Akiko Aizawa

Linguistic Techniques to Improve the Performance of Automatic Text Categorization

Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium (NLPRS2001)
pages 307–314
2001

Linguistic Techniques to Improve the Performance of Automatic Text Categorization

Akiko Aizawa
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, JAPAN
akiko@nii.ac.jp

Abstract

This paper presents a method for incorporating natural language processing into existing text categorization procedures. Three aspects are considered in the investigation: (i) a method for weighting terms based on the concept of a probability weighted amount of information, (ii) estimation of term occurrence probabilities using a probabilistic language model, and (iii) automatic extraction of terms based on POS tags automatically generated by a morphological analyzer. The effects of these considerations are examined in the experiments using Reuters-21578 and NTCIR-J1 standard test collections.

1 Introduction

The success of machine learning algorithms in automatic text categorization has recently become of interest to researchers in both machine learning and information retrieval fields (Lewis and Singer, 2000; Nagata and Taira, 2001). Many methods have been applied, including Support Vector Machines (Joachims, 1998), RIPPER and sleeping experts (Cohen and Singer, 1999), and stochastic decision lists (Li and Yamanishi, 1999). There are also comparative studies that compare the performance of different categorization strategies (Yang and Liu, 1999).

Generally speaking, machine learning algorithms work with mathematically well-defined feature spaces. Their objective is to find the best discriminators, sometimes in non-linear form, to separate points originating from different classes. In order to initiate a good feature space to work on, the algorithms require careful pre-processing of raw data specific to each application domain. In the case of text categorization, the features usually correspond to terms, and the pre-processing includes: (i) automatically extracting terms from text, (ii) assigning appropriate weights to terms in each category or document, and (iii) reducing the dimensions of the feature space by selecting significant terms.

Despite advances in linguistic processing techniques, most existing studies only apply simple procedures in their pre-processing. For example, (i) term extraction by standard stemming and stop words removal procedures, (ii) term weighting by conventional schemes such as *tf-idf*, and (iii) term selection by cutting out all the low frequency terms. This motivated us to examine whether the performance of existing text categorization methods can be improved using linguistic techniques such as morphological analysis or probabilistic language modeling. We expect that such improvements will become more crucial as the size of training data increases; when the size becomes very large, useful clues may be embedded in millions of feature terms and thus be hard to find if not appropriately prepared.

In section 2, we formulate a text categorization problem and examine the statistical nature of terms using actual corpus statistics. In section 3, a term weighting scheme, which we call the *probability weighted information* (PWI) is introduced. Also described in section 3 are a discounting tech-

nique to compensate the observed term occurrence probabilities, and a simple rule-based approach to extract compound words. In section 4, some illustrative experimental results are shown where two distinctive categorization methods, the vector space oriented and machine learning-based method, are applied to Reuters-21578 and NTCIR-J1 test collections. Section 5 is the conclusion.

2 Text Categorization Problems

2.1 Description of the Problems

Our text categorization problem is formalized as follows: Let $C = \{c_1, \cdots, c_k\}$ be a specified set of k categories. Assume that a set of training documents with known categories is given. The objective of the categorization task is to identify a category or categories of some given unknown documents.

There exist two different types of categorization tasks, depending on whether a document belongs to (a) multiple categories or (b) a uniquely determined category. The former case is usually formulated as the k number of 2-class classification problems, where the two classes represent c_j ($\in C$) and $\overline{c_j}$ ($= \{c_i | c_i \in C, i \neq j\}$). On the other hand, the later case can be directly formulated as a single k-class classification problem, although the previous 2-class formulation is also applicable.

The two test collections we use in our experiments correspond respectively to these cases: Reuters-21578 with Apte split [1] to (a), and NTCIR-J1 [2] to (b). Reuters-21578 is composed of newswire articles with manually assigned topic categories. With Apte split, totals of 9,603 training documents and 3,299 test documents are provided with 90 categories being effective. The average number of categories assigned to each document is 1.2. NTCIR-J1 is composed of abstracts of academic conference papers presented at national conferences organized by Japanese academic societies. By considering the 24 largest academic societies, we have extracted 309,999 training documents and 10,000 different test documents. The number of categories assigned to each document equals 1. With both test collections, the size distribution of the categories is considerably skewed, as is commonly observed in any text categorization problem.

