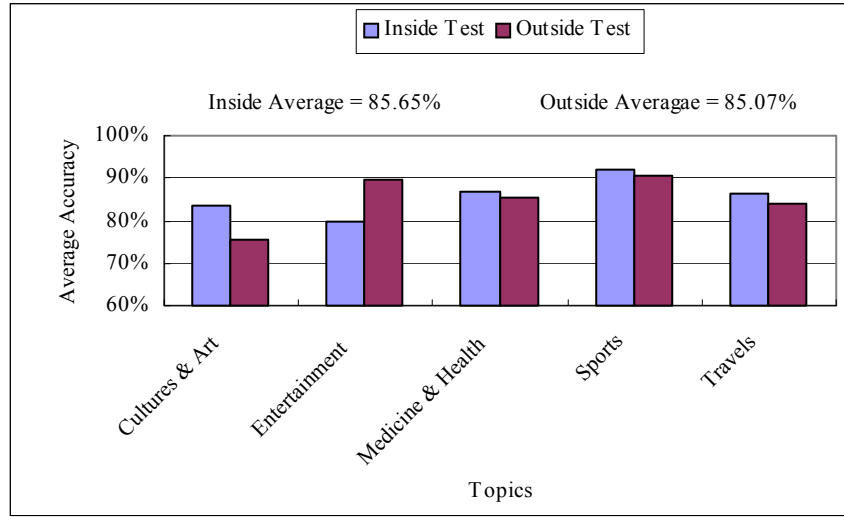


Fig. 10. Average accuracies for the unknown word-based classification in the second layer.

Table VI. Experimental Results for the Combined Approach Comparing TF×IDF and Discriminability

Term Weights		<i>Inside Test</i>		<i>Outside Test</i>	
		TF×IDF	Discriminability	TF×IDF	Discriminability
Topics	Culture & art	81.47%	85.76%	69.79%	78.10%
	Entertainment	77.92%	84.19%	87.29%	90.66%
	Medicine & health	84.90%	88.62%	79.86%	85.34%
	Sports	90.87%	94.84%	86.14%	95.46%
	Travel	83.83%	90.39%	80.72%	87.41%
Average accuracy		83.80%	88.76%	80.76%	87.40%

sionality of the feature space and the sparse data problem for the unknown-word-based approach.

5.2.5 Comparative Performance

In addition to statistical approaches, many learning methods are frequently used to construct text classifiers. Cohen and Singer [1999] compared two context-sensitive learning algorithms, set-valued RIPPER (Repeated Incremental Pruning to Produce Error Reduction) [Cohen 1995; 1996] and sleeping-experts [Freund et al. 1997], with other algorithms: Rocchio's algorithm [Rocchio 1971; Ittner et al. 1995], C4.5, and SWAP-1 on several benchmarks, the AP title corpus [Lewis and Catlett 1994; Lewis and Gale 1994], the TREC-AP corpus [Lewis et al. 1996], the Reuters-22173 corpus [Lewis 1992], and the Reuters-21578 collection. Their experimental results show that both context-sensitive algorithms generally perform better than algorithms that learn context-free classifiers.

Table VII. Accuracy (for outside test only) Comparisons for RIPPER2, Sleeping-Experts, Word + kNN, and PLU + kNN

	<i>RIPPER2</i>	<i>Sleeping-experts</i> (one word)	<i>Sleeping-experts</i> (four words)	<i>Word</i> + <i>kNN</i>	<i>PLU</i> + <i>kNN</i>
Culture & art	91.11%	87.32%	91.59%	90.93%	95.10%
Entertainment	90.33%	75.72%	79.61%	84.24%	96.54%
Medicine & health	85.46%	82.40%	86.53%	93.15%	98.79%
Sports	87.26%	85.79%	90.08%	85.52%	95.69%
Travel	83.53%	79.88%	83.87%	88.75%	95.86%
Average accuracy	87.54%	82.22%	86.34%	88.52%	96.40%

For RIPPER, the context of a word consists of a (usually small) number of other words that must co-occur with the word, but may occur in any order and in any location in the document. In contrast, sleeping-experts represents context by using sparse phrases. A sparse phrase is a sequence of nearby but not necessarily consecutive words. In others words, the context of a word for sleeping-experts consists of nearby words in a fixed order. More detailed descriptions of both can be found elsewhere [Cohen 1995; 1996; Freund et al. 1997].

