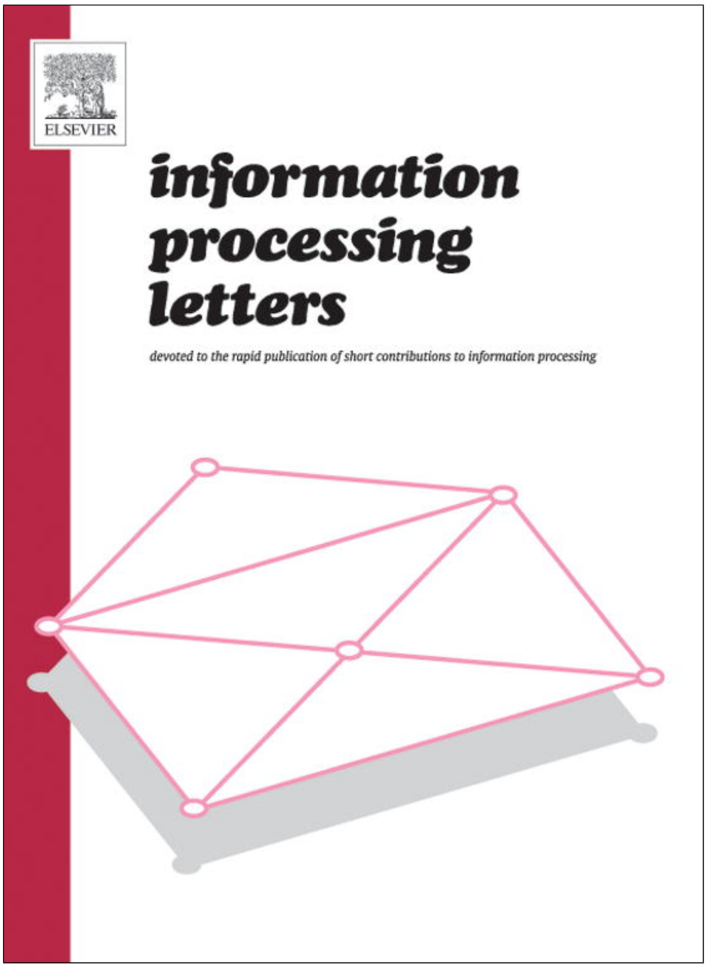


ELSEVIER
information processing letters
devoted to the rapid publication of short contributions to information processing

Information Processing Letters 113 (2013) 137–144

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Free-gram phrase identification for modeling Chinese text

Xi Peng, Zhang Yi [*], Xiao-Yong Wei, De-Zhong Peng, Yong-Sheng Sang

Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, 610065, China

ARTICLE INFO

Article history:
Received 29 June 2011
Received in revised form 11 September 2012
Accepted 23 November 2012
Available online 28 November 2012
Communicated by W.-L. Hsu

Keywords:
Information retrieval
Document representation model
Bag of phrase
Text categorization
Sparse coding
Unfixed gram phrase identification

ABSTRACT

Vector space model using bag of phrases plays an important role in modeling Chinese text. However, the conventional way of using fixed gram scanning to identify free-length phrases is costly. To address this problem, we propose a novel approach for key phrase identification which is capable of identify phrases with all lengths and thus improves the coding efficiency and discrimination of the data representation. In the proposed method, we first convert each document into a context graph, a directed graph that encapsulates the statistical and positional information of all the 2-word strings in the document. We treat every transmission path in the graph as a hypothesis for a phrase, and select the corresponding phrase as a candidate phrase if the hypothesis is valid in the original document. Finally, we selectively divide some of the complex candidate phrases into sub-phrases to improve the coding efficiency, resulting in a set of phrases for codebook construction. The experiments on both balanced and unbalanced datasets show that the codebooks generated by our approach are more efficient than those by conventional methods (one syntactical method and three statistical methods are investigated). Furthermore, the data representation created by our approach has demonstrated higher discrimination than those by conventional methods in classification task.

1. Introduction

Chinese is considered one of the most popular languages in international communication due to its large number of speakers. However, data representation model (or document indexing model) has seldom been built specifically for Chinese text, but mainly borrowed from those of English instead. A generally employed one is vector space model [9,7] which represents each document as a feature vector that encapsulates statistical characteristics (e.g., frequency) of a set of terms in the document. The most popularly used paradigm of vector space model might be bag of words (BoW), which defines the terms as the selected words from a certain corpus. A critical drawback of BoW is that words are treated independently even their occurrences are highly correlated in reality. Therefore, it has apparently limited the applicability of BoW to any uninflected language like Chinese that conveys meaning through word order.

Bag of phrase (BoP) model [1], which defines terms as selected phrases (multi-word strings) and thus be capable of remaining word order to a certain extent, is seemingly a more reasonable choice in this case. Nonetheless, there are rarely encouraging results with BoP reported in literature except those in [1,13,8,27,10], due to the difficulty in the selection of phrases. The most intuitive way for phrase selection is to select syntactical phrases, namely the fixed expressions of the language under investigation (e.g., idioms) available from dictionaries [21,24,26]. As widely reported in [12,20], a term set consisting of syntactical phrases may not be an efficient "codebook" for tasks such as information retrieval and classification, because those phrases do not carry enough statistical information and the codebook is not easy to be updated. Statistical phrases, which are selected from multi-word strings that are of statistically significant in a given corpus, are thus commonly utilized.

