Poisson Naive Bayes for Text Classification with Feature Weighting

Sang-Bum Kim, Hee-Cheol Seo and Hae-Chang Rim
Dept. of CSE., Korea University
5-ka Anamdong, SungPuk-ku, SEOUL 136-701, KOREA
{sbkim,hcseo,rim}@nlp.korea.ac.kr

Abstract

In this paper, we investigate the use of multivariate Poisson model and feature weighting to learn naive Bayes text classifier. Our new naive Bayes text classification model assumes that a document is generated by a multivariate Poisson model while the previous works consider a document as a vector of binary term features based on the presence or absence of each term. We also explore the use of feature weighting for the naive Bayes text classification rather than feature selection, which is a quite costly process when a small number of the new training documents are continuously provided.

Experimental results on the two test collections indicate that our new model with the proposed parameter estimation and the feature weighting technique leads to substantial improvements compared to the unigram language model classifiers that are known to outperform the original pure naive Bayes text classifiers.

1 Introduction

The naive Bayes classifier has been one of the core frameworks in the information retrieval research for many years. Recently, naive Bayes is emerged as a research topic itself because it sometimes achieves good performances on various tasks, compared to more complex learning algorithms, in spite of the wrong independence assumptions on naive Bayes.

Similarly, naive Bayes is also an attractive approach in the text classification task because it is simple enough to be practically implemented even with a great number of features. This simplicity enables us to integrate the text classification and filtering modules with the existing information retrieval systems easily. It is because that the frequency related information stored in the general text retrieval systems is all the required information in naive Bayes learning. No further complex generalization processes are required unlike the other machine learning methods such as SVM or boosting. Moreover, incremental adaptation using a small number of new training documents can be performed by just adding or updating frequencies.

Several earlier works have extensively studied the naive Bayes text classification (Lewis, 1992; Lewis, 1998; McCallum and Nigam, 1998). However, their pure naive Bayes classifiers considered a document as a binary feature vector, and so they can't utilize the term frequencies in a document, resulting the poor performances. For that reason, the unigram language model classifier (*or multinomial naive Bayes text classifier*) has been referred as an alternative and promising naive Bayes by a number of researchers(McCallum and Nigam, 1998; Dumais et al., 1998; Yang and Liu, 1999; Nigam et al., 2000). Although the unigram language model classifiers usually outperform the pure naive Bayes, they also have given the disappointing results compared to many other statistical learning methods such as nearest neighbor classifiers(Yang and Chute, 1994), support vector machines(Joachims, 1998), and boosting(Schapire and Singer, 2000), etc.

In the real world, an operational text classification system is usually placed in the environment where the amount of human-annotated training documents is small in spite of the hundreds of thousands classes. Moreover, re-training of the text classifiers is required since a small number of new training documents are continuously provided. In this environment, naive Bayes is probably the most appropriate model for the practical systems rather than other complex learning models. Therefore, more intensive studies about the naive Bayes text classification model are required.

In this paper, we revisit the naive Bayes framework, and propose a Poisson naive Bayes model for text classification with a statistical feature weighting method. Feature weighting has many advantages compared to the previous feature selection approaches, especially when the new training examples are continuously provided. Our new model assumes that a document is generated by a multivariate Poisson model, and their parameters are estimated by weighted averaging of the normalized and smoothed term frequencies over all the training documents. Under the assumption, we have tested the feature weighting approach with three measures: *information gain*, *χ^2-statistic*, and newly introduced *probability ratio*. With the proposed model and feature weighting techniques, we can get much better performance without losing the simplicity and efficiency of the naive Bayes model.

The remainder of this paper is organized as follows. The next section presents a naive Bayes framework for the text classification briefly. Section 3 describes our new naive Bayes model and the proposed technique, and the experimental results are presented in Section 4. Finally, we conclude the paper by suggesting possible directions for future work in Section 5.

2 Naive Bayes Text Classification

A naive Bayes classifier is a well-known and highly practical probabilistic classifier, and has been employed in many applications. It assumes that all attributes of the examples are independent of each other given the context of the class, that is, an independent assumption. Several studies show that naive Bayes performs surprisingly well in many domains(Domingos and Pazzani, 1997) in spite of its wrong independent assumption.

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

$$\begin{aligned} p(c|d_j) &= \frac{p(d_j|c)p(c)}{p(d_j)} \\ &= \frac{p(d_j|c)p(c)}{p(d_j|c)p(c) + p(d_j|\bar{c})p(\bar{c})} \\ &= \frac{\frac{p(d_j|c)}{p(d_j|\bar{c})} \cdot p(c)}{\frac{p(d_j|c)}{p(d_j|\bar{c})} \cdot p(c) + p(\bar{c})} \end{aligned} \tag{1}$$

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, Equation (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 Equation (3), we can get the posterior probability $p(c|d_j)$ by obtaining z_{jc}, which is a form of log ratio similar to the BIM retrieval model(Jones et al., 2000). It means that the linked independence assumption(Cooper et al., 1992), which explains that the strong independent assumption can be relaxed in the BIM model, is sufficient for the use of naive Bayes text classification model.