We implemented a version of the two learning algorithms for evaluating their performance on the MSDN corpus. To save the symbols' representation space, Cohen [1996] extended RIPPER to allow the value of an attribute to be a set of symbols. That is, a document is represented with a single attribute, having as its value the set of words that appear in the document. However, this extension may reduce performance in some cases. To avoid this problem and simplify the implementation, this extension is omitted. Optionally, RIPPER can also include tests, called negative tests, of the form " $w_i \notin \text{feature}$ " in its rule. Negative tests actually improve its performance, shown by several experimental results in Cohen and Singer [1999]. Hence, negative tests are adopted here.

In the sleeping-experts framework, the weight associated with each phrase is learned online to minimize a cost function, in our case, accuracy. For a fair comparison with other methods, RIPPER and ours, we update the weights only during the training phase and keep them fixed during the testing phase. The sleeping-experts algorithm is based on a sparse n -gram model. When $n = 1$, the set of sparse phrases is exactly the set of words in the corpus. As in the experiment in Cohen and Singer [1999], we use the values $n = 1$ and 4 for evaluation, since a sparse n -gram model with large n needs a very high space requirement, which is almost impracticable.

We do not propose a particular text classifier, since this article focuses on term extraction and selection. For a fair comparison with the two context-sensitive learning algorithms, the traditional term word and the proposed term, PLU, were associated with a context-sensitive classifier, kNN (k -nearest-neighbor), which is context sensitive in the sense that no independence is assumed between either input variables (terms) or output variables (categories) [Yang and Pederson 1997]. Table VII shows the experimental results. In the evaluation of the proposed term plus the kNN classifier on the MSDN

Table VIII. Comparing TMT Proportions In a Variety of Topics

	<i>Topics</i>					<i>Average</i>	
	Culture & art	Entertainment	Medicine & health	Sports	Travel	Macro avg.	Micro avg.
Number of extracted terms	3752	12273	4840	22675	6165	-	-
Number of TMTs	2975	10309	4220	18603	4707	-	-
TMT proportions	79.29%	84.00%	87.19%	82.04%	76.35%	81.77%	82.11%

corpus, the average accuracy is 96.40%, outperforming all the other systems evaluated on the same collection, including the traditional kNN term word (88.52%) and sleeping-experts (82.22%), the four-word sparse phrase by sleeping-experts (86.34%), and boolean combinations of words by RIPPER (87.54%).

5.3 Meaningful Terms vs. Practicability

To evaluate how meaningful the extracted terms are, the following experiment was performed. We manually tagged all the terms extracted from 5 distinct categories, consisting of 6,165, 3,752, 22,675, 12,273, and 4,840 terms, respectively. If a term is meaningful, we call it a true meaningful term (TMT); otherwise we call it a false meaningful term (FMT). To measure the meaningful degree of the extracted terms, a TMT proportion P_{TMT} is defined as follows:

$$P_{TMT} = \frac{\text{the number of the TMTs}}{\text{total number of the terms}} \quad (18)$$

Table VIII compares the TMT proportions for different topics. The macro average and micro average of the TMT proportions are 81.77% and 82.11%, respectively. The result shows that over 80% of the extracted terms are meaningful. These are good results for an automatic term extraction method, especially when the terms are unknown words, including proper nouns, organizations, company names, personal names, verb phrases, and so on.

Additionally, we also want to know the relationship between TMT proportions and PLR. Figure 11 illustrates the variation of TMT proportions for PLRs ranging from 0.5 to 1.0. The results show that the TMT proportion increases as the PLU-based likelihood ratio increases. Moreover, most of the TMTs (83.63%) are PLUs when $PLR = 1.0$.

According to the experimental results described in Section 5.2.2, the parameter setting with discriminability normalization (*Norm*), without numerical constraints on selected terms (*UCN*), and with a discriminability threshold of 1 achieved the best accuracy. Table IX lists the TMT proportions of the selected terms in the best case. Table X compares best inside test accuracy with best outside test accuracy. For an easy comparison of Table IX with Table X, the results of both tables are combined into Figure 12 with two different vertical axes, the TMT proportion and the reduction ratio.

