Distributional Features for Text Categorization

Xiao-Bing Xue and Zhi-Hua Zhou

National Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China
{xuexb, zhouzh}@lamda.nju.edu.cn

Abstract. In previous research of text categorization, a word is usually described by features which express that whether the word appears in the document or how frequently the word appears. Although these features are useful, they have not fully expressed the information contained in the document. In this paper, the *distributional features* are used to describe a word, which express the distribution of a word in a document. In detail, the *compactness* of the appearances of the word and the *position of the first appearance* of the word are characterized as features. These features are exploited by a TFIDF style equation in this paper. Experiments show that the distributional features are useful for text categorization. In contrast to using the traditional term frequency features solely, including the distributional features requires only a little additional cost, while the categorization performance can be significantly improved.

1 Introduction

In the last ten years, content-based document management tasks have gained a prominent status in the information system field, due to the increased availability of documents in digital form and the ensuring need to access them in flexible ways [14]. Among such tasks, text categorization has attracted more and more attention due to its wide applicability. Many classifiers widely used in Machine Learning community have been applied to this task, such as Naïve Bayes, Decision Tree, Neural Network and k-Nearest Neighbor (kNN). Recently, some excellent results have been obtained by SVM [6] and AdaBoost [13].

While a wide range of classifiers have been used, virtually all of them were based on the same text representation, 'bag of words', where a document is represented as a set of words appearing in this document. Features used to describe a word are usually the ones which express whether the word appears in a document or how frequently this word appears. Are these features enough?

Considering the following example, 'Here you are' and 'You are here' are two sentences corresponding to the same vector using the above features, but their meanings are totally different. Although this is a somewhat extreme example, it clearly illustrates that besides the appearance and the frequency of appearance of a word, the distribution of a word is also important. Therefore, this paper attempts to design some *distributional features* to measure the characteristics of a word's distribution in a document.

The first consideration is the *compactness of the appearances* of a word. Here, the 'compactness' measures the extent that the appearances of a word concentrate. A word is compact if its appearances concentrate in a specific part of a document, and less compact if its appearances spread over the whole document. This consideration is motivated by the following facts. A document usually contains several parts. If the appearances of a word are less compact, the word is more likely to appear in different parts and more likely to be related to the theme of the document. For example, consider Document A (NEWID=2367) and Document B (NEWID=7154) in Reuters-21578. Document A talks about the debate on whether expanding the 0/92 programme or just limiting this programme on wheat. Obviously, this document belongs to the category 'wheat'. Document B talks about the U.S. Agriculture Department's proposal on tighter federal standards about insect infections in grain shipments and this document belongs to the category 'grain' but not to the category 'wheat'. Let's consider the importance of the word 'wheat' in both documents. Since the content of A is more closely related to wheat than B, the importance of the word 'wheat' should be higher in A. However, the normalized frequency of this word is almost the same in both documents. Therefore, the frequency is not enough to distinguish this difference of importance. Here, the compactness of the appearances of a word can provide a different view. In A, since the document mostly discusses the 0/92 programme on wheat, the word 'wheat' appears in different parts of this document. In B, since the document mainly discusses the contents of the new standard on grain shipment and just one part of the new standard refers to wheat, the word 'wheat' only appears in one paragraph of this document. So the compactness of the appearances of the word 'wheat' is lower in A than in B, which well expresses the importance of this word.

The second consideration is the *position of the first appearance* of a word. This consideration is based on an intuition that the author naturally mentions the important contents in the earlier parts of a document. Let's consider Document A (NEWID=3981) and Document B (NEWID=4679) in Reuters-21578. Document A belongs to the category 'grain' and talks about the heavy rain in Argentine grain area. Document B belongs to the category 'cotton' and discusses that China is trying to increase cotton output. Obviously, the word 'grain' should be more important in A than in B. Unfortunately, the frequency of the word 'grain' is even lower in A. Now, let's consider the position of the first appearance of the word 'grain'. In A, it firstly appears in the title. It's not strange, because this document is mostly about Argentine grain area. In B , the word 'grain' firstly appears at the end of the document. It's not strange either. Since the theme of this document is about increasing cotton output, the suggestion that the production of cotton be coordinated with other crops such as grain is indirectly related to this theme, so the author naturally mentioned this suggestion at the end of the document. Obviously, the position of the first appearance of a word could express the importance of this word to some extent.