[*] Corresponding author.
E-mail address: zhangyi@scu.edu.cn (Z. Yi).

0020-0190/$ – see front matter Crown Copyright © 2012 Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ipl.2012.11.005

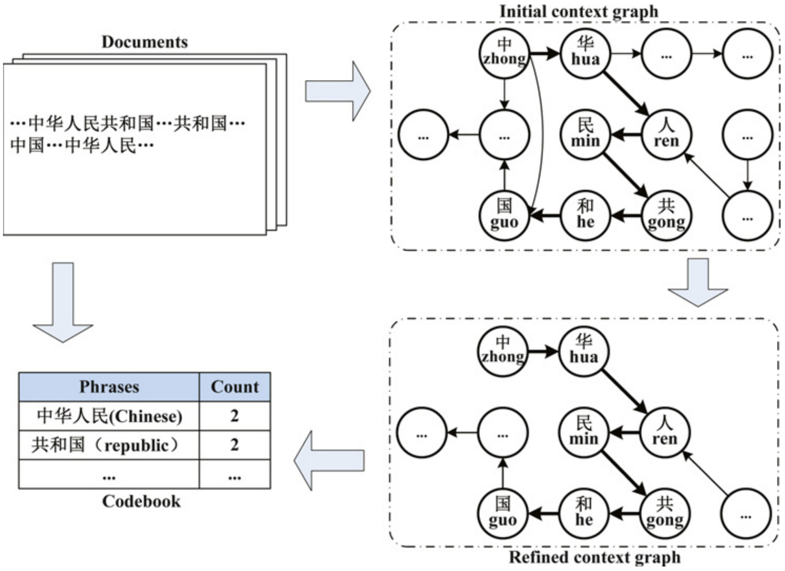


Fig. 1. Framework of free-gram phrase identification.

However, to identify statistical phrases, we need to study the statistical significance of multi-word strings with various lengths (known as full-text N-gram search), which is an extremely costly process, since it is usually impractical to investigate all possible N's (i.e., all possible lengths) in real applications. Therefore, bigram ($N = 2$) and trigram ($N = 3$) [28,15] are widely used in real application. However, given the fact that most of the fixed expressions of Chinese are with lengths larger than 3 (e.g., most idioms are composed of 4 words), neither bigram nor trigram search can form a compact and efficient codebook.

In this paper, we propose a free-gram phrase generation approach to address the difficulty of phrase selection. As shown in Fig. 1, the proposed approach starts by constructing a context graph for each document, where each node is a single word and each directed edge represents the presence of a 2-word string sequence with its frequency indicated by the edge width. By filtering out edges that are not statistically significant, each transmission path in the resulting graph indicates a hypothesis of a fixed expression (e.g., zhong-hua-ren-min-gong-he-guo (The people's republic of China)). A refinement step is then followed by simply referring to the original document to validate each hypothesis and decide whether to select the hypothesis directly or to sub-divide it into several substrings as phrase(s) with the hope of maximizing the efficiency of the coding. It is obvious that the use of context graph in our approach has avoided brute-force search for all possible lengths, significantly reduced the size of search space, and increased the possibility of identifying statistical significant phrases of various lengths. The resulting phrases can thus represent the semantic content of the documents more accurately and forms a codebook with representative phrases to fulfill the requirements of compactness and efficiency. This has been confirmed by the experimental analysis in Section 4.1 with a perspective of sparse coding [16].

The rest of this paper is organized as follows. In Section 2, we review the existing work of vector space model. Section 3 introduces construction the context graph and its usage for phrase identification. The experimental results are given in Section 4 and Section 5 finally concludes this paper.

2. Related work

Although text modeling has been studied over decades with numerous approaches proposed, vector space model (BoW or BoP) is considered by many researchers as the most successful one among others. From the view that BoW is a special case of BoP (each word can be treated as a one-word phrase), the major issue of vector space model is indeed how to select representative phrases. Existing methods along this direction can be roughly grouped into two types: dictionary-based and statistics-based approaches.

Dictionary-based approaches extract phrases on the basis of a sophisticated word segmentation rules and a predetermined dictionary, trying to retain the syntactical characteristics of the phrases as much as possible. For instance, in [26], the authors propose a method named ICTCLAS which forms Chinese phrases via a pre-defined dictionary and a hierarchical hidden Markov model constructed based on a set of grammatical rules; Collobert et al. do similar work by training a convolutional neural network on a dataset of English documents [2].

Statistics-based approaches select phrases by utilizing statistical information of word sequence, where an N-gram scanning is usually conducted on every document to find phrases from N-word strings that are statistically significant. For instance, in [11,3,4], phrases are extracted by fixed gram scanning based on the probability that the occurrence of a specific word may be affected by its immediately preceding words; Zhang et al. select statistically significant phrases via mutual information among words [27].