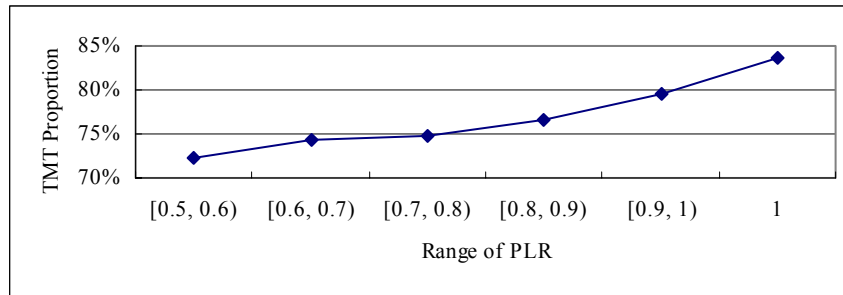


Fig. 11. Variation of the TMT proportions for PLR ranges from 0.5 to 1.

Table IX. Proportion of True Meaningful Terms (TMTs) Within Selected Terms for Each Category

	<i>Topics</i>				
	Culture & Art	Entertainment	Medicine & health	Sports	Travel
Number of selected terms	3191	3586	3360	5825	3903
Number of TMTs	2940	3110	3001	4715	3366
TMT proportions	92.13%	86.73%	89.32%	80.94%	86.24%

Table X. Classification Accuracy of Inside Testing vs. Outside Testing

	<i>Topics</i>				
	Culture & art	Entertainment	Medicine & health	Sports	Travel
Inside testing	96.04%	95.49%	98.17%	95.94%	96.07%
Outside testing	94.07%	95.13%	97.89%	94.52%	94.92%
Accuracy reduction ratio	2.05%	0.38%	0.29%	1.48%	1.20%

As seen in Figure 12, except for the test on the category “Culture and art,” we obtained the expected result: “the higher the TMT proportion, the lower the accuracy reduction from inside test to outside test.” This emphasizes an important point: that, when using compound words as terms in text categorization, extracting more meaningful terms is good for classifying untrained text documents in practice. The unexpected results for “Culture and art” may be due to the important terms in that category changing faster than the terms in other categories. This problem could be solved by term adaptation, as discussed in the next section.

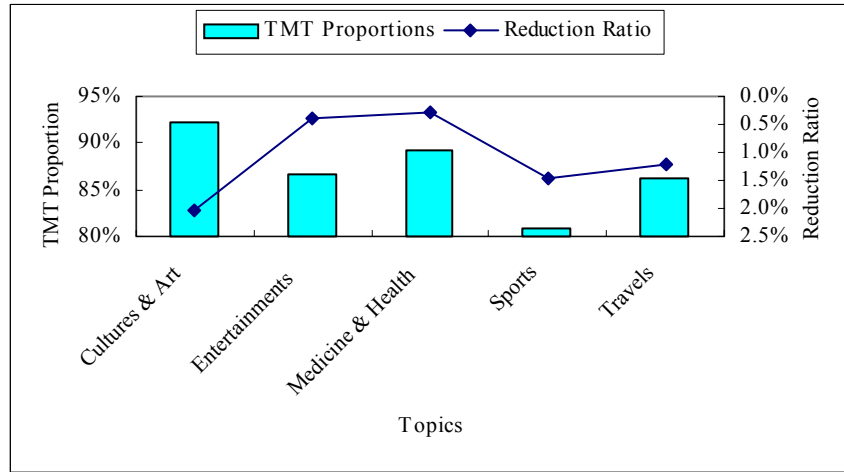


Fig. 12. TMT proportion against the accuracy reduction ratio for different topics.

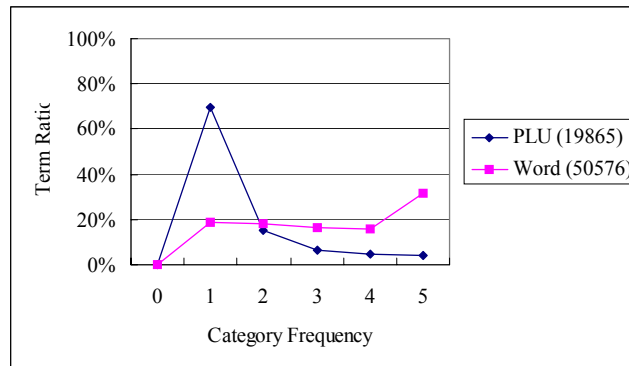


Fig. 13. Distribution of category frequencies in training data of first layer classification for PLUs and words.

