# Some Effective Techniques for Naive Bayes Text Classification

Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung Hyon Myaeng

**Abstract**—While naive Bayes is quite effective in various data mining tasks, it shows a disappointing result in the automatic text classification problem. Based on the observation of naive Bayes for the natural language text, we found a serious problem in the parameter estimation process, which causes poor results in text classification domain. In this paper, we propose two empirical heuristics: per-document text normalization and feature weighting method. While these are somewhat ad hoc methods, our proposed naive Bayes text classifier performs very well in the standard benchmark collections, competing with state-of-the-art text classifiers based on a highly complex learning method such as SVM.

**Index Terms**—Text classification, naive Bayes classifier, Poisson model, feature weighting.

## 1 Introduction

Naive Bayes has been one of the popular machine learning methods for many years. Its simplicity makes the framework attractive in various tasks and reasonable performances are obtained in the tasks although this learning is based on an unrealistic independence assumption. For this reason, there also have been many interesting works of investigating naive Bayes. Especially, [1] shows that naive Bayes can perform surprisingly well in the classification tasks where the probability itself calculated by the naive Bayes is not important.

With this background, text classifiers based on naive Bayes have been studied extensively by some researchers [2], [3], [4]. In their naive Bayes classifiers, a document is considered as a binary feature vector representing whether each word is present or absent. This naive Bayes is called *multivariate Bernoulli naive Bayes*.

Although this model is the closest one to the traditional naive Bayes, it is not equipped to utilize term frequencies in documents. Thus, the multinomial model has been introduced as an alternative naive Bayes text classifier. In recent years, many researchers usually regard it as the standard naive Bayes text classifier [4], [5], [6], [7]. However, their performances are not as good as some other statistical learning methods such as nearest-neighbor classifiers [8], support vector machines [9], and boosting [10]. Since naive Bayes is very efficient and easy to implement compared to other learning methods, it is worthwhile to improve the performance of naive Bayes in text classification tasks, especially for various practical applications such as spam filtering or news article classification.

In our experience with the multinomial naive Bayes text classification, we identified two serious problems. The first is its rough parameter estimation. In the multinomial model, the word probabilities for a class are estimated by calculating the likelihood in the entire positive training documents. It means that all the training documents for a given class are merged into a mega-document, serving as a unique example for the class. This means that parameter estimation in this model is affected more by long documents than by short documents; the longer a document, the more terms participate in parameter estimation.

The second problem lies in handling rare categories that contain only a few training documents. It is not difficult to imagine that a probabilistic model such as naive Bayes may not work well with an insufficient number of training examples. When a category has only a few training documents, a few informative words that are useful to determine the category of a document usually are buried by the relatively large number of noisy terms and their unreliable probability estimates. Thus, some statistical measures, like information gain or the chi-square test, are applied to selecting a subset of words, i.e., feature set.

With the feature selection strategy, however, one should determine the number of feature words for each class and balance it with other classes. Moreover, a feature term is either selected or not with no attempt to use the degree of importance that can be calculated in various ways. All the selected words by feature selection are considered equally important.

In this paper, we first propose a per-document length normalization approach by introducing multivariate Poisson model for naive Bayes text classification and a weight-enhancing method to improve performances on rare categories where the model parameters are unreliable.

- *S.-B. Kim, K.-S. Han, and H.-C. Rim are with the Department of Computer Science and Engineering, College of Information and Communications, Korea University, Anam-dong 1-ka, Sungpuk-gu, Seoul, 136-701, Korea. E-mail: {sbkim, kshan, rim}@nlp.korea.ac.kr.*
- *S.-H. Myaeng is with the Information and Communications University, F637, 119 Munji-ro, Yuseong-gu, Daejeon, 305-732, Korea. E-mail: myaeng@icu.ac.kr.*

*Manuscript received 26 Mar. 2005; revised 22 Nov. 2005; accepted 30 May 2006; published online 19 Sept. 2006.*
*For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0111-0305.*

1041-4347/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

## 2 Critiques for Multinomial Naive Bayes Text Classification

A naive Bayes classifier is a well-known and practical probabilistic classifier and has been employed in many applications. It assumes that all attributes (i.e., features) of the examples are independent of each other given the context of the class, i.e., an independence assumption. It has been shown that naive Bayes under zero-one loss performs surprisingly well in many domains in spite of the independence assumption [1].

In the context of text classification, the probability that a document $d_j$ belongs to a class $c$ is calculated by the *Bayes' theorem* as follows:

$$p(c|d_j) = \frac{p(d_j|c)p(c)}{p(d_j)} = \frac{p(d_j|c)p(c)}{p(d_j|c)p(c) + p(d_j|\bar{c})p(\bar{c})} = \frac{\frac{p(d_j|c)}{p(d_j|\bar{c})} \cdot p(c)}{\frac{p(d_j|c)}{p(d_j|\bar{c})} \cdot p(c) + p(\bar{c})}. \tag{1}$$

Now, if we define a new function $z_{jc}$,

$$z_{jc} = \log \frac{p(d_j|c)}{p(d_j|\bar{c})}, \tag{2}$$

then (1) can be rewritten as

$$P(c|d_j) = \frac{e^{z_{jc}} \cdot p(c)}{e^{z_{jc}} \cdot p(c) + p(\bar{c})}. \tag{3}$$

Using (3), we can get the posterior probability $p(c|d_j)$ by calculating $z_{jc}$.

With this framework, [4] arranged the naive Bayes text classification models according to derivation strategy for $z_{jc}$. They designated the pure naive Bayes classifier as the *multivariate Bernoulli model*, and the unigram language model classifier as the *multinomial model*. In the multivariate model, a document is considered as a $|V|$-dimensional vector $D = (w_1, w_2, \cdots w_{|V|})$ which is the result of $|V|$ independent Bernoulli trials, where $|V|$ is the vocabulary size and $w_k$ is a binary variable representing the occurrence or nonoccurrence of the $k$th word in the vocabulary. The most serious problem with this approach is that it cannot utilize term frequency information.