Even empirical comparisons among various selection schemes are widely reported [27,10], there is seemingly no conclusion having been reached about which ones are better than others. However, we may borrow some selection criteria from the theory of sparse coding [6,18], on the basis of the perspective that the text modeling is in fact a coding process. Within the scenario of BoP, to obtain a "sparse" codebook, what we should have are (1) every document can be represented with a small number of phrases (codes); (2) every phrase in the book is efficiently utilized during the coding [16]. It is easy to see that dictionary-based approaches are helpful to select phrases that have distinctive meanings so as to reduce the number of phrases needed to represent each document, but however fail to utilize these phrases efficiently by distinguishing them by their statistical significance. On the contrary, statistics-based approaches might be able to use the phrases discriminatively but meet the difficulty in selecting representative phrases.

The approach proposed in this paper has indeed provided a compromised solution for fulfilling the requirements of sparse coding, where we are possible to find the phrases with all lengths effectively and at the same time utilize phrases according to their statistical significance. The experimental results in Section 4 show that our approach can generate more sparse codes than the fixed gram methods and outperforms both the representatives of dictionary-based and statistics-based approaches in text categorization task. Furthermore, we have conducted comprehensive experiments on both balanced and unbalanced datasets and with various models of classifiers to validate that the superiority of our approach is invariant to datasets and classification models empirically.

3. Free-gram phrase identification

To identify the phrases with all lengths and avoid performing reclusive full-text scanning on the target document, the proposed approach only needs to scan the document once to construct a context graph encapsulating both statistical and positional information of the phrases. With the guide of the context graph, we can extract the candidate phrases with a well-targeted manner (compared to full-text scanning). A refinement step is then conducted on the candidates to find the resulting phrases. The details will be given as follows.

3.1. Construction of context graph

The context graph is indeed a directed graph with each node representing a Chinese word and each edge representing a 2-word string consisting of words at its ends. To facilitate the analysis, we use the frequency of the 2-word string as the weight of corresponding edge, and to find each word in the target document effectively, we attach each node a linked list recording the positions of the word in the document. An example of the context graph is shown in Fig. 1.

The algorithm for constructing the context graph is summarized as follows: (1) linearly scan the target document word by word; (2) create a node and an empty linked list for current word if necessary; (3) add the position of current word to the linked list; (4) increase frequency the 2-word string consisting of the predecessor word (if exists) and current word by 1; (5) set current word as the predecessor word and move to the next word; (6) repeat until reach the end of the document. To be more accurate, the pseudo-code of the construction algorithm is shown in Algorithm 1. Once the graph has been constructed, we need to filter out those non-significant 2-word strings by removing the corresponding edges. It can be done by simply checking the frequency of each edge and delete the edge (and the predecessor node) if its frequency is smaller than a threshold (empirically learned or adaptively determined). It is easy to see that the resulting graph has included all 2-word strings which are statistically significant. Furthermore, each transmission path with X nodes indicates a hypothesis that corresponding X-word string is a statistically significant phrase in the document (e.g., zhong-hua-ren-min-gong-he-guo in Fig. 1). (See Fig. 2.)

Algorithm 1 Construction of context graph

```
Input: $d$ and $\eta$, where $d$ is a target document and $\eta$ is a threshold to
   determine whether a 2-word string is statistically significant
Output: The context graph $G$
for the $i$th word $t_i$ in target document $d$ do
    if $t_i \notin G$ then
        create a node and an empty list $\ell$ to record the positions of $t_i$
        in $d$;
        add the edge $\{t_{i-1}, t_i\}$ from node $t_{i-1}$ to $t_i$, and set its weight
        to 1;
    else
        add the current position $i$ into $\ell$
        if edge $\{t_{i-1}, t_i\} \in G$ then
            increase the weight of $\{t_{i-1}, t_i\}$ by 1;
        else
            add edge $\{t_{i-1}, t_i\}$ from node $t_{i-1}$ to $t_i$, and set its weight
            to 1;
        end if
    end if
end for
for each edge $e \in G$ do
    if the weight of $e < \eta$ then
        remove $e$ from $G$;
    end if
end for
```

3.2. Extraction of candidate phrases

In this section, we extract candidate phrases by crossly referring to original document and the context graph. We select a X-word phrase in original document as a candidate if there is a corresponding hypothesis in the context graph. To be computationally effective, the procedure is as follows: (1) find a node with indegree equaling to zero; (2) following the linked list of the node to find a starting position i and create an empty string S; (3) connect the word t_i in the original document to the string (i.e., $S = S \cup \{t_i\}$) if there is a hypothesis for the resulting S in the context graph; (4) set $i = i + 1$ and repeat (3) until there is no hypothesis found in the context graph; (5) select S as a candidate phrase and move the next starting position of the linked list.