5.4 Distribution of Category Frequencies

With respect to evaluating discriminative term selection, we also take into account the distribution of category frequencies. Figure 13 shows the distribution of category frequencies in the training data of the first classification layer for two different terms, PLUs and words. Since the number of PLUs (19,865) is different from the number of words (50,576), the vertical axis indicates the ratio of terms rather than the number of terms. The figure shows two important things. Most of the extracted PLUs (69.47%) occur in only one category. This indicates that the extracted PLUs are fairly domain-specific. In other words, most of the PLUs are discriminative. But 31.4% of words occurred in five categories.

Figure 14 shows the distribution of category frequencies in the testing data of the first layer classification for both PLUs and words. Two interesting things are notable: (1)

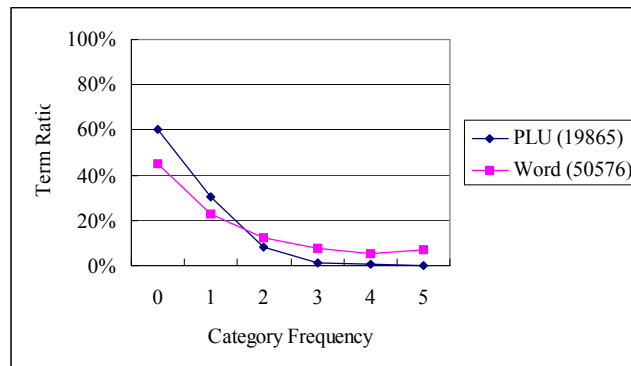


Fig. 14. Distribution of category frequencies in testing data of first layer classification for PLUs and words.

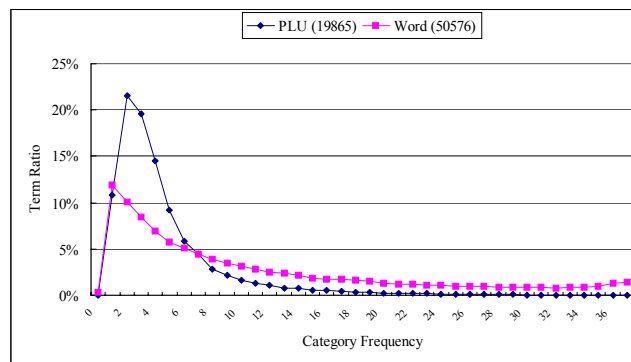


Fig. 15. Distribution of category frequencies in training data of second layer classification for PLUs and words.

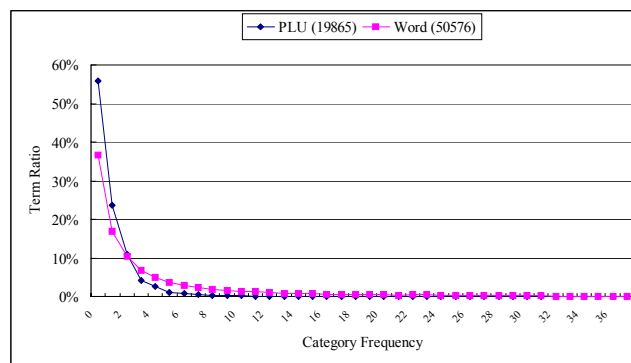


Fig. 16. Distribution of category frequencies in testing data of second layer classification for PLUs and words.

Table XI. Comparing the Ratio of Terms in Training Data Unseen in Testing Data for PLUs and Word n -Grams

	<i>PLUs</i>	<i>Word n-grams</i>					
		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
Number of terms in training data	19865	54169	2441756	5304471	5785308	5251466	4414285
Number of terms in training data unseen in testing data	11912	25690	2266661	5196903	5745398	5238426	4410055
Ratio of unseen terms (%)	59.96	47.43	9.83	97.97	99.31	99.90	99.96

as shown in Figure 13, PLU distribution is more concentrated than word distribution; (2) 60% of the PLUs are unseen in the testing data, which is higher than the ratio of unseen words (45%).

Figures 15 and 16 show the distribution of the category frequencies in the training and testing data for the second layer of classification for PLUs and words. For convenience, all the categories in second layer classification are mixed and the testing terms are the same as those in the first layer of classification. Clearly, the information shown in these two figures is the same as the experimental results of the first layer of classification.