[1] http://www.research.att.com/~lewis/reuters21578.html

[2] http://research.nii.ac.jp/ntcadm/

2.2 Statistical Analysis of Terms in the Text

In text categorization problems, documents are usually represented as weighted vectors of terms. Probabilistic approaches assume that the documents are sets of independent samples from some unknown distributions of terms, and try to estimate the probabilities of the originating distributions.

It is well known that terms in text roughly follow the Zipfian distribution. But what is the implication of Zipf's law when these terms are used for text categorization? Figure 1 shows some statistics collected from NTCIR-J1. Figure 1-(a) is the relation between n, the frequency of a term, and $N(n)$, the number of distinct terms with their frequencies being equal to n. The plot becomes almost linear, indicating Zipf's law does hold with this corpus. Figure 1-(b) is the relation between n and $n \cdot N(n)/T$, the probability that terms with n is observed in text where T ($= \sum nN(n)$) is the total frequency of terms. Intuitively, a large number of low frequency terms and a small number of very high frequency terms are frequently observed in text.

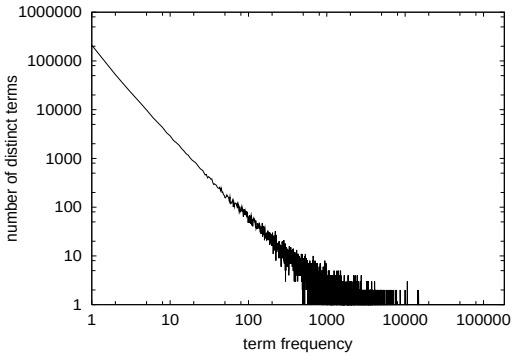
Figure 1-(c) shows the relation between n and the amount of 'information' of terms with n calculated as follows: Let W and C be random variables corresponding to terms and categories respectively, and $P(C|w_i)$ be a conditional probability that the given document contains term w_i. Then, the mutual information between W and C are given as:

$$\mathcal{I}(W, C) = \sum_{w_i} \sum_{c_j} P(w_i, c_j) log \frac{P(w_i, c_j)}{P(w_i)P(c_j)}. \quad (1)$$

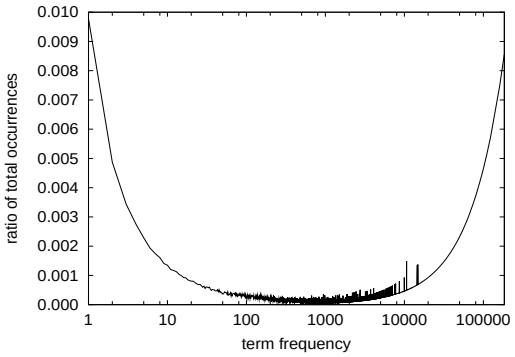
Considering the contribution of w_i in the above calculation, we obtain:

$$\delta\mathcal{I}(w_i, C) = \sum_{c_j} P(w_i, c_j) log \frac{P(w_i, c_j)}{P(w_i)P(c_j)}. \quad (2)$$

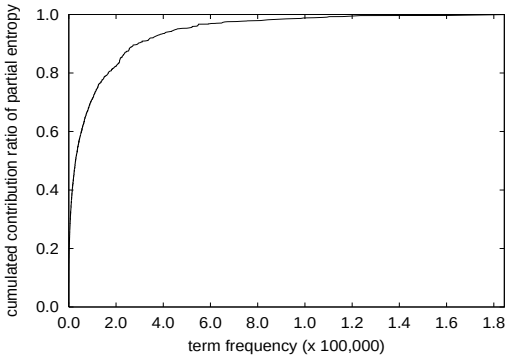
In Figure 1-(c), the value is accumulated for all the terms with the same frequencies, and then normalized (i.e., $\sum_{f(w_i)=n} \delta\mathcal{I}(w_i, C)/\mathcal{I}(W, C)$ where $f(w_i)$ is the frequency of w_i). It can be seen that the contribution is greater for low frequency terms.