Contrary to the multivariate Bernoulli model, the multinomial model treats a document as an ordered sequence of word occurrences, with each word occurrence as an independent trial. In other words, a document is drawn from a multinomial distribution of words. In the multinomial model, $z_{jc}$ is computed as follows [4]:

$$z_{jc} = log \frac{p(d_j|c)}{p(d_j|\bar{c})} = log \frac{P(|d_j|)|d_j|! \prod_{i=1}^{|V|} \frac{p(w_i|c)^{tf_{ij}}}{tf_{ij}!}}{P(|d_j|)|d_j|! \prod_{i=1}^{|V|} \frac{p(w_i|\bar{c})^{tf_{ij}}}{tf_{ij}!}} = \sum_{i=1}^{|V|} tf_{ij} \cdot log \frac{p(w_i|c)}{p(w_i|\bar{c})}, \tag{4}$$

where $tf_{ij}$ represents the number of $t_i$ in the document $d_j$.

In (4), the parameters $p(w_i|c)$ and $p(w_i|\bar{c})$ are estimated with Laplacian smoothing as follows:

$$p(w_i|c) = \frac{\theta + \sum_{j=1}^{|\mathcal{D}|} tf_{ij} P(y_j = c|d_j)}{\theta \cdot |V| + \sum_{k=1}^{|V|} \sum_{j=1}^{|\mathcal{D}|} tf_{kj} P(y_j = c|d_j)}, \tag{5}$$

$$p(w_i|\bar{c}) = \frac{\theta + \sum_{j=1}^{|\mathcal{D}|} tf_{ij} P(y_j \neq c|d_j)}{\theta \cdot |V| + \sum_{k=1}^{|V|} \sum_{j=1}^{|\mathcal{D}|} tf_{kj} P(y_j \neq c|d_j)}. \tag{6}$$

In the above equation, the value of $\theta$ is determined empirically.[1]

This model discards information about the order of the words, but takes the term frequency information of each word in the document. When calculating the probability of a document given a class, each probability of a word occurrence in the document given the class is sequentially multiplied from the first word to the last word of the document.

Although the categorizers based on the multinomial model significantly outperform those based on the multivariate model, the performance has still been unsatisfactory, specifically when there are few training documents. We found that there is a critical problem in the process of parameter estimation for the multinomial text classification model: This model considers the whole of the positive or negative documents as a huge document, and estimates the parameters from the document. Our work mainly focuses on the development of new naive Bayes model allowing more reasonable parameter estimation.

## 3 Proposed Methods for Improving Naive Bayes Text Classifier

### 3.1 Multivariate Poisson Model for Text Classification

The Poisson distribution is widely used to model the number of occurrences of a certain phenomenon in a fixed period of time or space, such as the number of telephone calls received at a switchboard during a given period or the number of defects in a fixed length of cloth or paper. The use of the Poisson model has been widely investigated in the IR literature since the frequency of occurrence of a particular word in a document can be explained by the Poisson distribution. Unfortunately, it has rarely been studied for text classification task [3] and it motivates us to investigate the use of Poisson model for learning the naive Bayes text classifier.

We assume that a document is generated by a multivariate Poisson model. A document $d_j$ is represented as a random vector which consists of Poisson random variables $X_{ij}$, where $X_{ij}$ has the value of within-document-frequency $f_{ij}$ for the $i$th term $t_i$ as follows:

$$p(d_j) = p(X_{1j} = f_{1j}, X_{2j} = f_{2j}, \cdots, X_{|V|j} = f_{|V|j}). \tag{7}$$

If we assume that each of the variables $X_{ij}$ is independent of one another, i.e., using an independence assumption, the probability of $d_j$ is calculated as follows:

[1] In our experiments, we set 0.0001 to $\theta$ where the classifier performs quite well.

$$p(d_j) = \prod_{i=1}^{|V|} p(X_{ij} = f_{ij}), \tag{8}$$

where $|V|$ is the vocabulary size and each $p(X_{ij} = f_{ij})$ is given by

$$p(X_i = f_{ij}) = \frac{e^{-\lambda_{ic}} {\lambda_{ic}}^{f_{ij}}}{f_{ij}!}, \tag{9}$$

where $\lambda$ is the *Poisson mean*.

As a result, the $z_{jc}$ function of (2) is rewritten using (8) and (9) as follows:

$$\begin{aligned} z_{jc} &= \sum_{i=1}^{|V|} log \frac{p(X_i = f_{ij}|c)}{p(X_i = f_{ij}|\bar{c})} = \sum_{i=1}^{|V|} log \frac{e^{-\lambda_{ic}} {\lambda_{ic}}^{f_{ij}}}{e^{-\mu_{ic}} {\mu_{ic}}^{f_{ij}}} \\ &= \sum_{i=1}^{|V|} (\mu_{ic} - \lambda_{ic}) + \sum_{i=1}^{|V|} f_{ij} \cdot \log \frac{\lambda_{ic}}{\mu_{ic}}, \end{aligned} \tag{10}$$

where $\lambda_{ic}$ and $\mu_{ic}$ are the *Poisson means* for the term $t_i$ in the positive class $c$ and for the negative class $\bar{c}$, respectively.

From the definition of the Poisson distribution, the Poisson parameter $\lambda_{ic}$ or $\mu_{ic}$ is the average number of occurrences of $t_i$ in the positive or negative document of a given unit length, respectively. It means that both $\sum_{i=1}^{|V|} \lambda_{ic}$ and $\sum_{i=1}^{|V|} \mu_{ic}$ are the same as a unit length. Thus, we can drop the first term of (10), resulting in

$$z_{jc} = \sum_{i=1}^{|V|} f_{ij} \cdot \log \frac{\lambda_{ic}}{\mu_{ic}}. \tag{11}$$

As we can see, (11) is very similar to the traditional multinomial model shown in (4). For example, if we take the actual frequency of $t_i$ in $d_j$ as $f_{ij}$, and the ratio of the frequency of $t_i$ in the positive (or negative) training corpus as the Poisson parameter $\lambda_{ic}$ (or $\mu_{ic}$), our proposed text classification model becomes the traditional multinomial model. It means that the multivariate Poisson text classification model is a more flexible model than the traditional model, which can be expanded by adopting the various methods to estimate $f_{ij}$, $\lambda_{ic}$, and $\mu_{ic}$. We will introduce our method of estimating parameters in the following subsections.

### 3.2 Parameter Estimation Using Normalized Term Frequencies

From the definition of the Poisson distribution, the Poisson parameter indicates the average number of events during the observation of a fixed period. Therefore, the Poisson parameters $\lambda_{ic}$ and $\mu_{ic}$ of our text classification model indicate how many times a term $t_i$ occurs in the positive and negative training documents on average.