Above all, while the frequency of a word expresses the intuition that *the more frequent, the more important*, the compactness of the appearances of a word shows that *the less compact, the more important* and the position of the first appearance of a word shows that *the earlier, the more important*. In order to test the effect of these distributional features, kNN and SVM are used. Experiments suggest that the distributional features are useful for text categorization.

This paper designs some distributional features for text categorization, which can help improve the performance while requiring only a little additional cost. The paper also explores how to use these distributional features and discusses that when these features are greatly helpful.

The rest of this paper is organized as follows. Section 2 briefly introduces some related works. Section 3 describes how to extract the distributional features. Section 4 discusses how to utilize these features. Section 5 reports on the experiments. Finally, Section 6 concludes.

2 Related Work

This section focuses on the features of a word used in previous text categorization work. A thorough review of text categorization can be found in [14].

Moschitti and Basili [9] has studied the effect of two kinds of linguistic features, POS-tag and word senses. The POS-tag describes the part of speech of each word, which includes verb, noun, pronoun, adjective and so on. A word's POS-tag is identified through Brill tagger. The word senses describe the meanings of a word. For example, consider the word 'bass', its two senses are: a type of fish; tones of low frequency. For a given word, its most appropriate sense is chosen from the possible senses in WordNet through some Word Sense Disambiguation (WSD) algorithms. In general, the improvement of performance brought by these linguistic features is not significant, especially when the cost of getting such features is considered.

Recently, Sauban and Phahringer [12] proposed a new text representation method. In their work, a discriminative score for every word is firstly calculated. Then, with every word input in sequence, a document is shown as a curve depicting the change of the accumulated scores. This curve is called 'Document Profiling'. Two different methods are used to turn a profile into a constant number of features. One is to sample from the profile with a fixed gap and the other is to get some high-level summary information from the profile. Comparable results with the 'bag of words' representation were achieved with lower computational cost. Although no new features are explicitly extracted for a word in this work, the information about the word sequence in a document is utilized.

3 How to Extract Distributional Features

From Section 2, it is noticed that the effect of the distributional features has not been explored in previous researches on text categorization. In this section, the extraction of the distributional features is discussed.

Firstly, a document is divided into several parts. Then, the distribution of a word could be modelled as an array where each element records the number of appearances of this word in the corresponding part. The length of this array is the total number of the parts.

For the above distributional model, what is a part is a basic problem. As mentioned by Callan [3], there are three types of passages used in information retrieval. Here, the meaning of ‘passage’ is the same as ‘part’ which is defined as any sequence of text from a document. *Discourse Passage* is based on logic components of documents such as sentences and paragraphs, *semantic passage* corresponds to a topic or subtopic and *window passage* is simply a sequence of words. Considering efficiency, the semantic passage is not used. Compared with window passage, discourse passage is more accurate. Furthermore, sentence is more consistent in length than paragraph. Thus, a sentence is used as a part in this work. For example, for a document d with 10 sentences, the distribution of the word ‘corn’ is depicted as Fig. 1, then the distributional array for ‘corn’ is [2,1,0,0,1,0,0,3,0,1].

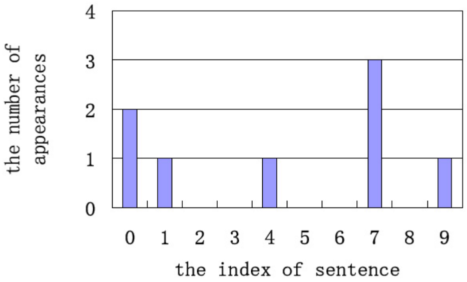

Fig. 1. The distribution of ‘corn’.