(a) the relation between n and $N(n)$



(b) the relation between n and $n \cdot N(n)/T$



(c) the relation between n and $\sum_{f(w_i)=n} \delta\mathcal{I}(w_i, C)$

Figure 1: Statistical analysis of terms in the text using NTCIR-J1.

2.3 Our Approach

These figures suggest that even though the contribution of each low frequency terms is small, the accumulated effect may not be neglected since there exist an exponential number of such terms. Thus a possible strategy for text categorization is to use as many terms as possible, including all the low frequency terms.

However, such a strategy often leads to a huge feature space that is beyond the capacity of most existing machine learning methods. Consequently, the existing methods often discard low frequency terms automatically, and further apply feature subset selection. Note here that the selection strategy, as well as the number of features to be selected, is basically chosen empirically.

With this background, we address the issue of how to improve the performance of scalable methods that can manipulate all the terms contained in the target corpus. Instead of applying heuristic selection strategies, we examine the possibility of using more elaborate language processing for weighting and extracting terms.

3 The Proposed Method

3.1 Weighting Terms

We first introduce an extended notion of *tf-idf*, a commonly used term weighting scheme in information retrieval. Denoting the frequency of term w_i as $f(w_i)$, the number of documents with w_i as $D(w_i)$, and the total number of documents as D, the classical definition of *tf-idf* is given as $f(w_i)log(D/D(w_i))$, i.e., a product of term frequency (*tf*) and the inverse of log-scaled document frequency (*idf*).

Here, we can consider the *tf* factor, if normalized by the total frequency, as the estimation of the occurrence probability of a term, and the *idf* factor as the amount of information relating to the occurrence of the term (Aizawa, 2000). Then, from an information-theoretic perspective, *tf-idf* can be interpreted as the quantity required for the calculation of the expected mutual information given by Eq.

(1). Based on this interpretation, we define the extended notion of *tf-idf* as follows:

$$\begin{aligned}\delta\mathcal{I}(w_i, c_j) &= P(w_i, c_j) log \frac{P(w_i, c_j)}{P(w_i)P(c_j)} \\ &= P(w_i)P(c_j|w_i) log \frac{P(c_j|w_i)}{P(c_j)}. \quad (3)\end{aligned}$$

Since the above definition is similar to the *weighted mutual information* by Fung and McKeown (1996), we refer to such a quantity as the *probability weighted information* (PWI) in this paper.

Now, we define a criterion for text categorization using the PWI. Let $C = \{c_1, \cdots, c_k\}$ be a specified set of categories, $w_{t_1}, \cdots, w_{t_n}$ be a sequence of n terms extracted from a document to be categorized. The strategy for text categorization is to identify $c_j \in C$ that maximizes the PWI value given $w_{t_1}, \cdots, w_{t_n}$:

$$\begin{aligned}&\underset{j}{argmax} \sum_{w_i = w_{t_1} \cdots w_{t_n}} \delta I(w_i, c_j)|_{P(w_i)=1} \\ = \; &\underset{j}{argmax} \sum_{w_i = w_{t_1} \cdots w_{t_n}} P(c_j|w_i) log \frac{P(c_j|w_i)}{P(c_j)} \quad (4)\end{aligned}$$

Since Eq. (4) can be expressed as a summation of a normalized product of within category frequency and the frequency in a target document of w_i, the formula naturally provides a weighting scheme for vector-space oriented representations.

Using NTCIR-J1, we have calculated the *tf-idf* and the PWI values of each term. The correlation coefficient equals 1.00 when considering individual documents as categories, and 0.58 when considering academic societies as categories. These figures indicate that *tf-idf* provides a good estimation of the PWI with document vectors but not so much with category vectors due to the skewed distribution of category sizes.