With this framework, two representative naive Bayes text classification approaches are well introduced in (McCallum and Nigam, 1998). They designated the pure naive Bayes as *multivariate Bernoulli model*, and the unigram language model classifier as *multinomial model*. Instead, we introduce *multivariate Poisson model* to improve the pure naive Bayes text classification in the next section.

3 Poisson Naive Bayes Text Classification

3.1 Overview

The Poisson distribution is most commonly used to model the number of random occurrences of some phenomenon in a specified unit of space or time, for example, the number of phone calls received by a telephone operator in a 10-minute period. If we think that the occurrence of each term is a random occurrence in a fixed unit of space (i.e. a length of document) the Poisson distribution is intuitively suitable to model the term frequencies in a given document. For that reason, the use of Poisson model is widely investigated in the IR literature, but it is rarely used for the text classification task(Lewis, 1998). It motivates us to adopt the Poisson model for learning the naive Bayes text classification.

Our model assumes that d_j is generated by multivariate Poisson model. In other words, a document d_j is a random vector which consists of the Poisson random variables X_i, and X_i has the value of within-term-frequency f_{ij} for the i-th term t_i. Thus, if we assume the independence among the terms in d_j, a probability of d_j is represented by,

$$p(d_j) = \prod_{i=1}^{|V|} P(X_i = f_{ij}) \tag{4}$$

where, $|V|$ is a vocabulary size, and each $P(X_i = f_{ij})$ is given by,

$$P(X_i = f_{ij}) = \frac{e^{-\lambda}\lambda^{f_{ij}}}{f_{ij}!} \tag{5}$$

where, λ is the *Poisson mean*.

As a result, the z_{jc} function of Equation (2) is rewritten using Equations (4) and (5) as follows:

$$\begin{aligned} z_{jc} &= \sum_{i=1}^{|V|} log \frac{P(X_i = f_{ij}|c)}{P(X_i = f_{ij}|\bar{c})} \\ &= \sum_{i=1}^{|V|} log \frac{e^{-\lambda_i}{\lambda_i}^{f_{ij}}}{e^{-\mu_i}{\mu_i}^{f_{ij}}} \end{aligned} \tag{6}$$

where, λ_i and μ_i is the *Poisson mean* for t_i in class c and class $\bar{c}$, respectively.

The most important issues of this work are as follows:

- How to decide the frequency f_{ij} representing the characteristic of document d_j?
- How to estimate the model parameter λ_i and μ_i representing the characteristic of each class?

We propose the possible answers in the next subsection.

3.2 Parameter Estimation

Since f_{ij} is a frequency of a term i in a document d_j with a *fixed length* according to the definition of Poisson distribution, we should normalize the actual term frequencies in the documents with the different length. In addition, many earlier works in NLP and IR fields recommend that smoothing term frequencies is necessary in order to build a more accurate model.

Thus, we estimate f_{ij} as the normalized and smoothed frequency of actual term frequency x_{ij}, represented by,

$$\tilde{f_{ij}} = \frac{x_{ij} + \theta}{dl_j + \theta \cdot |V|} \cdot \tau \tag{7}$$

where θ is a laplace smoothing parameter, τ is any huge value which makes all the $\tilde{f_{ij}}$ in our model an integer value[1], and dl_j is the length of d_j.

Conceptually, $\tilde{f_{ij}}$ can be regarded as the value estimated by the following steps : 1) Add θ of all $|V|$ terms to the document d_j, 2) Scale d_j up to d'_j whose total length is τ without changing the proportion of frequency for each term t_i, 3) Count t_i in d'_j.

Then, *Poisson mean* λ_i, which represents an average number of occurrence of t_i in the documents belonging to class c, is estimated using the normalized and smoothed $\tilde{f_{ij}}$ values over the training documents as follows:

$$\tilde{\lambda_i} = \sum_{d_j \in D_c} g(d_j|c) \cdot \tilde{f_{ij}} \tag{8}$$

where D_c is the set of training documents belonging to class c, and $g(d_j|c)$[2] is the interpolation of the uniform probability and the probability proportional to the length of the document, and the interpolation is calculated as follows:

$$g(d_j|c) = \alpha \frac{1}{|D_c|} + (1-\alpha)\frac{dl_j}{\sum_{d_j \in D_c} dl_j} \tag{9}$$

Simple averaging of $\tilde{f_{ij}}$, the case of α=1, seems to be correct to estimate λ_i. However, the statistics

[1] Since f_{ij} is a value of random variable X_{ij} representing the *frequency* in our Poisson distribution, we multiply the normalized frequency with some unnatural constant τ to make f_{ij} integer value. However, τ is dropped in the final induced function.