In order to measure the compactness of the appearances of a word, the mean distance between all appearances of this word and the centroid of this word is used. The centroid of a word records the mean position of all appearances of this word. If a word appears in a given sentence, then the position of this appearance is the index of the sentence. The position of the first appearance of a word can be calculated similarly.

Suppose in a document d containing n sentences, the distributional array of the word t is $array(t, d) = [c_0, c_1, ..., c_{n-1}]$. Then, the compactness ($ComPact$) of the appearances of the word t and the position of the first appearance ($FirstApp$) of the word t are defined, respectively, as follows:

$$count(t, d) = \sum_{i=0}^{n-1} c_i \qquad centroid(t, d) = \frac{\sum_{i=0}^{n-1} c_i \times i}{count(t, d)}$$

$$ComPact(t, d) = \frac{\sum_{i=0}^{n-1} c_i \times |i - centroid(t, d)|}{count(t, d)} \tag{1}$$

$$FirstApp(t, d) = \min_{i \in \{0..n-1\}} c_i > 0?i : n \tag{2}$$

In Eq. 2, $exp = a?b : c$ means if the condition a is satisfied, then the value of expression exp is b, otherwise the value is c.

The example in Fig. 1 is used again to illustrate how to calculate the distributional features.

$$\begin{aligned}
count(\text{‘corn’}, d) &= 2 + 1 + 1 + 3 + 1 = 8 \\
centroid(\text{‘corn’}, d) &= (2 \times 0 + 1 \times 1 + 1 \times 4 + 3 \times 7 + 1 \times 9)/8 = 4.375 \\
ComPact(\text{‘corn’}, d) &= (2 \times 4.375 + 1 \times 3.375 + 1 \times 0.375 + 3 \times 2.625 \\
&\quad +1 \times 4.625)/8 = 3.125 \\
FirstApp(\text{‘corn’}, d) &= min\{0, 1, 10, 10, 4, 10, 10, 7, 10, 9\} = 0
\end{aligned}$$

Then, let’s compare the cost of extracting the distributional features and that of extracting only term frequency. Suppose the size of the longest document in corpus is l, the size of the vocabulary is m, the biggest number of sentences a document contains is n and the number of documents in corpus is s. A memory block with size l is required for loading a document and an $m \times 1$ array is required for recording the number of appearances of each word in the vocabulary. When the scan of a document is completed, the term frequency can be directly obtained from the above array. In order to extract the distributional features, an additional $m \times n$ array is needed, since for each word, an $n \times 1$ array is used to record the distribution of this word. When the scan of a document is completed, Eq. 1 and Eq. 2 are used to calculate the distributional features. No other additional cost is needed compared with extracting the term frequency. Overall, the additional computational cost for extracting the distributional features is $s \times m \times$ (Cost of Eq. 1+Cost of Eq. 2) and the additional storage cost is $m \times n$. It is worth noting that the above additional computational cost is the worst case, since practically the calculation is only required for words that appear at least once in a document. Actually, the number of such words of a document is significantly smaller than m. Thus, the additional computational and storage cost for extracting the distributional features is not big. The process of extracting term frequency and the distributional features is illustrated in Fig. 2.