3.3. Codebook generation

Intuitively, the resulting candidate phrases $\mathcal{P}$ is the best to represent the content of the target document and can

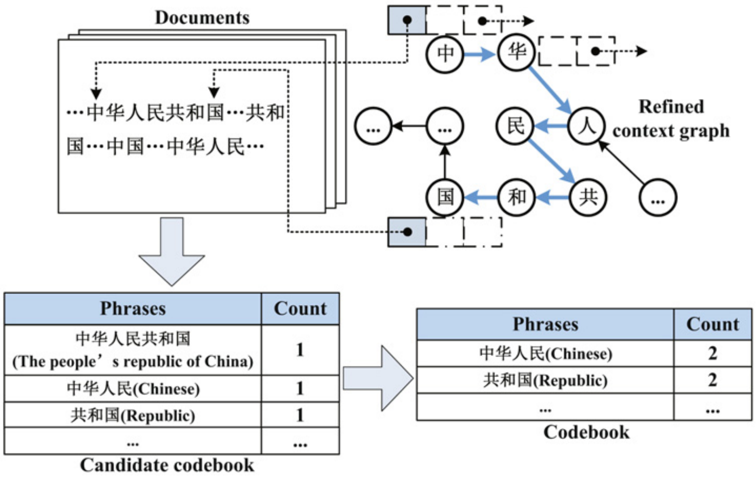


Fig. 2. Procedure of phrase identification: (1) After scanning the document, the statistical and positional information of words are stored in a context graph with each node associated with a linked list; (2) Candidate phrases are identified by crossly referring to the document and the context graph; (3) Complex phrases are divided into sub-phrases to improve the efficiency of the coding.

Algorithm 2 Extraction of candidate phrases

```
Input: target document $d$, context graph $G$
Output: candidate phrase set $\mathcal{P}$
 1: find an unprocessed node $t$ with its indegree equaling to 0 and its position $\ell$;
 2: for each position $i \in \ell$ do
 3:     set string $S = \{t_i\}$, $j = i$;
 4:     if $t_{j+1}$ has not been processed, $t_{j+1}$ is not a punctuation and edge $\{t_j, t_{j+1}\} \in G$ then
 5:         $S = S \cup \{t_{j+1}\}$, mark $t_{j+1}$ as processed in $d$, set $j = j + 1$, and goto step 4;
 6:     end if
 7:     add $S$ into $\mathcal{P}$ and mark $t_i$ as processed in $d$;
 8: end for
 9: mark node $t$ as processed in $G$;
10: goto step 1 if there are unprocessed nodes with indegrees equaling to 0;
```

be combined into the codebook directly. However, to improve the efficiency of the coding, we propose to split every long phrase into several sub-phrases if and only if any one of its substrings is also a candidate phrase, i.e., we divide a candidate phrase $C = \{w_0, \ldots, w_i, \ldots, w_j, \ldots, w_n\}$ into three sub-phrases $\{w_0, \ldots, w_{i-1}\}$, $\{w_i, \ldots, w_j\}$ and $\{w_{j+1}, \ldots, w_n\}$ if and only if $\{w_i, \ldots, w_j\}$ is also a candidate phrase. The rationale of the phrase splitting is based on the fact that sub-dividing a complex pattern into several sub-patterns that are also statistically significant can increase the entropy of the coding scheme. This is exactly consistent with our expectation for achieving a codebook fulfilling the requirements of sparse coding [6], because on one hand the codebook consisting of phrases with unfixed lengths enables us represents each document with only a small number of phrases (see Section 2), and on the other hand the increase of entropy makes sure that every code has been efficiently utilized in coding.

4. Experiments

In this section, we evaluate the efficiency and effectiveness of our approach for free-gram phrase indexing (**FreeG**) in comparison with 4 popular indexing models including:

- **ICTCLAS** [26] representing syntactical BoP model;
- **BoW** representing 1-gram statistical BoP;
- **Bigram**[1] representing 2-gram statistical;
- **Trigram**[1] representing 3-gram statistical BoP.

To investigate the dataset dependency of the five approaches, all the experiments have been conducted on both balanced dataset SogouC[2] with 80,000 documents and unbalanced dataset TanCorp60 [23] with 14,150 documents. SogouC, which includes 10 groups with equal size, is used to examine the performance of our model in balanced data distribution, i.e., the samples size per subject is equal. Alternatively, TanCorp60 contains 14,150 news files which are categorized into 60 groups, where the minimum group includes 19 files and the maximum group includes 1317 files. This database is used to investigate the performance of our model in unbalanced case. To the best of our knowledge, these two data sets are two of the most popular Chinese corpuses [25,22]. SogouC is well known owing to its large scale and balanced distribution, while TanCorp60 is popular since its extreme unbalanced distribution. Consequently, these two corpuses could model well the situations in practical application. For example, another popular unbalanced corpus, Fudan corpus, which contains 19,637 files distributed over 20 groups, where the minimum group includes 52 files and the maximum group includes 2507 files. We can see that it has fewer subjects and distributes much smooth than TanCorp60. Moreover, the balanced corpus, the TREC-5 People's Daily Corpus, consists of 33,047 documents over 6 subjects, and each subject has equal size. We can see that SogouC is more challenging than TREC-5 since the former has more subjects and samples for each group. In the following sections, we will compare the 5 approaches in three aspects: the coding efficiency in data representation, discrimination in classification task, and time cost.