[2] We use the notation $g(d_j|c)$ for the distribution defined only in the training documents, to distinguish it from the notation $p(d_j|c)$ used in the Section 2.

from the long documents can be more reliable than those in the short documents, hence we try to interpolate between the two different probabilities with the parameter α ranging from 0 to 1. Consequently, λ_i is a weighted average over all training documents belonging to the class c, and μ_i for the class $\bar{c}$ can be estimated in the same manner.

3.3 Feature Weighting

Feature selection is often performed as a preprocessing step for the purpose of both reducing the feature space and improving the classification performance. Text classifiers are then trained with various machine learning algorithms in the resulting feature space. (Yang and Pedersen, 1997) investigated some measures to select useful term features including mutual information(MI), information gain(IG), and χ^2-statistics(CHI), etc. On the contrary, (Joachims, 1998) claimed that there is no useless term features, and it is preferable to use all term features. It is clear that learning and classification become very efficient when the feature space is considerably reduced. However, there is no definite conclusion about the contribution of feature selection to improve overall performances of the text classification systems. It may considerably depend on the employed learning algorithm. We believe that proper external feature selection or weighting is required to improve the performances of naive Bayes since the naive Bayes has no framework of the discriminative optimizing process in itself. Of the two possible approaches, feature selection is very inefficient in case that the additional training documents are provided continuously. It is because the feature set should be redefined according to the modified term statistics in the new training document set, and classifiers should be trained again with this new feature set. For that reason, we prefer to use feature weighting to improve naive Bayes rather than feature selection. With the feature weighting method, our z_{jc} is redefined as follows:

$$z_{jc} = \sum_{i=1}^{|V|} \frac{w_{ic}}{\mathrm{W}_c} \cdot log \frac{e^{-\tilde{\lambda_i}} \tilde{\lambda_i}^{\tilde{f_{ij}}}}{e^{-\tilde{\mu_i}} \tilde{\mu_i}^{\tilde{f_{ij}}}} \tag{10}$$

where, w_{ic} is the weight of feature for the class c, and W_c is the normalization factor, that is, $\sum_{i=1}^{V} w_{ic}$.

In our work, three measures are used to weight each term feature: *information gain*, *χ^2-statistics* and *probability ratio*. Information gain (or *average mutual information*) is an information-theoretic measure defined by the amount of reduced uncertainty given a piece of information. We use the information gain value as the weight of each term for the class c, and the value is calculated using a document event model as follows:

Table 1: Two-way contingency table

	presence of t_i	absence of t_i
labeled as c	a	b
not labeled as c	c	d

$$\begin{aligned} w_{ic} &= H(C) - H(C|W_i) \qquad (11) \\ &= \sum_{c_s \in \{c,\bar{c}\}} \sum_{w_t \in \{w_i,\bar{w}_i\}} p(c_s, w_s) \log \frac{p(c_s, w_t)}{p(c_s)p(w_t)} \end{aligned}$$

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.

Second measure we used is χ^2- statistics developed for the statistical test of the hypothesis. In the text classification, given a two-way contingency table for each term t_i and class c as represented in Table 1, w_{ic} is calculated as follows:

$$w_{ic} = \frac{(ad - bc)^2}{(a+b)(a+c)(b+d)(c+d)} \tag{12}$$

where, a,b,c and d indicate the number of documents for each cell in the above contingency table.

(Yang and Pedersen, 1997) compared the various feature selection methods, and concluded that these two measures are most effective for their kNN and LLSF classification models.

Finally, we introduce a new measure - probability ratio. Probability ratio is defined by,

$$w_{ic} = \frac{p(w_i|c)}{p(w_i|\bar{c})} + \frac{p(w_i|\bar{c})}{p(w_i|c)} \tag{13}$$

This measure calculates the sum of the ratio of two class-conditional probabilities from each class and its reciprocal. The former term and the latter term are representing the degree of predicting positive and negative class respectively. The weight using this measure always has a positive value higher than 2.

We have conducted the experiments with these three measures for the feature weighting test, and the results are given in Section 4.