One possible approach is that we simply consider the relative frequency of $t_i$ in the training documents as the Poisson parameter $\lambda_{ic}$ or $\mu_{ic}$ as follows:

$$\begin{aligned} \lambda_{ic} &= \frac{\#\text{occurrences for } t_i \text{ in the pos. training documents}}{\#\text{total tokens in the pos. training documents}}, \\ \mu_{ic} &= \frac{\#\text{occurrences for } t_i \text{ in the neg. training documents}}{\#\text{total tokens in the neg. training documents}}. \end{aligned} \tag{12}$$

TABLE 1
Various Length Normalization Methods

| Method | $\hat{f}_{ij}$ |
|---|---|
| RF | $\frac{tf_{ij}}{NF}$ |
| SRF | $\frac{tf_{ij}+\theta}{NF+\theta\cdot\lvert V\rvert}$ |
| BM25 | $\frac{(k_1+1)\cdot tf_{ij}}{k_1\cdot(\alpha+(1-\alpha)\cdot dl_j/avdl)+tf_{ij}}$ |
| PLN | $\frac{(1+log(tf))/(1+log(avtf))}{(1-\alpha)\cdot avdu+\alpha\cdot du_j}$ |

By this approach, the $f_{ij}$ value in our model is the actual term frequency of $t_i$ in $d_j$. If we further adopt a smoothing method to estimate the above $\lambda_{ic}$ and $\mu_{ic}$, our model becomes the same as the traditional multinomial model. In other words, the multinomial model is a special case of the proposed multivariate Poisson model with a particular parameter estimation method.

Another possible parameter estimation is a *per-document length normalization approach*, which makes our new text categorizer different to the traditional multinomial text classifier. We first normalize the term frequencies in each document according to the document length. Second, the normalized frequencies from each document are linearly combined according to the probability that each document belongs to the document set where we are going to estimate the parameter from as follows:

$$\begin{aligned} \lambda_{ic} &= \sum_{j=1}^{|D|} \hat{f}_{ij} \cdot P(d_k = d_j | d_k \in D_c) = \frac{1}{|D_c|} \cdot \sum_{j=1}^{|D_c|} \hat{f}_{ij}, \\ \mu_{ic} &= \sum_{j=1}^{|D|} \hat{f}_{ij} \cdot P(d_k = d_j | d_k \in D_{\bar{c}}) = \frac{1}{|D_{\bar{c}}|} \sum_{j=1}^{|D_{\bar{c}}|} \hat{f}_{ij}, \end{aligned} \tag{13}$$

where $\hat{f}_{ij}$ is a normalized frequency of $t_i$ in $d_j$ as explained in Table 1 and $D$, $D_c$, and $D_{\bar{c}}$ are the set of training documents, the set of positive training documents, and the set of negative training documents, respectively. We assume a uniform prior distribution over the positive or the negative document collection, i.e., each document is equally important to learn the classifier. It is similar to other machine learning algorithms which consider each sample as an equally important instance to learn classifiers.

It should be noticed that per-document length normalization with prior distribution over a positive or negative document set makes the proposed model remarkably different from the traditional multinomial model. The traditional multinomial model considers each term occurrence as an equally important event, which results in giving different importance to each training document for the learning classifier according to the length of each document. On the other hand, the term occurrences in the short training documents are treated as more informative events than those in the long documents with the proposed model using per-document length normalization and uniform prior for all documents. Since the short documents usually have fewer unnecessary terms and probably consist of more essential terms to represent a concept of a topic, we believe that our proposed model using per-document length normalization with uniform prior is more appropriate than the traditional multinomial model.

In order to obtain normalized term frequencies, we have tested several length normalization heuristics that have been developed for text retrieval systems. Table 1 shows four different length normalization methods we have tested in our experiments: `RF` (relative frequency), `SRF` (smoothed relative frequency), `BM25` used in the Okapi system [11], and `PLN` (pivoted length normalization) used in the SMART system [12].[2] In this table, $NF$ used in `RF` and `SRF` indicates a normalization factor.

While `RF` and `SRF` just normalize the term frequency by dividing it by the normalization factor, `BM25` and `PLN` transform the term frequency based on some theoretical or empirical reasons, with some successful results in the previous works. The `BM25` formula is derived from 2-Poisson probabilistic model and the `PLN` from the investigation of the relationship between document length and the probability of relevance in many test collections.

On the grounds that both the number of tokens and the number of unique terms have been used for the normalization of term frequencies in the literature, we have also designed two different versions of linearly interpolated $NF$ using average document length over the collection as follows:

$$\begin{aligned} NF_l &= \alpha \cdot avdl + (1-\alpha) \cdot dl_j, \\ NF_u &= \alpha \cdot avdu + (1-\alpha) \cdot du_j, \end{aligned} \tag{14}$$

where $avdl$ and $avdu$ indicate the average number of tokens in a document and the average number of unique terms in a document over the collection, respectively. $dl_j$ and $du_j$ means the number of tokens and unique terms in a document $d_j$. This linear interpolation smoothes the length of each document using the characteristics of document collection. It becomes the denominator of `PLN` and the denominator of `BM25` also has a form of showing the same behavior to $NF_l$ and $NF_u$. We show some experimental results using all these normalization methods in Section 4.

## 3.3 Considerations of Feature Weights

Feature selection has been an issue in text classification because it can reduce the complexity of the machine learning algorithms. Yang and Pedersen [13] investigated some measures to select useful term features, including mutual information, information gain, and $\chi^2$-statistics, etc. Koller and Sahami [14] proposed an optimal feature selection method to minimize the loss of the predictive information in reducing feature space.

It remains true that using all the word features usually makes the classifier work better. Joachims [15] claimed that there are no useless term features and it is preferable to use all word features. Even Koller and Sahami [14] do not claim that any machine learning method using their optimal feature selection can be superior to those using all the word features. For multivariate naive Bayes text classification, it has been shown empirically that feature selection is useful to improve the classification performance for some categories [4]. However, we think that it is caused by the large accumulated error from poor parameter estimation of the multivariate naive Bayes text classification model.

TABLE 2
Two-Way Contingency Table

| | presence of $t_i$ | absence of $t_i$ |
|---|---|---|
| labeled as $c$ | w | x |
| not labeled as $c$ | y | z |

Moreover, the process of feature selection presents several practical problems. The number of features for each category should be determined while maintaining the balance among the numbers of features. Once features are selected, they are often treated equally, ignoring the degree of importance used as the cutoff criterion.