[1] http://homepages.inf.ed.ac.uk/lzhang10/ngram.html.

[2] http://www.sogou.com/labs/dl/tce.html.

Table 1
The coding efficiency of different methods over SogouC and TanCorp60. n denotes the average number of the atoms per document; TR denotes Treves–Rolls metric.

Metrics	SogouC					TanCorp60				
	FreeG	BoW	ICTCLAS	Bigram	Trigram	FreeG	BoW	ICTCLAS	Bigram	Trigram
# codebook	190342	5189	119529	385654	923181	67982	4300	73448	179243	321365
n	27.90	91.48	69.48	86.10	47.67	27.31	91.96	70.92	88.42	48.58
TR	5.40E-3	7.26E-2	7.13E-4	6.90E-3	2.07E-2	7.30E-3	9.16E-2	1.00E-2	1.55E-2	5.73E-2
Kurtosis	1.28E+4	9.66E+1	6.88E+4	9.20E+3	1.14E+4	6.06E+3	8.39E+1	4.23E+4	3.00E+3	2.68E+3

Table 2
Precision (mean ± std. deviation) of different classifiers with five DRMs on two datasets using 20-fold cross validation.

Dataset	Classifier	DRMs				
		FreeG	BoW	ICTCLAS	Bigram	Trigram
SogouC	CNB	**0.9454 ± 0.0052**	0.8808 ± 0.0051	0.9103 ± 0.0052	0.9201 ± 0.0036	0.9139 ± 0.0049
	SVM	**0.9116 ± 0.0056**	0.8842 ± 0.0077	0.8909 ± 0.0049	0.9011 ± 0.0045	0.8705 ± 0.0059
	NBM	**0.9392 ± 0.0058**	0.8845 ± 0.0062	0.9104 ± 0.0052	0.9180 ± 0.0040	0.9165 ± 0.0055
TanCorp60	CNB	**0.7973 ± 0.0124**	0.6550 ± 0.0119	0.7238 ± 0.0134	0.7721 ± 0.0107	0.7561 ± 0.0104
	SVM	0.7493 ± 0.0064	0.7313 ± 0.0139	0.6889 ± 0.0125	**0.7519 ± 0.0097**	0.7059 ± 0.0141
	NBM	**0.8057 ± 0.0135**	0.7851 ± 0.0111	0.7132 ± 0.0096	0.7805 ± 0.0124	0.7642 ± 0.0089

4.1. Comparison of efficiency in coding

After the documents have been indexed with various models, we collect following statistics to investigate the coding efficiency: (1) number of phrases in codebook, which indicates the compactness of the vocabulary set; (2) average number of phrases for representing each document, which indicates the representativeness of the extracted phrases; and (3) Treves–Rolls sparseness measure and kurtosis (the fourth statistical moment of a distribution) [18], which are commonly adopted metrics to measure the degree that corresponding model has fulfilled the requirements of sparse coding. The smaller the value of Treves–Rolls metric, the better the sparseness, while the bigger the value of kurtosis, the better the sparseness. Actually, the last three metrics provide different aspects to study the sparseness of DRMs. The results on SogouC and TanCorp60 have been shown in Table 1.

The size of the codebook of ICTCLAS which uses syntactical phrases can be treated as an ideal case, because syntactical phrases are usually the best to reflect the semantic content of a document. It is obviously that FreeG is the most close one to ICTCLAS in terms of codebook size, an indication that FreeG can also capture the syntactical structure of Chinese to a certain extent, this conclusion is also supported by similar results on Treves–Rolls metric and kurtosis. Moreover, even codebook of BoW seems to have smaller size than the rest of 4 models, it apparently needs more phrases for representing each document. It thus reveals the lack of representativeness of this commonly used 1-gram model. Our proposed model FreeG, which requires the smallest number of phrases for each document, has demonstrated the best representativeness in this point of view, especially when compared with ICTCLAS that has created an ideal codebook but needs more phrases for each document due to the lack of statistical information to improve the coding efficiency. Trigram model seems to be the second best in coding, which might be due to the fact that it carries more syntactical information than Bigram and BoW.