3.2 Estimating Probabilities

The simplest way to estimate the probabilities used in Eq. (4) is to assign values proportional to the observed frequency in the training data. Denoting the frequency of w_i within category c_j as $f(w_i, c_j)$, and the total frequency of all the terms as F, the allocation policy is expressed as:

$$P(w_i) = \frac{f(w_i)}{F}, \quad P(c_j|w_i) = \frac{f(w_i, c_j)}{f(w_i)}. \quad (5)$$

However, Eq. (5) generally overestimates the probability of low frequency terms while assigning zero probability for unobserved terms. We therefore use the following absolute discounting in probabilistic language modeling studies (Kita, 1999):

$$P(w_i) = \frac{f(w_i) - \delta}{F}, \quad (6)$$

where δ is a discounting coefficient common for all the terms. The value of δ is determined either as (i)$\delta = \frac{N(1)}{|W|}$ or (ii) $\delta = \frac{N(1)}{N(1)+2N(2)}$ with $N(i)$ being the number of terms that appear exactly i times in the training corpus, and $|W|$ being the total number of distinct terms. Using the equation, the probability of unobserved term is calculated as $\frac{\delta|W|}{F}$. The estimated and observed probabilities of unobserved terms in the actual corpora are: 0.016 (estimated) and 0.029 (observed) for Reuters-21578, and 0.012 (estimated) and 0.014 (observed) for NTCIR-J1. Therefore, we can conclude that the model agrees reasonably well with our target corpora.

The estimation of $P(w_i)$ given by Eq. (6) cannot be directly applied to Eq. (4). Instead, we assume the following mixture distribution for the estimation:

$$P^*(c_j|w_i) = r(w_i)\frac{f(w_i, c_j)}{f(w_i)} + (1 - r(w_i))P(c_j). \quad (7)$$

Denoting the total frequency of terms in category c_j as $f(c_j)$, $P(c_j)$ is given by $P(c_j) = f(c_j)/F$. The mixture ratio is determined using the discounting coefficient, δ, as:

$$r(w_i) = \frac{f(w_i) - \delta}{f(w_i)}. \quad (8)$$

Measuring the distance between two probability distributions by Kullback-Leibler divergence ($\mathcal{D}$), we obtain $\mathcal{D}(P^*(C|w_i)||P(C)) \leq \mathcal{D}(P(C|w_i)||P(C))$ from the convex property of divergence. More generally, $P^*(C|w_i)$ becomes closer to $P(C)$ for smaller $r(w_i)$, and

therefore the discounting effect works more effectively for low frequency terms. Note that the expected contribution of the second term of Eq. (7) equals $\frac{\delta|W|}{F}$, the probability originally assigned to unobserved terms.

3.3 Extracting Compound Words

Existing text categorization approaches, including ours, are mostly based on the so-called 'bag-of-words' assumption that views a document as a collection of independent words. However, we can expect that compound words such as <text categorization> serve as better features than separately considered unit words such as <text> or <categorization>. Some machine learning algorithms such as RIPPER (Cohen and Singer, 1999) or a stochastic decision tree (Li and Yamanishi, 2000) are capable of extracting dependencies between these words. But general speaking, such an analysis requires much computation time. In addition, from a lexical point of view the extracted associations may be neither exhaustive nor of good quality.

As an alternative, we use standard morphological analyzers at the pre-processing stage and extract compound words using simple matching rules, defined as patterns of POS tags. In the matching, not only the longest but also all the sub-sequences of words that match the patterns are extracted. Afterwards, stemming and stop word removal procedures are applied in the case of English. For example, from "a supplementary budget" in the original text, <supplementari budget>, <supplementari>, and <budget> are obtained. After the pre-processing, all the extracted terms are considered to be independent of each other.

The morphological analyzer currently used in our implementation is Brill Tagger for English (Brill, 1994), and ChaSen for Japanese (Matsumoto et al., 1999). At present, the matching patterns are given heuristically.

4 Experimental Results

4.1 Reuters-21578

Since a single document is possibly associated with multiple categories, with Reuters-21578 the categorization task is formulated as the 90 numbers of 2-class problems, each corresponding to the 90 categories of the corpus. The overall performance is measured by the "micro-average precision-recall break-even point," following the convention of past studies using Reuters-21578.

The objective of the experiments is to investigate the effect of the following factors on the categorization performance.

(1) the effect of categorization methods

Two different types of categorization method are used in the experiments: The first, referred to as *PWI*, considers a category as a single distribution of terms and directly calculates the PWI value of each category using Eq. (4). The calculated PWI values are used to rank the categories for each tested document. This method does not require complex parameter tuning in its calculation and is less time consuming.