To alleviate the problems and the potential drawback of using binary features, we propose a feature weighting approach rather than a feature selection and have modified (11) follows:

$$z_{jc} = \sum_{i=1}^{|V|} \frac{fw_{ic}}{\mathrm{FW}_c} \cdot f_{ij} \cdot log \frac{\lambda_i}{\mu_i}, \tag{15}$$

where $fw_{ic}$ is the weight of feature word $w_i$ for the class $c$ and $\mathrm{FW}_c$ is the normalization factor, that is, $\sum_{i=1}^{V} fw_{ic}$.

Our model with feature weights, as reflected in (15), can be viewed as a generalized version of the model using feature selection. If we give one or zero of $fw_{ij}$ to all the feature words or nonfeature words, respectively, we end up with a text classifier using a subset of the vocabulary by feature selection.

### 3.3.1 Feature Weighting Scheme

In our proposed multivariate Poisson approach to Naive Bayesian classification, we test the following three measures to weight each term feature: information gain, $\chi^2$-statistics, and an extended version of risk ratio. Information gain and $\chi^2$-statistics are selected because they are known to be most effective for some models like kNN and LLSF [13], while a risk ratio is newly tested in text categorization domain.

Information gain (or average mutual information) is an information-theoretic measure defined by the amount of reduced uncertainty given a piece of information. Information gain for a term given a class, which becomes the weight of the term, is calculated using a document event model as follows:

$$\begin{aligned} fw_{ic} &= H(C) - H(C|W_i) \\ &= \sum_{c_s \in \{c,\bar{c}\}} \sum_{w_t \in \{w_i,\bar{w_i}\}} p(c_s, w_s) \log \frac{p(c_s, w_t)}{p(c_s)p(w_t)}, \end{aligned}$$

where, for example, $p(c)$ is the number of documents belonging to the class $c$ divided by the total number of documents and $p(\bar{w})$ is the number of documents without the term $w$ divided by the total number of documents, etc.

The second measure we used is $\chi^2$-statistics developed for the statistical test of the hypothesis. In text classification, given a two-way contingency table for each term $t_i$ and the class $c$ as represented in Table 2, $fw_{ic}$ is calculated as follows:

[2] Length normalization techniques used in the Okapi and SMART systems are the most widely used ones in the literature.

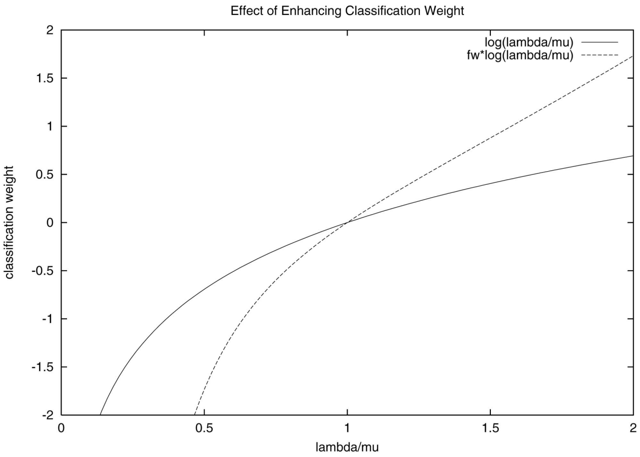

Fig. 1. Classification weight enhancing effect.

$$fw_{ic} = \frac{(wz - xy)^2}{(w+x)(w+y)(x+z)(y+z)}, \tag{16}$$

where $w$, $x$, $y$, and $z$ indicate the number of documents for each cell in Table 2.

While the above two measures have been widely tested in text categorization domain, we have tested additional measure: a extend version of *risk ratio*(ExtRR) as follows:[3]

$$fw_{ic} = \frac{\lambda_{ic}}{\mu_{ic}} + \frac{\mu_{ic}}{\lambda_{ic}}. \tag{17}$$

A risk ratio is a simple but widely used testing measure in a biomedical researches [16], but not in a text categorization domain. A risk ratio is known for correcting the odds ratio measure, which is tried in the several text categorization literature [17], [18].

With this ExtRR measure, our $z_{jc}$ is finally defined as follows:

$$z_{jc} = \sum_{i=1}^{|V|} \frac{1}{\mathrm{FW}_c} \cdot \hat{f}_{ij} \cdot \left(\frac{\lambda_{ic}}{\mu_{ic}} + \frac{\mu_{ic}}{\lambda_{ic}}\right) \cdot log\frac{\lambda_{ic}}{\mu_{ic}}. \tag{18}$$

This measure indicates the sum of the ratio of two Poisson parameters and its reciprocal. The first term represents how term $t_i$ is more likely to be presented in the class $c$ compared to outside of the class $c$ and the second term represents the reverse. With this measure, $fw_{ic}$ has the minimum value of 2.0 for the term which does not have any information to predict the class of document, i.e., $\lambda_{ic} = \mu_{ic}$. As the gap between $\lambda_{ic}$ and $\mu_{ic}$ increases, $fw_{ic}$ gets higher.

The most important property of ExtRR measure is that this measure enhances its original classification weight (i.e., $log\frac{\lambda_{ic}}{\mu_{ic}}$) in a predictable way, as shown in Fig. 1. In this figure, the x-axis and y-axis represents $\frac{\lambda_{ic}}{\mu_{ic}}$ and the classification weight, respectively. While the solid curve plots the original classification weight $log\frac{\lambda_{ic}}{\mu_{ic}}$ according to $\frac{\lambda_{ic}}{\mu_{ic}}$, the dotted curve plots the new classification weight by ExtRR, i.e., $(\frac{\lambda_{ic}}{\mu_{ic}} + \frac{\mu_{ic}}{\lambda_{ic}}) \cdot log\frac{\lambda_{ic}}{\mu_{ic}}$. This figure shows that the new classification weight is emphasizing the ratio of the two Poisson parameters $\lambda_{ic}$ and $\mu_{ic}$, which may enhance the discriminative power of the proposed classifier.

In addition, the ExtRR measure has a computational advantage since it can be computed directly from the values of existing model parameters $\lambda_{ic}$ and $\mu_{ic}$. These parameters may be determined manually or using cross validation in each class.