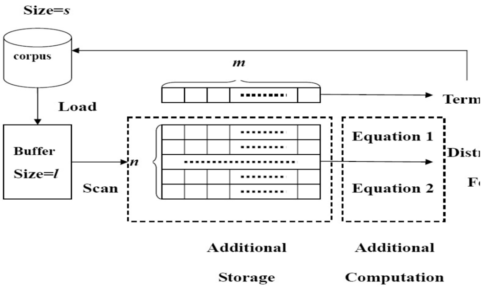

Fig. 2. The process of extracting term frequency and distributional features

4 How to Utilize Distributional Features

Term Frequency in TFIDF could be regarded as a value that measures the importance of a word in a document. As discussed in Section 1, the importance of a word not only can be measured by its term frequency, but also can be measured by the compactness of its appearances and the position of its first appearance. Therefore, the standard TFIDF equation can be generalized as follows:

$$tfidf(t,d) = Importance(t,d) \times IDF(t) \tag{3}$$

When different features are involved, $Importance(t,d)$ corresponds to different values. When the feature is the frequency of a word, TermFrequency (TF) is used. When the feature is the compactness of the appearances of a word, ComPactness (CP) is used. When the feature is the position of the first appearance of a word, FirstAppearance (FA) is used. TF, CP and FA are calculated as follows:

$$TF(t,d) = \frac{count(t,d)}{size(d)} \tag{4}$$

$$CP(t,d) = \frac{ComPact(t,d)+1}{len(d)} \tag{5}$$

$$FA(t,d) = 1 - \frac{FirstApp(t,d)}{len(d)} \tag{6}$$

$size(d)$ in Eq. 4 is the total number of words of Document d. $len(d)$ in Eqs. 5 and 6 is the total number of sentences of Document d. In Eq. 5, $ComPact(t,d)$ is added by 1 in order to ensure $CP(t,d) > 0$. In Eq. 6, the position of the first appearance of word t is subtracted from 1 to reflect the intuition that the earlier a word appears, the more important this word is. Actually, FA value assumes different importance for different positions of a document. Fig. 3 shows the FA value of a word when it firstly appears in different sentences in a document containing 10 sentences. In this paper, this importance is simply assumed to decrease linearly with the index of sentence. Notice that $len(d)$ in Eq. 6 determines the speed of the decreasing of the FA value. The smaller the value of $len(d)$, the faster the speed of decreasing. For short documents, this property contradicts the intuition that the importance of words in short documents differs slightly with the change of position. So a heuristic rule is used here. When the number of sentences in a document is less than 10, $len(d)$ is set to 10, otherwise $len(d)$ is set to the actual number of sentences. Finally, if a word t doesn't appear in Document d, $Importance(t,d)$ is set to 0, no matter which feature is used.

Since TF, CP and FA measure the importance of a word from different views, the combination of them may improve the performance. The strategy used here is to exploit the ensemble learning technique [5]. A group of classifiers are trained based on different features. The label of a new document is decided by the combination of the outputs of these classifiers. Note that the outputs of each classifier are the confidence scores which approximately indicate the probabilities that this new document belongs to each category.

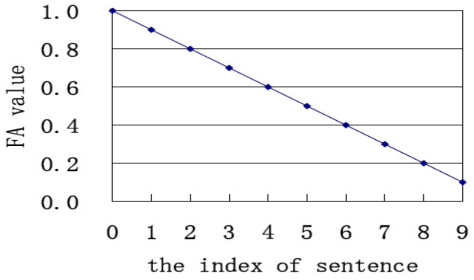

Fig. 3. FA value when a word firstly appears in different sentences.

5 Experiments

In this section, the effect of the distributional features is explored for kNN and SVM on three datasets: Reuters-21578, 20 Newsgroup and WebKB. On these three datasets, a lot of work has been published [1][2][6][11][15].

5.1 Datasets

The Reuters-21578 corpus [8] contains 21578 articles taken from Reuters newswire. The 'ModeApte' split is used. Following Yang and Liu [15], 90 categories which have at least one document in both training set and test set are extracted.

The 20 Newsgroup corpus [7] contains 19997 articles taken from the Usenet newsgroup collections. Following Schapire and Singer [13], the duplicate documents are removed and the documents with multiple labels are detected both using the 'Xrefs' header. There are 19465 documents left. Four-fold cross validation is executed.