The second method, referred to as *SVM*, considers documents as independent points on the feature space, and calculates the optimal decision boundary of the positive and negative points using Support Vector Machines. *Tf-idf* weighting is used to calculate document vectors. We use Support Vector Machines, which work especially well with high-dimensional feature spaces and also performed best in the past studies using Reuters-21578. SVM^{light} V.3.50 [3] with linear kernel option is used in the experiments.

(2) the effect of probability estimations

Two probability estimation methods are compared in the experiments: The first, referred to as *freq*, is given by Eq. (5) and simply uses the observed sample distribution as the estimation of the 'true' probability. The second, referred to as *mixture*, is given by Eq. (7) and takes the discounting effect into account in the estimation.

(3) the effect of compound terms

Two sets of documents are prepared for comparison: For the first, words are extracted

[3] http://ais.gmd.de/~thorsten/svm_light/

using simple stemming and stop-words removal procedures, the resulting 20,507 basic words being used as feature terms. For the second, the documents are first morphologically analyzed using Brill Tagger V.1.14. Then, compound words are extracted using a set of pre-determined matching patterns. After applying the same stemming and stop-words removal procedures, the resulting 99,000 words are all used as feature terms. The extraction methods are referred to as *basic words* and *compound words*, respectively.

Table 1 summarizes the results. It shows that the performance is consistently better with *compound words* cases than with *basis words* cases; and also with *mixture* estimation than with *freq* estimation. Also, comparing the performance of PWI and SVM, the advantage of the latter is obvious. Since both methods employ similar formulae that are based on the inner product of term vectors, the difference may be attributed to the fine parameter adjustment of SVM. While PWI simply averages all the documents in the same category, SVM carefully calculates the total errors of training documents in the margin region. The computational costs of these two methods are compared in our next experiments using NTCIR-J1.

Table 1: Results with Reuters-21578.

term extraction methods	probability estimation methods	categorization methods	
		PWI	*SVM*
basic words	*freq*	0.782	0.871
	mixture	0.794	0.873
compound words	*freq*	0.806	0.873
	mixture	0.814	0.875

The performance values reported in the past studies are: 0.776 for Rocchio [4], 0.820 for Ripper, and 0.827 for sleeping experts (Cohen and Singer, 1999); 0.773 for naive Bayes method and 0.820 for the stochastic decision tree with ESC (Li and Yamanishi, 2000). Also, with slightly different but rather advantageous conditions, 0.799 for Rocchio and 0.864 for SVM (Joachims, 1998); 0.796 for naive Bayes and 0.860 for SVM (Yang and Liu, 1999). Based on these figures, we can confirm that the performance of SVM in our experiments is consistent with the past studies, and also that the proposed PWI-based naive method performs better than the traditional naive Bayes or Rocchio methods, and is quite good as a method without learning.

[4] *tf-idf*-based method with relevance feedback.

4.2 NTCIR-J1

With NTCIR-J1, each document belongs to the one and only category. Then, we formulate the categorization task as a single 24-class problem for PWI, and the 24 numbers of 2-class problems for SVM. For PWI, a category with the highest PWI value is selected. For SVM, the values of the decision function for the 24 2-class problems are compared, and the category with the highest score is selected. The performance is measured by the ratio of correct judgments, i.e., the number of documents classified into the class that they originally belong to, divided by the number of tested documents. ChaSen Version 2.02 is used as the Japanese morphological analyzer. Other conditions are the same as in the previous experiments.

The large scale of NTCIR-J1 allows us to examine the performance under variable sizes of training data. The sizes are: $size$ = 1,000, 2,000, 5,000, 10,000, 20,000, 50,000, and 309,999 (the maximum). For each of $1{,}000 \sim 50{,}000$ sizes, 10 different sets are randomly picked from the whole training data. The smallest size 1,000 is determined so that at least five documents are sampled for each category. The average number of extracted terms for each size is shown in Figure 2. It can be seen that the number of distinct terms is almost proportional to the size of the training data, with the maximum value being nearly four million terms. Obviously, we need a scalable method to manipulate low frequency terms in such large-size data.