## 4 Experimental Results

### 4.1 Data and Evaluation Measure

We ran experiments on the two commonly used corpora in text categorization: Reuters21578[4] and 20 Newsgroups.[5] The Reuters21578 collection consists of 21,578 news articles in 1987 from Reuter Ltd. For the experiments, the documents were separated according to the "ModApte" split to have 9,603 training documents and 3,299 test documents. Ninety categories are assigned to the training documents, but the distribution is skewed. The 20 Newsgroups collection consists of 19,997 Usenet articles collected from 20 different newsgroups. About 1,000 messages from each of the 20 newsgroups were chosen at random and partitioned by the newsgroup names. Unlike the Reuters collection, the distribution of documents for the classes is uniform. We used two-thirds of the documents for training and the rest for testing. Stopword removal and stemming were applied to both collections. To evaluate the performance, we calculated the F1 measure for each category and micro/macroaveraged the performance [19]. We have chosen these two collections because they are quite different from each other. The Reuters collection consists of relatively short documents (861 bytes on average) and the class distribution is quite skewed. In contrast, the 20 Newsgroups collection consists of relatively long documents (1,892 bytes on average) and the class distribution is uniform.

We use both micro and macroaverages since they show different aspects of the classifiers depending on the nature of the test collections. When the distribution of the training documents is skewed, the performance of a category having a large amount of test documents dominates microaveraged F1 performance. On the contrary, macroaveraged F1 performance is just a simple averaged value over all the categories. In the Reuters21578 collection, macroaveraged F1 performance is greatly influenced by the performance on rare classes.

### 4.2 Effect of Per-Document Length Normalization

Fig. 2 shows the microaveraged F1 and macroaveraged F1 performances of the proposed Poisson text classifiers on Reuters21578 collection. In this experiment, we compare `RF` and `SRF` normalization using normalization factors $NF_l$ and $NF_u$ described in Section 3.2. `RF-l` and `SRF-l` use $NF_l$, while $\mathtt{RF} - \mathtt{u}$ and $\mathtt{SRF} - \mathtt{u}$ use $NF_u$ for the normalization factor. We measured the performances according to the normalization parameter $\alpha$. In addition, `multin`. indicates the baseline performance, i.e., the traditional multinomial naive Bayes text classifier.

[3] We have extended the original risk ratio, i.e., $\frac{\lambda_{ic}}{\mu_{ic}}$, by adding the reciprocal term so that it also emphasizes the negative evidence.

[4] http://www.research.att.com/~lewis.

[5] http://www-2.cs.cmu.edu/afs/cs/project/theo-3/www/.

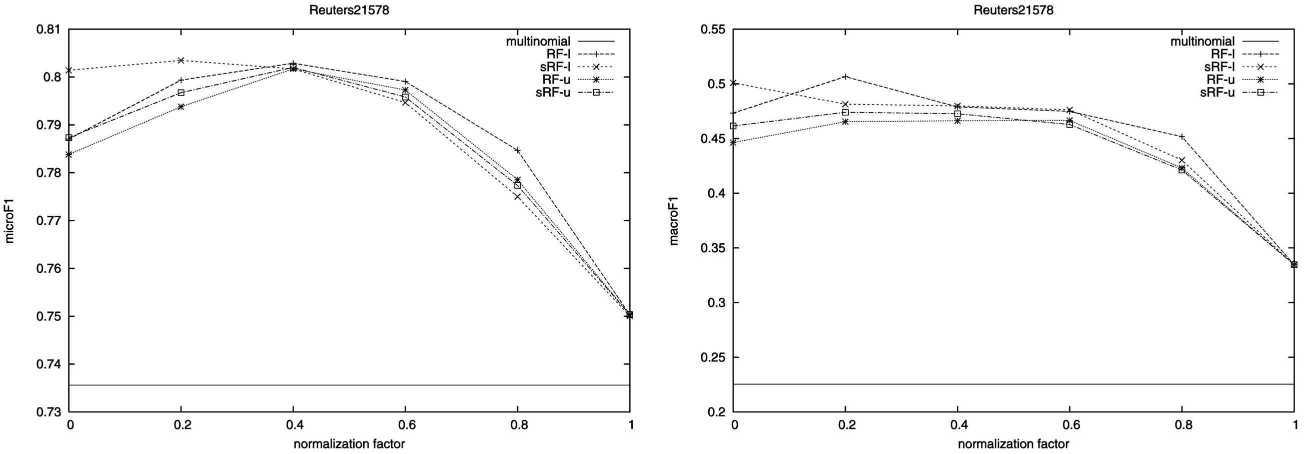

Fig. 2. MicroF1 and MacroF1 performances of per-document length normalization in Reuters21578 collection.

We observe that all the variants of the proposed classifiers significantly outperform the traditional multinomial naive Bayes at least when this collection is used. In the upper graph showing the microF1 values, $\mathtt{SRF}-\mathtt{l}$ at $\alpha$ of 0.2 achieves the best performance. $\mathtt{RF}-\mathtt{u}$ and $\mathtt{SRF}-\mathtt{u}$ also achieve better performance than baseline performance and less so than the $\mathtt{Rf}-\mathtt{l}$ or $\mathtt{SRF}-\mathtt{l}$, but trivial. It means that there is no significant difference between using the number of tokens and the number of unique terms. In addition, the proposed per-document length normalization is more effective when the normalization parameter $\alpha$ is lower than 0.5. Since a lower $\alpha$ value means a higher weight on the per-document length normalization, we can conclude that normalization based on the length of individual documents is quite effective.

In the lower graph showing the macroF1 values, RF-l at $\alpha$ of 0.2 achieves the best performance similarly in the case of microF1. Another observation is that RF-l and SRF-l show better results than RF-u and $\mathtt{SRF}-\mathtt{u}$. The biggest difference between the microF1 and macroF1 is that the performance increase by the normalization over the baseline is much greater in the case of macroF1 (0.2238 for the baseline versus 0.5066 for RF-l). Since macroF1 values in the Reuters21578 collection tend to be dominated by a large number of small categories, which have a small number of training documents, we can conclude that our proposed normalization methods are quite effective, particularly in the categories where the number of positive training documents is small. Subsequent experiments are reported below with fuller explanations.

It should be noticed that our proposed classifier achieves much higher macroF1 performance even when the normalization parameter is 1.0, which indicates that term frequencies are not normalized using the lengths of documents where they occur. In this case, Poisson parameters $\lambda_{ic}$ and $\mu_{ic}$ in our model just become the average value of actual term frequencies of a term $t_i$ in the positive and negative training documents. The fact that even the proposed model without any information about document length outperforms the traditional multinomial naive Bayes model in terms of the macroF1 measure reveals that the way of parameter estimation in a traditional model has a serious problem for the rare categories.