5.5 Consistency between Training and Testing Data

Since consistency between training and testing data is very important for corpus-based approaches, the following experiment compares the ratio of word n -grams of $n=1, 2, \dots, 6$ occurring in training data but unseen in testing data with the ratio of PLUs occurring in training data but unseen in testing data. Table XI shows the experimental results. Each number of terms in the table is obtained by counting distinct terms. From the table, over 50% of the words occurring in the training data also occur in the testing data. Words have the best consistency between the training data and the testing data. However, n -grams for $n>1$ have a serious inconsistency problem. Around 40% of the PLUs are consistent. Thus, we see that meaningful terms such as words and PLUs maintain better consistency than n -grams for $n>1$. Additionally, although the PLUs are extracted from n -grams of $n=8$, the inconsistency problem for the PLUs is less serious than that for n -grams of $n=8$ (n -grams of $n=8$ inconsistency results are not shown in the table).

6. DISCUSSION

In addition to this article's two main themes, meaningful term extraction and discriminative term selection, the important topic of term adaptation is also discussed in the following section.

6.1 Meaningful Term Extraction

In corpus-based approaches, researchers usually separate the collected corpus into two sets, one for training and another for testing, and always attempt to train the proposed

system or model toward high performance. However, the system or model is constructed or trained by the training set. For this reason, the training set often dominates system performance. If the characteristics represented by the testing set are different from those represented by the training set, it tends to make system performance of the testing set worse than that of the training set.

Our approach is also based on a collected corpus. The most important features extracted from the corpus are terms. The terms are extracted from the training set and used in the testing set. So term consistency between the training and testing sets has a great effect on system performance in the real world. According to the experimental results in Figure 12, in some cases extracting more meaningful terms is good for reducing the inside test to outside test accuracy ratio. Moreover, in our experiments with the inconsistency problem, we found that meaningful terms like words and PLUs have better consistency compared with unmeaningful terms such as n -grams for $n > 1$.

6.2 Discriminative Term Selection

Another important task for textual-similarity-based text categorization is to select the most appropriate terms for indexing. It means the selected terms have to be able to discriminate one category from others.

A commonly used textual measure, TF×IDF [Salton and Buckley 1988], takes into account interdocument term distribution as well as intradocument term frequency. It is based on the assumption that if a term occurs in few documents, the term may more useful for distinguishing that document from others. The terms selected by TF×IDF has a high resolution for distinguishing documents, and so it is certainly good for text retrieval. However, the problem is that the documents represented do not necessarily belong to the same category. That is, distinguishing documents is not equal to distinguishing categories.

By contrast, in this article, the proposed term, PLU, is extracted from word-based n -grams of larger n , $n=8$, and selected according to the discriminability value. Since PLU is longer and discriminability is defined as the probability ratio of the represented category to others, the combination of both concentrates the selected terms in fewer categories, as confirmed by the experimental results in Section 5.4. In other words, the proposed terms, PLUs, are very discriminative in distinguishing categories.

6.3 Term Adaptation

This article uses unknown words and the high domain dependency characteristic in a textual-similarity-based approach to provide an effective method for text categorization. However, new words and new phrases are created every day, which are not only unseen in the directory but also unseen in the collected corpus. Therefore, seeking unknown words (unseen in the collected corpus) and adding them to the collection of terms becomes more important. On the other hand, some of the selected terms may become less important due to changing representations, usage, time, and so on. Thus, infrequently used terms should be removed from the term collection. Moreover, frequently used terms in some categories may change to other categories. For instance, Michael Jordan retired from basketball in 1993, but then returned to the spotlight in a baseball uniform in 1994. Thus, we believe that term adaptation is very important for text categorization due to the inevitable changes in our fast-changing world.

7. CONCLUSIONS AND FUTURE WORK

For high accuracy text categorization, we have proposed two new concepts, meaningful term extraction and discriminative term selection. The methodology is fully systematic and automatic. The experimental results show that the proposed features, PLUs, improve the performance of text categorization, even when using just a simple vector space model. The proposed purification process reduces the dimensionality of the feature space from 84,644 (the unpurified level) to 19,865. By contrast, the word-based approach that achieved a dimensionality of 50,576 is much higher than ours. Moreover, it was found that the proposed system achieved classification accuracies of 96.34% for inside testing and 95.31% for outside testing. In addition, evaluation of meaningful term extraction showed that over 80% of the automatically extracted terms were meaningful.

We draw the following conclusions from an analysis of the experimental results. First, meaningful term extraction is effective and practical for avoiding inconsistency problems and minimizing the reduction in the accuracy ratio. Second, to save time and manpower, terms should be extracted from the corpus automatically and systematically. Third, selecting the PLUs with high discriminability can, indeed, improve system performance.