The WebKB corpus [4] is a collection of 8282 web pages obtained from four academic domains. Following Nigam [10], four categories: *course*, *faculty*, *project*, *student* are used and this part of corpus contains 4199 documents. Four-fold cross validation is executed. Since this corpus consists of web pages, it is difficult to accurately extract sentence from each document as on Reuters-21578 and 20 Newsgroup. Therefore, as mentioned in Section 3, window passage is used on this corpus. Empirically, 20 words are used as the window size.

5.2 Performance Measure

Reuters-21578 and 20 Newsgroup are multi-label datasets. For evaluating the performance on these two corpus, the standard precision, recall and F1 measure is used. Given the contingency table of category C_i (Table 1), the precision(p_i), recall(r_i) and F1 measure($F1_i$) of category C_i is calculated as follows.

$$p_i = \frac{TP_i}{TP_i + FP_i} \qquad r_i = \frac{TP_i}{TP_i + FN_i} \qquad F1_i = \frac{2 \times p_i \times r_i}{(p_i + r_i)}$$

Table 1. The contingency table for category C_i

Category		Expert Judgement	
C_i		Yes	No
Classifier	Yes	TP_i	FP_i
Judgement	No	FN_i	TN_i

These measures can be aggregated over all categories in two ways. One is to average each category's precision, recall and F1 to get the global precision, recall and F1. This method is called *macro-averaging*. The other is based on the global contingency table (Table 2), which is called *micro-averaging*. Macro-averaging is more affected by the classifier's performance on rare categories while micro-averaging is more affected by performance on common categories. In this paper, micro-F1 and macro-F1 are both reported. WebKB is a uni-label dataset, and therefore *accuracy* is used for evaluating performance on this dataset.

Table 2. The global contingency table

Category set		Expert Judgement	
$C = C_1, C_2, ..., C_{\|C\|}$		Yes	No
Classifier	Yes	$TP = \sum_{i=1}^{\|C\|} TP_i$	$FP = \sum_{i=1}^{\|C\|} FP_i$
Judgement	No	$FN = \sum_{i=1}^{\|C\|} FN_i$	$TN = \sum_{i=1}^{\|C\|} TN_i$

In order to explore the effect of the distributional features, the traditional TFIDF is used as baseline. First, CP and FA are respectively used as the importance measure. Second, three combinations of any two features are tested. Finally, the result of the combination of all three features is reported.

5.3 Results

Table 3 shows results of kNN and SVM on three datasets where the best performance is boldfaced. Here, *All* means the combination of three features, i.e. TF+FA+CP. The first question is: are the distributional features useful for text categorization?

On Reuters-21578, distributional features behaves well for micro-F1. The results are similar for SVM and kNN. FA is slightly inferior to TF while CP is slightly better than TF. When different combinations are tried, the combined results are always better than each component. At last, the best result is achieved by TF+FA+CP. For macro-F1, distributional features failed to show any improvement for kNN while CP significantly improves the baseline for SVM.

On 20 Newsgroup, distributional features significantly improve the baseline result for micro-F1. For kNN, FA and CP significantly improve the baseline result. For different combinations, no combination significantly further improves the result of FA except the combination of FA and CP. The best result is achieved

Table 3. Results of *k*NN and SVM on three datasets

	*k*NN					SVM				
	Reu		New		Web	Reu		New		Web
	miF1	maF1	miF1	maF1	acc	miF1	maF1	miF1	maF1	acc
TF	0.844	**0.495**	0.815	0.816	0.808	0.857	0.509	0.887	0.886	0.916
FA	-0.7%	-12.4%	6.7%	6.6%	**5.2**%	-0.4	-2.1%	1.5%	1.6%	**3.1**%
CP	0.5%	-1.9%	3.9%	3.8%	0.9%	0.5%	**2.7**%	0.0%	0.0%	1.4%
TF+FA	1.2%	-4.9%	5.7%	5.6%	3.9%	1.1%	0.0%	**2.0**%	**2.0**%	2.9%
TF+CP	1.0%	-1.5%	3.2%	3.1%	1.4%	0.9%	2.1%	0.7%	0.7%	1.5%
FA+CP	0.7%	-7.6%	**6.9**%	**6.8**%	3.9%	0.8%	1.5%	1.7%	1.7%	3.0%
All	**1.8**%	-6.2%	6.0%	5.9%	3.5%	**1.3**%	1.9%	1.7%	1.7%	2.8%