Table 2 illustrates the effect of low frequency terms on the categorization tasks. It shows the performance of PWI with full size training data, where only terms such as $f(w) > K$ are used as features. The performance increases monotonically as the value of

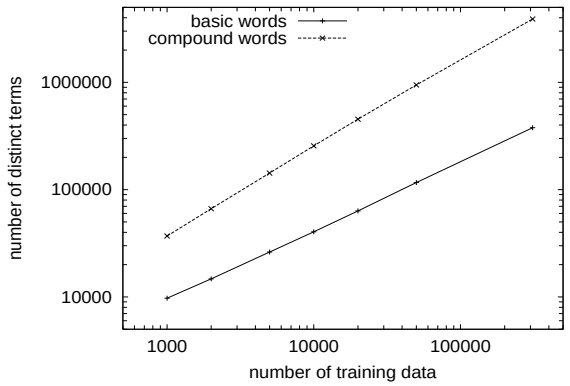


Figure 2: The number of distinct terms.

K becomes greater, showing the advantage of utilizing low frequency terms. Although not shown here, we have also tested the case when information gain is used for selecting terms, but the tendency remains the same. Also, comparing the performance of *compound* and *basic words*, it becomes clear that the former shows better performance for approximately the same number of feature terms.

Table 2: Effect of low frequency terms.

terms used as features	*basic words*		*compound words*	
	number of terms	performance	number of terms	performance
$f(w) > 0$	377,603	0.7596	3,754,779	0.8149
$f(w) > 1$	166,958	0.7557	1,236,856	0.8093
$f(w) > 2$	114,506	0.7537	722,014	0.8034
$f(w) > 3$	89,811	0.7524	514,167	0.8007
$f(w) > 4$	75,101	0.7518	398,858	0.7958
$f(w) > 5$	65,188	0.7505	327,553	0.7926
$f(w) > 10$	42,291	0.7473	175,229	0.7830
$f(w) > 15$	32,971	0.7462	121,567	0.7785
$f(w) > 20$	27,712	0.7439	93,695	0.7737

Figure 3 compares the performance for different training sizes where the *mixture* model is used for the estimation. We can confirm that significant improvement is obtained by considering *compound words* instead of *basic words*: Looking at the results in more detail, *compound words* outperforms *basic words* in all the 61 runs for PWI. The same is also true for SVM with $size \geq 5000$, although the difference is not clear for SVM with $size = 1000, 2000$. The performance of SVM is better than the performance of PWI, showing the advantage of machine learning approaches in text categorization problems. Unlike the case with Reuters-21578, the difference between the *freq* and the *mixture* estimation methods was not so obvious with NTCIR-J1.

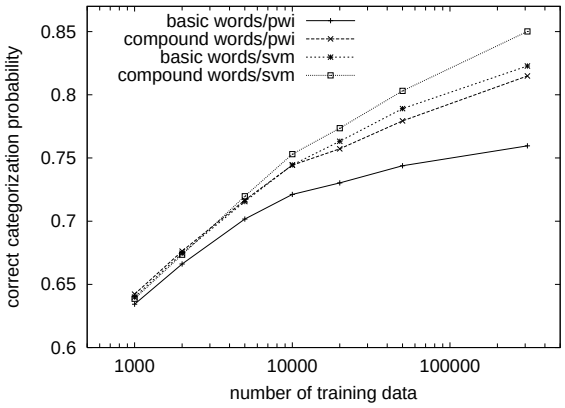


Figure 3: Comparison of the performance.

Figure 4 compares the average execution time using Pentium III 696MHz (Linux). Note that the execution time is plotted using log-scale. Despite the good scalability of SVM compared with other dedicated learning methods, the computational cost of SVM is still much greater than of PWI. For example, the execution time for the largest dataset is 135 seconds for PWI and 80,131 seconds (about one day) for SVM.

In addition to the computation time shown in Figure 4, PWI and SVM commonly require about 35 minutes for morphological analysis and term extraction, 18 minutes for indexing the whole data set. Also, SVM requires another 18 minutes for calculating normalized weights for the largest data set. Note that (i) as a language with no explicit word boundaries, such pre-processing is quite common in Japanese, and also (ii) the computational cost of standard NLP is usually proportional to the text size, and consequently the relative cost for learning becomes dominant when the size of the training data becomes large.