In the future, we will work on two things. First, although the unknown word-based approach effectively reduces the dimensionality of feature space, the sparse data problem in the classification's second layer causes trouble. Hence we will attempt to find a way to reduce the influence of the sparse data problem. Second, because term adaptation is important due to the time-variant properties of textual terms, we will work on solving the problems of term adaptation.

ACKNOWLEDGMENT

The authors thank the National Science Council, Republic of China, for its financial support under contract NSC89-2218-E-006-029. We also thank Kam Fai Wong, Kim Teng Lua, and the anonymous reviewers for many constructive comments and suggestions.

REFERENCES

- AHO, A. V. AND CORASICK, M. J. 1975. Efficient string matching: An aid to bibliographic search. *Commun. ACM* 18, 6, 333-340.
- AOE, J. I. 1989. An efficient implementation of static string pattern matching machines. *IEEE Trans. Softw. Eng.* 15, 8, 1010-1016.
- BAASE, S. 1989. *Computer Algorithms*. 2nd ed. Addison-Wesley, 286-287.
- BENKHALIFA, M. AND BENSALID, A. 1999. Text categorization using the semi-supervised fuzzy c-means algorithm. In *Proceedings of the 18th International Fuzzy Information Conference of North American* (New York, NY, June), 561-565.
- CHANG, J. S., CHEN, S. D., KER, S. J., CHEN, Y., AND LIU, J. 1994. A multiple-corpus approach to recognition of proper names in Chinese text. *Comput. Process. Chinese Oriental Lang.* 8, 1, 75-85.
- CHEN, K. J. AND BAI, M. H. 1998. Unknown word detection for Chinese by a corpus-based learning method. *Comput. Linguist. Chinese Lang. Process.* 3, 1, 27-44.
- COHEN, W. W. 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning* (Tarragona, Spain, Nov.), 115-123.
- COHEN, W. W. 1996. Learning trees and rules with set-valued features. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI)*, 1, Portland, OR, Aug.), 709-716.
- COHEN, W. W. and SINGER, Y. 1999. Context-sensitive learning methods for text categorization. *ACM Trans. Inf. Syst.* 71, 2, 141-173.