Table 4. Statistical significance test of *k*NN and SVM

	*k*NN										SVM									
	Reu				New				Web		Reu				New				Web	
vs. TF	s	S	T	T'	s	S	T	T'	s	p	s	S	T	T'	s	S	T	T'	s	p
FA	<	≪	≪	∼	≫	≫	≫	≫	≫	≫	∼	∼	∼	>	≫	≫	≫	≫	≫	≫
CP	∼	∼	∼	∼	≫	≫	≫	≫	∼	∼	∼	>	∼	≫	∼	∼	∼	∼	≫	>
TF+FA	≫	∼	∼	∼	≫	≫	≫	≫	≫	≫	≫	>	∼	≫	≫	≫	≫	≫	≫	≫
TF+CP	≫	∼	∼	≫	≫	≫	≫	≫	≫	∼	≫	>	∼	≫	≫	≫	≫	≫	≫	≫
FA+CP	∼	<	∼	∼	≫	≫	≫	≫	≫	≫	>	>	∼	≫	≫	≫	≫	≫	≫	≫
All	≫	∼	∼	∼	≫	≫	≫	≫	≫	≫	≫	>	∼	≫	≫	≫	≫	≫	≫	≫

by FA+CP. For SVM, FA performs better than TF while CP shows no improvement for TF. When different combinations are tried, the combined results are better than each component. The best result is achieved by TF+FA. The result of macro-F1 is almost the same as micro-F1, since each category is almost equally distributed on this corpus.

On WebKB, distributional features also significantly improve the baseline. The results of SVM and *k*NN are similar. FA significantly improves the basline on this corpus, and CP slightly improves the baseline. When different combinations are tried, no combination can further improve the result of FA. The best result is achieved by FA.

Statistical significance tests including s-test, p-test, S-test, T-test and T'-test are conducted on the results reported in Table 3. The s-test and p-test were designed to evaluate the performance at a micro level and the S-test, T-test and T'-test were designed to evaluate the performance at a macro level. Further information on these tests can be found in [15]. Note that for each corpus the types of significance tests conducted are determined by the performance measure used on this corpus. The results are summarized in Table 4, where 'A≫B' implies that the performance with A is significantly better than B at 0.01 significance level, 'A>B' implies that the performance with A is significantly better than B at 0.05 significance level, and 'A∼B' implies that the performances of A and B are comparable at 0.05 significance level. In general, it is clear that the distributional features are helpful in text categorization.

Table 5. Results of *k*NN on Short and Long datasets

	ReuS		NewS		WebS	ReuL		NewL		WebL
	miF1	maF1	miF1	maF1	acc	miF1	maF1	miF1	maF1	acc
TF	0.880	0.467	0.767	0.774	0.817	0.680	0.291	0.817	0.805	0.752
FA	-0.8%	-8.9%	**7.7%**	**7.0%**	**2.9%**	-1.5%	-9.7%	**7.0%**	**7.2%**	**12.2%**
CP	0.2%	-2.1%	5.4%	4.9%	0.2%	-0.9%	-1.6%	2.6%	2.6%	4.7%
TF+FA	0.6%	**2.5%**	6.1%	5.8%	2.1%	0.3%	-0.8%	4.8%	4.9%	7.6%
TF+CP	0.5%	0.0%	3.7%	3.4%	0.5%	1.0%	3.9%	2.5%	2.8%	4.2%
FA+CP	0.0%	-5.5%	7.6%	**7.0%**	1.4%	**2.2%**	**4.3%**	6.4%	6.4%	10.5%
All	**1.1%**	2.4%	6.6%	6.2%	1.7%	-0.1%	3.2%	5.3%	5.5%	8.3%