Fig. 3 shows the experimental results on the 20 Newsgroups collection. Differently from the experimental results on Reuters21578 collection, the performances for microF1 and macroF1 on this collection are very similar because the classes in this collection have almost the same number of training documents. Moreover, there is no significant improvement compared to the baseline performance. Unlike the experimental result on Reuters 21578, the performance with $\alpha$ being near 1.0 becomes lower than the baseline. We can observe that perfect per-document normalization (i.e., $\alpha = 0$) should be done to get a full benefit from the Poisson naive Bayes classifiers. One explanation for this result is that the number of training documents for each class is enough to train the traditional multinomial classifiers.

In order to further investigate, we conducted another experiment on the 20 Newsgroups collection, artificially varying the number of training documents and the results appear in Table 3. We see from this table that reducing the number of training documents makes the difference of the performances get larger. In addition to the macroF1 results on the Reuters21578 collection, this result also leads us to the conclusion that our per-document length normalization used in the proposed multivariate Poisson model for text classification is quite effective especially in the lack of training documents.

Table 4 summarizes the performances of the proposed classifiers, including BM25 and PLN.[6] In this table, the performances of RF and SRF indicate the ones of $\mathtt{RF}-\mathtt{l}$ and $\mathtt{SRF}-\mathtt{l}$ when $\alpha$ is 0.2. Unexpectedly, BM25 and PLN known as effective heuristics in ad hoc IR do not make further meaningful improvements compared to the RF and SRF. We also performed a microsign test suggested by [6] as a significant test. Although there is no significant performance variation among the normalization methods, all of these per-length normalization methods work significantly better than the traditional multinomial text classifier on the

[6] For BM25, we set $k1$ to 1.2 and $b$ to 0.5. For PLN, we set the pivot value to 0.75.

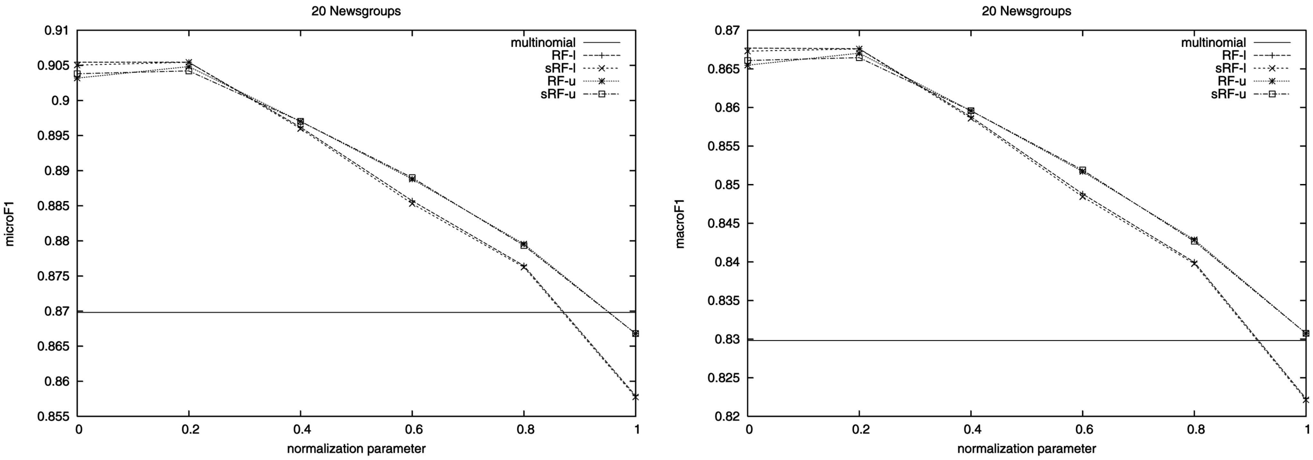

Fig. 3. MicroF1 and MacroF1 performances of per-document length normalization in the 20 Newsgroups collection.

TABLE 3
Effect of Per-Document Length Normalization in the 20 Newsgroup Collection with a Different Number of Training Documents

| # docs | 50 | 100 | 250 | 500 | 900 |
|---|---|---|---|---|---|
| multin. | 0.4655 | 0.5840 | 0.7221 | 0.8032 | 0.8698 |
| $\alpha = 0.2$, RF-l | 0.5804 | 0.6993 | 0.7998 | 0.8601 | 0.9026 |

TABLE 4
Performances with Various Length Normalization Methods

| | Reuters 21578 | | 20 Newsgroups | |
|---|---|---|---|---|
| | microF1 | macroF1 | microF1 | macroF1 |
| multin. | 0.7359 | 0.2238 | 0.8694 | 0.8298 |
| RF | 0.8029 | **0.5066**(+126.36) | 0.9054 | 0.8677 |
| SRF | **0.8035**(+9.19) | 0.5010 | 0.9054 | 0.8677 |
| BM25 | 0.7908 | 0.4838 | 0.8948 | 0.8568 |
| PLN | 0.7935 | 0.4795 | **0.9058**(+4.1) | **0.8679**(+4.59) |

TABLE 5
Comparison of the Performances of Different Weight-Enhancing Methods

| | Reuters 21578 | | 20 Newsgroups | |
|---|---|---|---|---|
| | microF1 | macroF1 | microF1 | macroF1 |
| PNB-best | 0.8035(SRF) | 0.5066(RF) | 0.9058(PLN) | 0.8679(PLN) |
| wSRF-IG | 0.8119 | 0.5815 | 0.9021 | 0.8981 |
| wSRF-CHI | 0.7889 | **0.6312** | 0.8704 | **0.9223** |
| wSRF-ExtRR | **0.8380** | 0.5973 | **0.9217** | 0.8830 |

Reuters 21578 collection ($p < 0.01$) and the 20 newsgroups collection ($p < 0.05$).

### 4.3 Effect of Feature Weights

To investigate the effect of feature weights, we have fixed the normalization parameter as 0.2 and compared the weight-enhanced Poisson classifiers compared to pure Poisson classifiers on the Reuters21578 and the 20 Newsgroups collections shown in Table 5. We obtain further improvements on both collections using feature weights. Especially, macroF1 performance of weight-enhanced Poisson classifier significantly outperforms the pure Poisson classifier (0.5066) as well as the baseline multinomial classifier (0.2238) in the experiments on the Reuteres21578 collection.