Figure 5 compares the performance averaged over 24 classes, assuming the test data contains equal numbers of documents from each class. *Compound words* are used as features. It can be seen that the *mixture* estimation is better than the *freq* estimation,

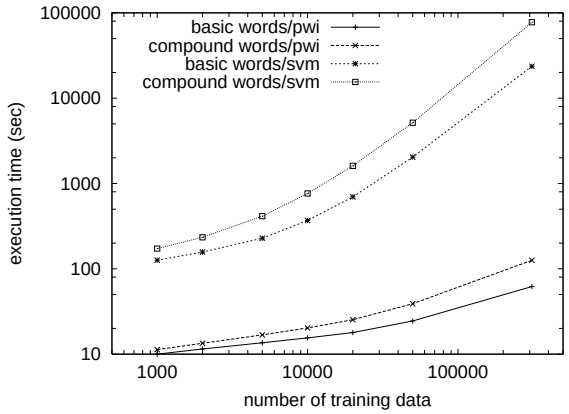


Figure 4: Comparison of the execution time.

and also that *PWI* is better than *SVM* in terms of macro averaging. The results suggest that there is more than a simple trade-off of 'the computational cost' and 'the performance' between *SVM* and *PWI*: While *SVM* exploits the class size distribution in its operation, *PWI* treats all the classes equally, regardless of the distribution in the training data.

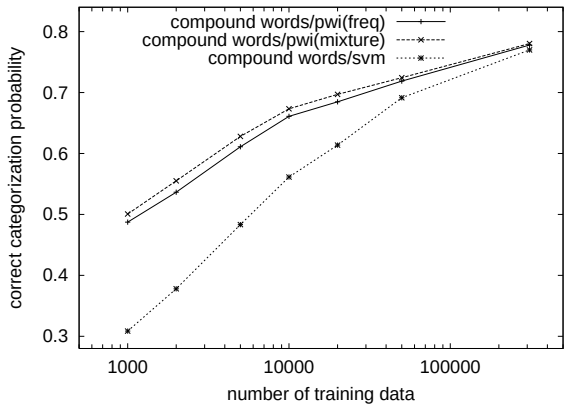


Figure 5: Effect of discounting on the macro averaged performance.

5 Discussion

In this paper, we have first analyzed the statistical nature of terms in text and proposed a naive text categorization criterion to exploit information carried by low frequency terms. We have also investigated the advantage of considering morphological information and probabilistic language modeling at the pre-processing of text categorization.

At present, we consider all the possible compound words as features. Future issues include application of more dedicated methods of term extraction, and use of different sampling and feature selection strategies that can reduce the execution cost of SVM.

References

Lewis, D. D. and Singer, Y. 2000. *Introduction to Machine Learning for Information Retrieval*, Tutorial in ACM SIGIR 2000.

Nagata, M. and Taira, H. 2001. *Text Classification – Showcase of Learning Theories –*, Information Processing Society of Japan Magazine, 42(1): 32-37.

Joachims, T. 1998. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, Proc. of ECML' 98, 137-142.

Cohen, W.W. and Singer, Y. 1999. *Context-sensitive learning methods for text categorization*, ACM Trans. on Information Systems, 17(2): 141-173.

Li, H. and Yamanishi, K. 1999. *Text Classification Based on ESC*, Proc. of 1999 Workshop on Information-Based Induction Sciences, 239-244.

Yang, Y. and Liu, X. 1999. *A Re-examination of Text Categorization Methods*, Proc. of ACM SIGIR 1999, 42-49.

Fung, P. and McKeown, K. 1996. *A technical word and term translation aid using noisy parallel corpora across language groups*, The Machine Translation Journal, 12(1-2): 53–87.

Aizawa, A. 2000. *The Feature Quantity: An Information Theoretic Perspective of Tfidf-like Measures*, Proc. of ACM SIGIR 2000, 104-111.

Kita, K. 1999. *Probabilistic language models*. University of Tokyo Press.

Brill, E. 1994. *Some advances in rule-based part of speech tagging*, Proc. of AAAI' 94.

Matsumoto, Y., Kitauchi, A., Yamashita, T., Hirano, Y., Matsuda, K., and Asahara, M. 1999. *Morphological analysis system ChaSen 2.0.2 users manual*, NAIST Technical Report, NAIST-IS-TR99012, Nara Institute of Science and Technology.