Table 6. Results of SVM on Short and Long datasets

	ReuS		NewS		WebS	ReuL		NewL		WebL
	miF1	maF1	miF1	maF1	acc	miF1	maF1	miF1	maF1	acc
TF	0.895	0.498	0.845	0.850	0.916	0.673	0.333	0.904	0.896	0.890
FA	-0.7%	-0.7%	**2.4%**	**2.5%**	**2.8%**	1.2%	-6.8%	1.8%	2.0%	**4.1%**
CP	-0.1%	**1.4%**	1.0%	1.1%	0.9%	3.4%	-0.2%	-0.5%	-0.4%	-1.1%
TF+FA	-0.1%	-0.7%	**2.4%**	2.4%	2.3%	3.1%	-1.2%	**2.0%**	**2.1%**	4.0%
TF+CP	**0.1%**	0.3%	1.0%	1.0%	1.2%	4.2%	**6.0%**	0.7%	0.8%	0.7%
FA+CP	-0.3%	0.2%	**2.4%**	2.4%	2.6%	**4.8%**	-0.8%	1.9%	2.0%	3.8%
All	-0.1%	1.0%	2.0%	2.0%	2.2%	3.3%	-2.2%	1.9%	**2.1%**	3.5%

Furthermore, note that when the distributional features are introduced, there is slight improvement on Reuters-21578 but significant improvement on 20 Newsgroup and WebKB. Therefore, the second question arises: when are the distributional features greatly useful?

As mentioned before, when the compactness of the appearances of a word is introduced, it is assumed that a document contains several parts and the word only appears in a part is not closely related to the theme of the document. When the position of the first appearance of a word is introduced, it is assumed that the word mentioned late by the author is not closely related to the theme of the document. Intuitively, these two assumptions are more likely to be satisfied when a document is long enough. This intuition is based on human's habit of writing. When the length of a document is limited, the author will concentrate on the most related content, such as when writing the abstract section in an academic paper. When there is no limit for the length, the author may write some indirectly related content, such as when writing the body of a paper. In order to verify this intuition, the mean length of documents from these three experimental corpora is reported. Here, the length of a document is measured by its number of sentences. The average length of documents is 6.99, 13.66 and 14.11 respectively for Reuters-21578, WebKB and 20 Newsgroup. It seems that the improvement brought by the distributional features is closely related to the mean length of documents. In order to further verify this idea, each of the three corpora is split into two new corpora, i.e. the Short corpus and the Long corpus, according to the length of documents. For each corpus, the Short corpus contains documents with length no more than 10 and the Long corpus contains documents with length more than 10. Experiments are repeated for these six new generated datasets. The results of kNN on Short and Long datasets are reported in Table 5. The results of SVM on Short and Long datasets are reported in Table 6.

According to Tables 5 and 6, the distributional features brought more significant improvement on the Long dataset than on the Short dataset of Reuters-21578 and WebKB. Comparable improvements are achieved on the Short and Long datasets of 20 Newsgroup. In general, the effect of the distributional features is more obvious on the Long datasets than on the Short ones.

However, the differences of the improvement brought by the distributional features still exist among three corpora in Tables 5 and 6. The improvement is more significant on 20 Newsgroup and WebKB than on Reuters-21578 in most situation. It seems that there are other factors that also contribute to the performance of the distributional features. Note that the sources of three corpora are different, the documents in Reuters-21578 are taken from news reports, the documents in 20 Newsgroup are taken from newsgroup documents and the documents in WebKB are taken from web pages. For news reports, they are written by professional journalists and editors and the writing style is formal and precise, therefore the loosely related content is less likely to appear in this type of articles. In contrast, for newsgroup documents and web pages, they are written by ordinary web users and the writing style is very causal, therefore the loosely related content is more likely to appear in this type of articles. Thus, it seems that the effect of the distributional features is more obvious for informal documents than for formal ones.