All three tested feature weighting measures perform well, but wSRF-IG and wSRF-CHI do not improve microF1 performance on the 20 Newsgroup collection and the Reuters collection, respectively. wSRF-IG and wSRF-CHI improved the performances on rare categories, but hurt the performances on large categories in the Reuters collection, resulting in disappointing results of microF1 performances. In addition, the proposed weight-enhancing method does not seem to contribute to improving performances in the experiments on the 20 Newsgroups collection. Even wSRF-ExtRR achieves 0.9217 of microF1 and 0.8830 of macroF1 on the 20 Newsgroups collection, which are not much different from the PLN Poisson classifier that achieves 0.9058 and 0.8679. Since the 20 Newsgroups

TABLE 6
Effect of Per-Document Length Normalization and Enhancing Classification Weights According to the Size of Categories on Reuters21578

| | Bin1 | Bin2 | Bin3 | Bin4 | Bin5 |
|---|---|---|---|---|---|
| # docs | 1-10 | 11-50 | 51-100 | 101-1000 | 1000- |
| # categories | 32 | 33 | 9 | 14 | 2 |
| multin. | 0.0051 | 0.1751 | 0.3994 | 0.6645 | **0.9632** |
| SRF | 0.1586 | 0.4147 | 0.5544 | 0.6838 | 0.9604 |
| wSRF-ExtRR | **0.4021** | **0.6019** | **0.6696** | **0.7812** | 0.9554 |

TABLE 7
Comparison of Performance: Multinomial, Bin-Optimized wSRF(wSRF$^*$), SVM on Reuters21578

| | microF1 | macroF1 | Speed(train/test) |
|---|---|---|---|
| multin. | 0.7359 | 0.2238 | 82s/35s |
| wSRF-ExtRR$^*$ | 0.8466(+15.04%) | 0.6002(+169.08%) | 90s/41s |
| SVM | 0.8589 | 0.6114 | 879s/106s |

collection has enough training documents for each class, these results show that the weight-enhancing method is mainly effective in the rare categories.

### 4.4 Relationship with Training Document Sizes

From the experimental results shown in Tables 4 and 5, it is possible to conclude that both the proposed Poisson classifier and the weight-enhancing method are effective, especially for the classes where the number of training documents is small. To investigate the relationship between the size of training documents and the effect of weight-enhancing method more clearly, we have grouped the categories in Reuters21578 into five bins according to the number of training documents for each class. We have chosen SRF and wSRF-ExtRR to compare because of their good performances in the previous experiments. Then, we have averaged the F1 performances for the classes in each bin by multinomial classifier, PNB with SRF and PNB with wSRF-ExtRR, as shown in Table 6.

In this table, we can observe that both SRF and wSRF-ExtRR considerably improve the baseline classifier in all bins except for Bin5, which contains the two most frequent categories, "earn" and "acq." The traditional multinomial classifier shows terribly poor performances in Bin1 and Bin2, while the baseline classifier competes with our proposed classifiers in Bin5. The SRF also shows poor performance in Bin1, but improves the baseline performances greatly in Bin2 and Bin3. In addition, we obtain more noticeable improvements by wSRF-ExtRR in Bin1 and Bin2 than those in Bin3 and Bin4.

This observation suggests that per-document length normalization requires an adequate amount of training documents. Our method estimates the parameters better than the multinomial classifier so that it works well in many classes, even where the multinomial classifier seriously drops performance because of its lack of training documents. However, the proposed classifier also shows poor performance in Bin1 or does not make a meaningful contribution in Bin5 where training documents are sufficient. The main reason for this limited improvement is that even the multinomial classifier can achieve very good performance by reliable parameter estimation using sufficient training documents. In addition, it is obvious that even per-document length normalization is not a solution to the lack of training data such as the case in Bin1.

On the other hand, it is surprising that the proposed classifier achieves relatively good performances by enhancing classification weights even for the classes where only a few training documents are available. Although the weight-enhancing method is somewhat crude from a theoretical point of view, this method seems to be quite effective by giving huge weight to a few words that occur in the positive training documents but not in the negative training documents. It is clear that a parameter estimation for a probabilistic model method does not work well if there are only a few training documents, and many experiments in the previous works also point out this problem [6]. We think that enhancing classification weights can be a good strategy for the naive Bayes to improve performance, although we need further analysis of experimental results and find some theoretical justification.

From the results, we know that weight-enhancement slightly degrades the performance for the categories where sufficient training documents are available, such as in Bin5. Table 7 compares the performances by multinomial, SVM, and Bin-optimized wSRF-ExtRR$^*$ classifiers, which adopt weight-enhancing methods for Bin1 to Bin4, but not for Bin5. The linear kernel function was used in the SVM experiment because we have found that there is no significant difference among the kernel functions in terms of classification performance. Our wSRF-ExtRR$^*$ can achieve high performance competing against a state-of-the-art text classification method such as SVM,[7] while requiring a tenth of the training time and a third of the test time of SVM. It means that naive Bayes can also achieve good performance in text classification domain by preserving its simple model architecture and computational efficiency.

[7] We have used $mSVM^{light}$ by Thorsten Joachims, which is available at http://svmlight.joachims.org/.

## 5 Conclusions and Future Works

In this paper, we propose a Poisson naive Bayes text classification model with weight-enhancing method. Our new model assumes that a document is generated by a multivariate Poisson model. We suggest per-document term frequency normalization to estimate the Poisson parameter, while the traditional multinomial classifier estimates its parameters by considering all the training documents as a unique huge training document. We also discuss enhancing classification weights for the rare categories where it is difficult to build a proper probabilistic classification model because of its lack of training documents. Instead, we strongly enhance the classification weights of a few informative words and make them play an important role to determine the label of the document.

Experimental results on both collections show that the proposed model is quite useful to build probabilistic text classifiers with little extra cost in terms of time and space, compared to the traditional multinomial classifiers. Relative frequency or smoothed relative frequency is enough to normalize term frequencies. However, there is no remarkable effect when the heuristic term frequency transforming method, such as BM25 and pivoted length normalization, is used. Enhancing classification weights is especially effective where the number of training documents is too small. We think that this method is a good alternative to traditional naive Bayes classifier, which necessarily performs poorly due to the lack of the training documents. Although our naive Bayes text classifier fails to outperform the state-of-the-art SVM classifier, we believe that the proposed classifier can be highly useful in a wide number of practical systems, such as a spam-filtering system or a adaptive news-alert system, because of its simplicity, efficiency, and guarantee for the incremental learning.

For future work, we will develop more elaborate per-document normalization techniques and a framework where our weight-enhancing method for rare categories is involved in a sound way. Furthermore, we will explore applications of our approach in other tasks such as adaptive filtering and relevance feedback.