- FARKAS, J. 1995. Towards classifying full-text using recurrent neural networks. In *Proceedings of the 1995 Canadian Conference on Electrical and Computer Engineering* (Montreal, Que., Sept.), 511-514.
- FARKAS, J. 1996. Improving the classification accuracy of automatic text processing systems using context vectors and back-propagation algorithms. In *Proceedings of the 1996 Canadian Conference on Electrical and Computer Engineering* (University of Calgary, Alberta, May), 696-699.
- FREUND, Y., SCHAPIRE, R. E., SINGER, Y., AND WARMUTH, M. K. 1997. Using and combining predictors that specialize. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing* (El Paso, TX, May), 334-343.
- HAYASHI, Y. AND MOCHIZUKI, H. 1999. An efficient method of determining relationships among compound keywords using machine-AC. In *Proceedings of the 4th International Workshop on Information Retrieval with Asian Languages (IRAL '99, Academia Sinica, Taipei, Taiwan, Nov.)*, 91-96.
- ITTNER, D. J., LEWIS, D. D., AND AHN, D. D. 1995. Text categorization of low quality images. In *Proceedings of the Symposium on Document Analysis and Information Retrieval* (Las Vegas, NV, April), 301-315.
- JACOBS, P. S. 1993. Using statistical methods to improve knowledge-based news categorization. *IEEE Expert*, 8, 13-23.
- JOACHIMS, T. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML '97, Nashville, TN, July)*, 143-151.
- JO, T. C. 1999. Text categorization with the concept of fuzzy set of informative keywords. In *Proceedings of the 1999 IEEE International Conference on Fuzzy Systems* (Seoul, Korea, Aug.), 609-614.
- KHAN, I. AND CARD, H. C. 1997. Personal adaptive Web agent: A tool for information filtering. In *Proceedings of the 1997 Canadian Conference on Electrical and Computer Engineering* (St. Johns, Newfoundland, May), 305-308.
- KIM, J. T. AND MOLDOVAN, D. I. 1995. Acquisition of linguistic patterns for knowledge-based information extraction. *IEEE Trans. Data Eng.* 7, 5, 713-724.
- LAI, Y. S. AND WU, C. H. 2000. Unknown word and phrase extraction using a phrase-like-unit-based likelihood ratio. *Int. J. Comput. Process. Oriental Lang.* 13, 1, 83-95.
- LAM, L. Y. AND LEE, D. L. 1999. Feature reduction for neural network based text categorization. In *Proceedings of the 6th International Conference on Database Systems for Advanced Applications* (Taiwan, April), 195-202.
- LARKEY, L. S. AND CROFT, W. B. 1996. Combining classifiers in text categorization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-96, Zurich, Switzerland, Aug.)*, 289-297.
- LEWIS, D. D. 1992. Representation and learning in information retrieval. Ph.D. dissertation, Dept. of Computer Science, University of Massachusetts, Amherst, MA.
- LEWIS, D. D. AND CATLETT, J. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the 11th International Conference on Machine Learning* (New Brunswick, NJ, July), 148-156.
- LEWIS, D. D. AND GALE, W. 1994. Training text classifiers by uncertainty sampling. In *Proceedings of the 17th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR-9, Dublin, Ireland, July)*, 3-12.
- LEWIS, D. D., SCHAPIRE, R., CALLAN, J. P. AND PAPKA, R. 1996. Training algorithms for linear text classifiers. In *Proceedings of the 19th Annual ACM International Conference on Research and Development in Information Retrieval (SIGIR-96., Zurich, Switzerland, Aug.)*, 298-306.
- LI, B. I., LIN, S., SUN, C. F., AND SUN, M. S. 1991. A maximal matching automatic Chinese word segmentation algorithm using corpus tagging for ambiguity resolution. In *Proceedings of the Conference on Research on Computational Linguistics* (Taiwan, Aug.), 135-146.
- LI, Y. AND JAIN, A. K. 1998. Classification of text documents. In *Proceedings of the 14th International Conference on Pattern Recognition* (Brisbane, Australia, Aug.), 1295-1297.
- LIANG, T. AND YE, D. R. 2000. Using word formation principles and neural network for unknown word extraction. In *Proceedings of the XIII Conference on Research in Computational Linguistics* (Taipei, Taiwan, Aug.), 21-40.
- LIU, K. C. Y. AND LIANG, N. 1989. On methods of Chinese automatic word segmentation. *J. Chinese Inf. Process.* 3, 1, 13-20.
- MANNING, C. D. AND SCHÜTZE, H. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA, 575-577.
- MAY, A. D. 1997. Automatic classification of e-mail message by message type. *J. Am. Soc. Inf. Sci.*, 32-39.

- MURTHY, K. R. K. AND KEERTHI, S. S. 1999. Context filters for document-based information filtering. In *Proceedings of the 5th International Conference on Document Analysis and Recognition* (Bangalore, India, Sept.), 709-712.
- NIE, J. Y., HANNAN, M. L., AND JIN, W. 1995. Unknown word detection and segmentation of Chinese using statistical and heuristic knowledge. *Commun. COLIPS*, 5, 1.
- OJA, E. 1983. *Subspace Methods of Pattern Recognition*. Wiley, New York, NY.
- ROCCHIO, J. J. 1971. Relevance feedback information retrieval. In *The Smart Retrieval System – Experiments in Automatic Document Processing*. Prentice Hall, Englewood Cliffs, NJ, 313-323.
- SALTON, G. AND BUCKLEY, C. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.* 24, 5, 513-523.
- SASAKI, M. AND KITA A, K. 1998. Rule-based text categorization using hierarchical categories. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics* (San Diego, CA, Oct.), 2827-2830.
- SCHÄUBLE, P. 1997. *Multimedia Information Retrieval: Content-Based Information Retrieval from Large Text and Audio Databases*. Kluwer Academic, Amsterdam, The Netherlands, 49-59.
- SCHÜTZE, H., HULL, D. A. AND PEDERSEN, J. O. 1995. A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-95)*, Seattle, WA, July), 229-237.
- SUN, M. S., HUNAG, C. N., GAO, H. Y. AND FANG, J. 1994. Identifying Chinese names in unrestricted texts. *Commun. COLIPS* 4, 2, 113-122.
- TSAY, J. J. AND WANG, J. D. 1999. Term selection with distributional clustering for Chinese text. In *Proceedings of the XII Conference on Research in Computational Linguistics* (Hsinchu, Taiwan, Aug.), 151-170.
- YANG, Y. AND PEDERSEN J. O. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML-97)*, Nashville, TN, July), 412-420.

Received August 2000; revised January 2001; accepted October 2001.