4.2. Comparison of classification performance

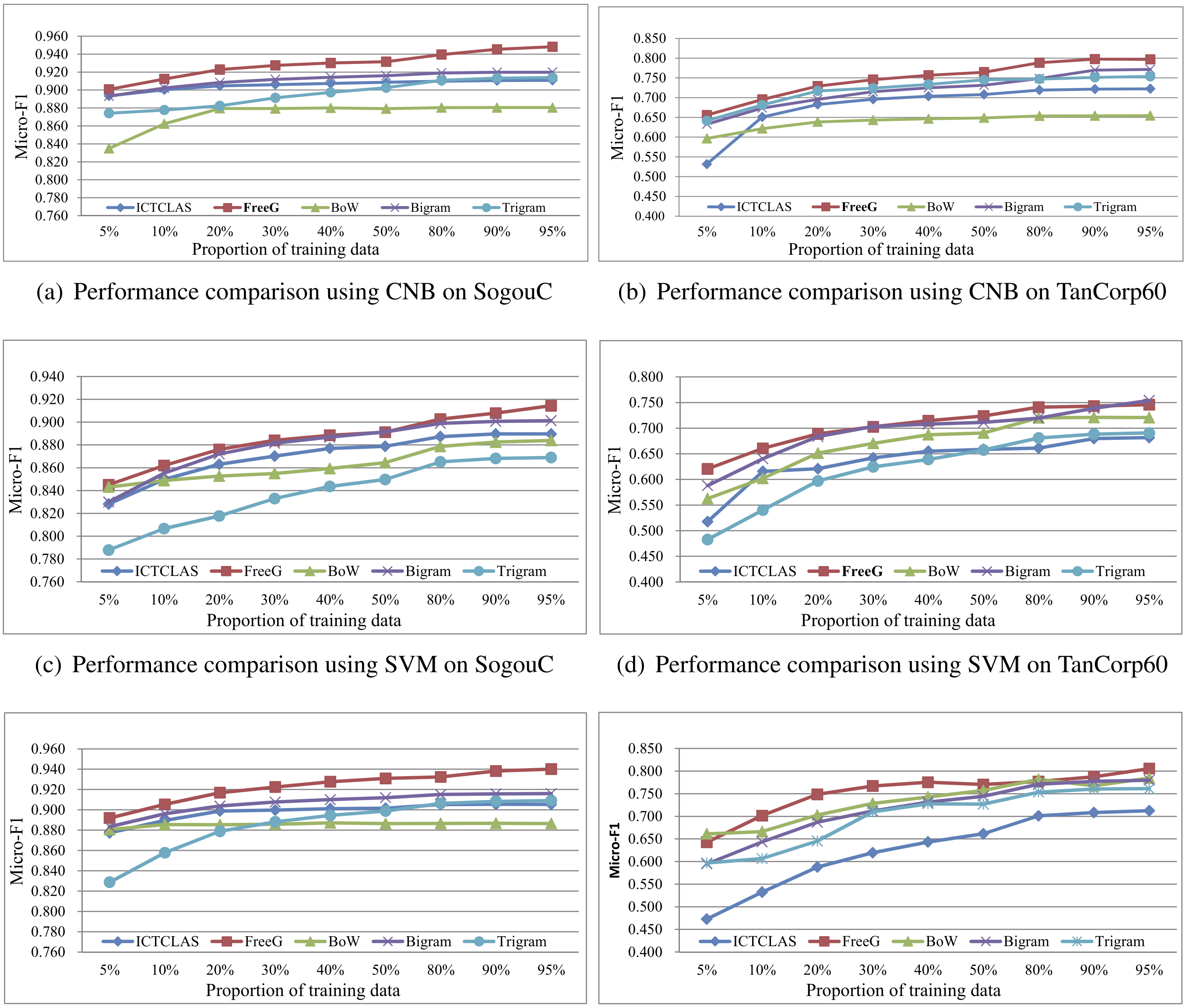
To study the discrimination of different indexing models, we used the feature vectors after indexing for classification task, under the assumption that the discrimination of a coding scheme can be indicated by its performance in distinguishing one document category from the others. Ten categories defined on SogouC and sixty categories defined on TanCorp60 have been used in the experiments, with micro-averaging Precision, Recall and F1-measure [7] as metrics for classification performance. Note that micro-averaging F1-measure gives an equal weight to each document but each category as macro-averaging F1-measure does. It means that micro and macro F1 scores are same with respect to balanced data, while micro F1 is much better to unbalanced data.

To avoid the case that the results are biased to any classification model, three types of classifiers have been experimented including naive Bayes multinomial (NBM) [14], complement naive Bayes (CNB) [17] and SVM with linear kernel [5], which are popularly adopted in text categorization in literature. All the classification experiments have been conducted in manner of k-fold cross validation [7], where we can vary the k to control the proportion of training data.

Tables 2 and 3 report the micro-averaging Precision and Recall scores of the evaluated methods, respectively, where the best result on each testing case is shown in bold face. We report the performance of five DRMs using widely-used 20-fold cross validation (95% training samples), and calculate the mean and standard deviation of each metrics. Here, micro Recall is equivalent to micro F1-measure since each sample is always classified to belong to one of classes. If one would have some samples which are not classified to belong to any of known classes, the micro Recall would differ from micro F1-measure. From the results,

Table 3
Recall (mean ± std. deviation) of different classifiers with five DRMs on two datasets using 20-fold cross validation.