## Acknowledgments

This work was partly supported by the JSPS Postdoctoral Fellowship Program and the Okumura Group at Tokyo Institute of Technology. H.-C. Rim was the corresponding author.

## References

[1] P. Domingos and M. J. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," *Machine Learning,* vol. 29, nos. 2/3, pp. 103-130, 1997.

[2] D.D. Lewis, "Representation and Learning in Information Retrieval," PhD dissertation, Dept. of Computer Science, Univ. of Massachusetts, Amherst, 1992.

[3] D.D. Lewis, "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," *Proc. ECML-98, 10th European Conf. Machine Learning,* C. Nédellec and C. Rouveirol, eds., pp. 4-15, 1998.

[4] A.K. McCallum and K. Nigam, "Employing EM in Pool-Based Active Learning for Text Classification," *Proc. ICML-98, 15th Int'l Conf. Machine Learning,* J.W. Shavlik, ed., pp. 350-358, 1998.

[5] S. Dumais, J. Plat, D. Heckerman, and M. Sahami, "Inductive Learning Algorithms and Representation for Text Categorization," *Proc. CIKM-98, Seventh ACM Int'l Conf. Information and Knowledge Management,* pp. 148-155, 1998.

[6] Y. Yang and X. Liu, "A Re-Examination of Text Categorization Methods," *Proc. SIGIR-99, 22nd ACM Int'l Conf. Research and Development in Information Retrieval,* M.A. Hearst, F. Gey, and R. Tong, eds., pp. 42-49, 1999.

[7] K. Nigam, A.K. McCallum, S. Thrun, and T.M. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM," *Machine Learning,* vol. 39, nos. 2/3, pp. 103-134, 2000.

[8] Y. Yang and C.G. Chute, "An Example-Based Mapping Method for Text Categorization and Retrieval," *ACM Trans. Information Systems,* vol. 12, no. 3, pp. 252-277, 1994.

[9] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Proc. ECML-98, 10th European Conf. Machine Learning,* C. Nédellec and C. Rouveirol, eds., pp. 137-142, 1998,

[10] R.E. Schapire and Y. Singer, "BOOSTEXTER: A Boosting-Based System for Text Categorization," *Machine Learning,* vol. 39, nos. 2/3, pp. 135-168, 2000.

[11] S.E. Robertson, S. Walker, and M. Hancock-Beaulieu, "Okapi at Trec-7: Automatic Ad Hoc, Filtering, VLC, and Interactive," *Proc. Text REtrieval Conf. (TREC),* pp. 199-210, 1998.

[12] A. Singhal, J. Choi, D. Hindle, D.D. Lewis, and F.C.N. Pereira, "AT & T at Trec-7," *Proc. Text REtrieval Conf. (TREC),* pp. 186-198, 1998.

[13] Y. Yang and J.O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," *Proc. ICML-97, 14th Int'l Conf. Machine Learning,* D.H. Fisher, ed., pp. 412-420. 1997.

[14] D. Koller and M. Sahami, "Toward Optimal Feature Selection," *Proc. Int'l Conf. Machine Learning,* pp. 284-292, 1996, citeseer.ist.psu.edu/koller96toward.html.

[15] T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," *Proc. ICML-97, 14th Int'l Conf. Machine Learning,* D.H. Fisher ed., pp. 143-151, 1997.

[16] J. Zhang and K.F. Yu, "What's the Relative Risk? A Method of Correcting the Odds Ratio in Cohort Studies of Common Outcomes," *J. Am. Medical Assoc.,* vol. 280, no. 19, pp. 1690-1691, 1998.

[17] D. Mladeni, "Feature Subset Selection in Text-Learning," *Proc. 10th European Conf. Machine Learning,* pp. 95-100, 1998.

[18] B.C. How and K. Narayanan, "An Empirical Study of Feature Selection for Text Categorization Based on Term Weightage," *Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence (WI '04),* pp. 599-602, 2004.

[19] F. Sebastiani, "Machine Learning in Automated Text Categorisation: A Survey," Technical Report IEI-B4-31-1999, Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT, revised version, 2001.

**Sang-Bum Kim** received the BS, MS, and PhD degrees from Korea University. He is now a visiting researcher at the Tokyo Institute of Technology invited and supported by the Japan Society of Promotion of Science since October 2005. He worked as a research assistant professor at the Research Institute of Information and Communication, Korea University. He has studied information retrieval and natural language processing and published a number of papers in the field. Recently, he has been studying sentiment processing of natural language text for text mining application.

**Kyoung-Soo Han** received the PhD degree in computer science from Korea University, Korea, in 2006. He is currently a research assistant professor at the Research Institute of Computer Information and Communication of Korea University. His research interests include question answering, dialogue modeling, information retrieval, text summarization, and machine learning.

**Hae-Chang Rim** received the BS degree in 1981 and the MS degree in computer science in 1983 from the University of Missouri-Columbia, respectively, and the PhD degree in computer science from the University of Texas at Austin in 1990. He is a professor in the Department of Computer Science and Engineering at Korea University, which he joined in 1991. He is currently an editorial staff member of the Asia Information Retrieval Symposium (AIRS) and an associate editor of *ACM Transactions on Asian Language Information Processing* (*TALIP*). Professor Rim was also an editorial staff member of the Association for Computational Linguistics (ACL) and the chief editor of the International Conference on Computer Processing of Oriental Languages (ICCPOL). He was the director of the Laboratory of Computer Science and Technology at Korea University, 1999-2000. His research interests are in several areas of natural language processing and information retrieval.

**Sung Hyon Myaeng** is currently a professor at the Information and Communications University (ICU), Korea, where he serves as the dean of the Academic Information Services Office. Prior to this appointment, he was a faculty member at Chungnam National University, Korea, and Syracuse University, New York, where he was granted tenure. He has served on the program committees of many international conferences in the areas of information retrieval, natural language processing, and digital libraries, including his role as a program cochair for ACM SIGIR, 2002. He is an editorial board member for *Information Processing and Management*, the *Journal of Natural Language Processing*, and the *Journal of Computer Processing of Oriental Languages* for which he is the information retrieval (IR) area chair. He was an associate editor for the *ACM Transactions on Asian Information Processing* from 1999 to 2003. He is currently the chair of SIG-HLT (Human Language Technology), Korea Information Science Society. He has published numerous technical articles about conceptual graph-based IR, cross-language IR, automatic summarization, text categorization, topic detection and tracking, and distributed. Recently, he has embarked on projects to use a common sense knowledge base for IR, text mining, and e-health applications.