Therefore, the answer to the second question, i.e. when the distributional features are greatly useful, is: when the documents are long enough and when the documents are informal.

6 Conclusion

Previous researches on text categorization usually use the features of appearance or the frequency of appearance to characterize a word. These features are not enough for fully capturing the information contained in a document. In this paper, the distributional features of a word are explored. These features encode a word's distribution from some aspects. In detail, the compactness of the appearances of a word and the position of the first appearance of a word are used. A TFIDF style equation is constructed to utilize these distributional features. Experiments show that the distributional features are useful for text categorization, especially when they are combined with term frequency. Further analysis reveals that the effect of the distributional features is obvious when the documents are long enough and when the documents are informal.

It is noticed that the task on WebKB is somewhat genre-based while the tasks on Reuters-21578 and 20 Newsgroup are topic-based. Intuitively, it is convincing that the distributional features may bring more benefits on genre-based corpus than on topic-based corpus. This supposition will be explored in the future.

In this paper, Eqs.1 and 2 are mainly designed for validating the usefulness of distributional features. More careful designs are anticipated to improve the performance, e.g. an alternative to Eq. 1 may work well on more peaks, which is an interesting work in the future. How to design an alternative feature to IDF in Eq. 3 specifically to work with the proposed distributional features is another interesting future issue.

Acknowledgment

Supported by the NSFC (60505013, 60325207) and JiangsuSF (BK2005412).

References

1. L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In Proceedings of SIGIR-98, Melbourne, Australia, 1998. 96-103.
2. R. Bekkerman, R. El-Yaniv, N. Tishby and Y. Winter. Distributional word clusters vs. words for text categorization. Journal of Machine Learning Research, 2003, **3** : 1182-1208.
3. J. P. Callan. Passage retrieval evidence in document retrieval. In Proceedings of SIGIR-94, Dublin, Ireland, 1994. 302-310.
4. M. Craven, D. DiPasquo, D. Freitag, A. K. McCallum, T. M. Mitchell, K. Nigam and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In Proceedings of AAAI-98, Madison, WI, 1998. 509-516.
5. T. G. Dietterich. Machine learning research: Four current directions. AI Magazine, 1997, **18** (4): 97-136.
6. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of ECML-98, Chemnitz, Germany, 1998. 137-142.
7. K. Lang. Newsweeder: Learning to filter netnews. In Proceedings of ICML-95, Tahoe City, CA, 1995. 331-339.
8. D. Lewis. Reuters-21578 text categorization test colleciton, Distrib. 1.0, September 26, 1997.
9. A. Moschitti and R. Basili. Complex linguistic features for text classification: A comprehensive study. In Proceedings of ECIR-04, Sunderland, UK, 2004. 181-196.
10. K. Nigam, A. K. McCallum, S. Thrun and T. M. Mitchell. Learning to classify text from labeled and unlabeled documents. In Proceedings of AAAI-98, Madison, WI, 1998. 792-799.
11. J. Rennie, L. Shih, J. Teevan and D. Karger. Tackling the poor assumptions of Naive Bayes text classifiers. In Proceedings of ICML-03, Washington, DC, 2003. 616-623.
12. M. Sauban and B. Pfahringer. Text categorization using document profiling. In Proceedings of PKDD-03, Cavtat-Dubrovnik, Croatia, 2003. 411-422.
13. R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. Machine Learning, 2000, **39** (2-3): 135-168.
14. F. Sebastiani. Machine learning in automated text categorization. ACM Computing Surverys, 2002, **34** (1): 1-47.
15. Y. Yang and X. Liu. A re-examination of text categorization methods. In Proceedings of SIGIR-99, Berkeley, CA, 1999. 42-49.