Dataset	Classifier	DRMs				
		FreeG	BoW	ICTCLAS	Bigram	Trigram
SogouC	CNB	**0.9450 ± 0.0053**	0.8805 ± 0.0049	0.9102 ± 0.0050	0.9198 ± 0.0036	0.9137 ± 0.0049
	SVM	**0.9115 ± 0.0056**	0.8841 ± 0.0078	0.8907 ± 0.0049	0.9011 ± 0.0045	0.8690 ± 0.0059
	NBM	**0.9388 ± 0.0059**	0.8843 ± 0.0063	0.9089 ± 0.0052	0.9161 ± 0.0040	0.9160 ± 0.0055
TanCorp60	CNB	**0.7969 ± 0.0096**	0.6544 ± 0.0095	0.7233 ± 0.0105	0.7715 ± 0.0096	0.7538 ± 0.0090
	SVM	**0.7490 ± 0.0056**	0.7207 ± 0.0140	0.6816 ± 0.0130	0.7502 ± 0.0123	0.6907 ± 0.0134
	NBM	**0.8054 ± 0.0106**	0.7847 ± 0.0078	0.7124 ± 0.0094	0.7800 ± 0.0103	0.7612 ± 0.0068



(a) Performance comparison using CNB on SogouC

(b) Performance comparison using CNB on TanCorp60

(c) Performance comparison using SVM on SogouC

(d) Performance comparison using SVM on TanCorp60

(e) Performance comparison using NBM on SogouC

(f) Performance comparison using NBM on TanCorp60

Fig. 3. Performance comparison of five DRMs with different classification models on two datasets.

FreeG significantly outperforms the other methods in almost all tests in Precision and Recall scores, and Bigram achieves the second best result. With respect to SogouC and the classifier CNB, for example, the micro Precision of FreeG is about 6.47% higher than BoW, 3.51% higher than ICTCLAS, 2.53% higher than Bigram and 3.15% higher than Trigram.

Moreover, Fig. 3 reports the micro F1-measure of five competing DRMs in the context of different training proportion, where FreeG obviously outperforms the other 4 models most of the times and its superiority is basically independent to datasets, classification models and proportion of training data. Furthermore, we can see that, except those of FreeG, the classification performance of the rest of

Table 4
Significance test at 0.05 level and 10,000 iteration. X≫Y indicates that X is significantly better than Y.

DRMs	Datasets	
	SogouC	TanCorp60
CNB	**FreeG** ≫ Bigram ≫ ICTCLAS ≫ Trigram ≫ BoW	**FreeG** ≫ Bigram = Trigram ≫ ICTCLAS ≫ BoW
SVM	**FreeG** ≫ Bigram ≫ ICTCLAS ≫ BoW ≫ Trigram	**FreeG** ≫ Bigram = BoW ≫ Trigram = ICTCLAS
NBM	**FreeG** ≫ Bigram ≫ ICTCLAS ≫ BoW = Trigram	**FreeG** ≫ BoW ≫ Bigram ≫ Trigram ≫ ICTCLAS

Table 5
Comparison of time costs for training and testing in classification.

Algorithms	Time for Training (s)			Time for Testing (s)		
	CNB	SVM	NBM	CNB	SVM	NBM
FreeG	**0.18**	**19.39**	**0.23**	**0.71**	0.59	**0.79**
ICTCLAS	0.36	34.69	0.45	1.69	1.30	1.61
BoW	0.48	53.22	0.42	1.35	1.24	1.33
Bigram	1.02	109.35	1.36	2.38	3.09	2.69
Trigram	2.27	67.81	1.74	6.70	**0.41**	7.30

4 approaches are not necessarily consistent with their coding efficiency we obtained in Section 4.1. It again confirms that FreeG is a compromised solution between syntactical and statistical phrases, which not only efficiently captures the syntactical structure of Chinese (like by ICTCLAS) in coding but also ensure promising determinativeness of the feature vectors.

We have also conducted significant test [19] using randomization test on the above results presented in Table 4, and the results have validated the hypothesis that the performance of FreeG is superior to those of the other 4 models.

4.3. Comparison of effectiveness

In this section, we compare the effectiveness of the five models by investigating their time cost for training and testing in the classification task. As the time for the construction and document indexing can be done offline, we only count the average time for training the corresponding classifier and using the classifier for prediction. Table 5 demonstrates this results, which shows that FreeG is basically the most effective one among the 5 schemes under investigation. This is not surprise because a more representative model will create feature vectors with smaller dimension and thus saves the computational time for training and testing, another advantage of sparse coding.

5. Conclusion and future work

In this paper, we propose a free-gram phrase identification scheme for efficient codebook generation. We have demonstrated that by converting each document into a context graph and using that graph to identify possible phrases with all lengths can effectively reduce the search space and significantly increase the chance of finding more representative phrases. The experimental result have validated the efficiency of our approach in codebook generation on both balanced and unbalanced datasets. In the experiment of classification, the codebook generated with our method have exhibited higher discrimination than those by conventional methods with various classification models, which is contributed to the assumption that syntactical feature is beneficial to form sparse code, while statistical feature is helpful to improve the representativeness of specific document.

It worth mentioning that the two datasets SogouC and TanCorp60 are all in news domain, so that the phrases in their codebooks might be domain specific. Therefore, whether this will affect the efficiency and discrimination when moved to other domains, or whether there is a set of phrases that globally applicable to all domains, are questions worth further investigation. Furthermore, it is interesting to test the proposed approach in other tasks such as information retrieval for better understanding of the data representation created by our method. We will include this in our future work when queries and corresponding ground truths are available.

Acknowledgements

This work was supported by National Basic Research Program of China 973 Program under Grant No. 2011CB302201, and Doctoral Fund of Ministry of Education of China under Grant No. 20110181110049.

References

[1] M.F. Caropreso, S. Matwin, F. Sebastiani, A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization, in: Text Databases and Document Management: Theory and Practice, IGI Publishing, 2001, pp. 78–102.

[2] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 2008, pp. 160–167.

[3] L.L. Dai, An aggressive algorithm for multiple string matching, Inform. Process. Lett. 109 (11) (2009) 553–559.

[4] B. Durian, J. Holub, H. Peltola, J. Tarhio, Improving practical exact string matching, Inform. Process. Lett. 110 (4) (2010) 148–152.

[5] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: A library for large linear classification, J. Mach. Learn. Res. 9 (2008) 1871–1874.

[6] D.J. Field, What is the goal of sensory coding?, Neural Comput. 6 (4) (1994) 559–601.

[7] A. Hotho, A. Nürnberger, G. Paaß, A brief survey of text mining, LDV Forum, GLDV J. Comput. Linguist. Lang. Technol. 20 (1) (2005) 19–62.

[8] S. Jaillet, A. Laurent, M. Teisseire, Sequential patterns for text categorization, Intell. Data Anal. 10 (3) (2006) 199–214.

[9] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, in: Machine Learning: ECML-98, in: Lecture Notes in Computer Science, vol. 1398, 1998, pp. 137–142.

[10] M. Keikha, A. Khonsari, F. Oroumchian, Rich document representation and classification: An analysis, Knowledge-Based Syst. 22 (1) (2009) 67–71.

[11] T. Lecroq, Fast exact string matching algorithms, Inform. Process. Lett. 102 (6) (2007) 229–235.

[12] D.D. Lewis, An evaluation of phrasal and clustered representations on a text categorization task, in: Proceedings of the Fifteenth Annual

International Acm Sigir Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, 1992, pp. 37–50.

[13] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins, Text classification using string kernels, J. Mach. Learn. Res. 2 (2002) 419–444.

[14] A. McCallum, K. Nigam, A comparison of event models for Naive Bayes text classification, in: Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, AAAI Press, 1998, pp. 41–48.

[15] F. Mhamdi, R. Rakotomalala, M. Elloumi, A Hierarchical *n*-Grams Extraction Approach for Classification Problem, Lecture Notes in Computer Science, 2009, pp. 211–222.

[16] M.A. Ranzato, C. Poultney, S. Chopra, Y. Lecun, Efficient learning of sparse representations with an energy-based model, in: Advances in Neural Information Processing Systems, 2006.

[17] J.D.M. Rennie, L. Shih, J. Teevan, D.R. Karger, Tackling the poor assumptions of Naive Bayes text classifiers, in: Proceedings of the Twentieth International Conference on Machine Learning, Washington DC, 2003, pp. 616–623.

[18] E.T. Rolls, M.J. Tovee, Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex, J. Neurophysiology 73 (2) (1995) 713–726.

[19] J.P. Romano, On the behavior of randomization tests without a group invariance assumption, J. Am. Stat. Assoc. 85 (411) (1990) 686–692.

[20] F. Sebastiani, Machine learning in automated text categorization, Acm. Comput. Surv. 34 (1) (2002) 1–47.

[21] W.M. Soon, H.T. Ng, D.C.Y. Lim, A machine learning approach to coreference resolution of noun phrases, Comput. Linguist 27 (4) (2001) 521–544.

[22] S. Tan, An effective refinement strategy for KNN text classifier, Expert Syst. Appl. 30 (2) (2006) 290–298.

[23] S. Tan, X. Cheng, M.M. Ghanem, B. Wang, H. Xu, A novel refinement approach for text categorization, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, ACM, Bremen, Germany, 2005, pp. 469–476.

[24] D. Trenkic, Definiteness in Serbian/Croatian/Bosnian and some implications for the general structure of the nominal phrase, Lingua 114 (11) (2004) 1401–1427.

[25] F.R. Wei, W.J. Li, Q. Lu, Y.X. He, A document-sensitive graph model for multi-document summarization, Knowledge-Inf. Syst. 22 (2) (2010) 245–259.

[26] H.-P. Zhang, Q. Liu, X.-Q. Cheng, H. Zhang, H.-K. Yu, Chinese lexical analysis using hierarchical hidden Markov model, in: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Association for Computational Linguistics, Sapporo, Japan, 2003, pp. 63–70.

[27] W. Zhang, T. Yoshida, X.J. Tang, Text classification based on multi-word with support vector machine, Knowledge-Based Syst. 21 (8) (2008) 879–886.

[28] S. Zhou, J. Guan, Chinese documents classification based on N-grams, in: the Third International Conference on Computational Linguistics and Intelligent Text Processing, 2002, pp. 31–50.