3.4 Implementation Issues

By a couple of simple arithmetic operations, our final z_{jc} function can be rewritten as follows:

$$z_{jc} = \frac{\tau}{\mathrm{W}_c}(A_c + (B_c + \hat{z}_{jc})\frac{1}{dl_j{}'}) \quad (14)$$

where,

$$\begin{aligned} A_c &= \sum_{i=1}^{|V|} w_{ic}(\tilde{\mu_i}' - \tilde{\lambda_i}') \\ B_c &= \theta \sum_{i=1}^{|V|} w_{ic} \log \frac{\tilde{\lambda_i}'}{\tilde{\mu_i}'} \\ \hat{z}_{jc} &= \sum_{\forall i, t_i \in d_j} w_{ic} x_{ij} \log \frac{\tilde{\lambda_i}'}{\tilde{\mu_i}'} \\ \tilde{\lambda_i}' &= \frac{1}{\tau}\tilde{\lambda}_i = \sum_{d_j \in D_c} g(d_j|c) \cdot \frac{\tilde{f}_{ij}}{\tau} \\ dl'_j &= dl_j + \theta|V| \end{aligned}$$

In this equation, $\tilde{\lambda_i}'$ and $\tilde{\mu_i}'$ are just weighted average of τ-dropped $\tilde{f}_{ij}$, that is, $\frac{x_{ij}+\theta}{dl_j+\theta|V|}$. W_c, A_c and B_c are the class-specific constants, and τ is a constant over all the classes and documents. If the class c is fixed, W_c, A_c and τ can be dropped, and the ranking function z^*_{jc} is defined as follows:

$$z^*_{jc} = (B_c + \hat{z}_{jc})\frac{1}{dl_j{}'} \quad (15)$$

When we use this ranking function z^*_{jc}, the calculation of the exact posterior probability $p(c|d_j)$ presented in Section 2 becomes impossible. However, it is trivial since most of IR systems do not have interest on exact posterior probability. In addition, all the parameters in our model is guaranteed to be calculated by the incremental way.

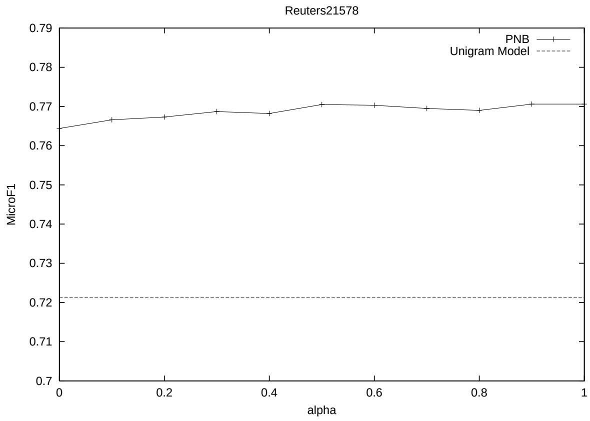

Figure 1: MicroF1 Performances for Reuters21578 according to interpolation parameter α for estimating λ and μ(without feature weighting)

4 Experimental Result

4.1 Data and Evaluation Measure

Our experiments were performed on the two datasets: Reuters21578 and KoreanNews2002 collection. Reuters21578 collection is the most widely used benchmark dataset for the text categorization research. We have used "ModApte" split version, which consists of 9603 training documents and 3299 test documents. There are 90 categories, and each document has one or more of the categories.

We have built another benchmark collection - KoreanNews2002 collection. KoreanNews2002 collection is composed of 15,000 news articles published during the year of 2002. The articles are collected from a number of Korean news portal websites, and each article is labeled with exactly one of the 46 classes. All the documents have date stamps attached and have been ordered according to their date stamps. With this date order, we divided them into the former 10,000 documents for training and the latter 5,000 documents for testing.

The performances are evaluated using popular F1 measure, and the F1 values for each class are micro-averaged(MicroF1) and macro-averaged(MacroF1) to examine the general classification performances.

4.2 Proposed Model : PNB (vs. UM)

Figure 1 shows the performances of our new model named Poisson naive Bayes(PNB) classifiers ac-

Table 2: Performances of UM and PNB on the Reuters21578 collection

	UM	PNB(min)	PNB(max)
MicroF1	0.7212	0.7644	0.7706
MacroF1	0.3214	0.4227	0.4358

Table 3: Performances of UM and PNB on the KoreanNews2002 collection

	UM	PNB(min)	PNB(max)
MicroF1	0.6502	0.7031	0.7094
MacroF1	0.5208	0.5859	0.5949

cording to the interpolation parameter α for estimating Poisson mean λ and μ. The baseline method is a unigram model classifier (UM) which is also referred to multinomial naive Bayes classifier described in (McCallum and Nigam, 1998). Our proposed PNB clearly outperforms the UM.

Although there is no significant difference of MicroF1 values among the various α values, the F1 value of each class is considerably affected by the α values. Figure 2 presents the fluctuations of the F1 values for 4 classes in Reuters21578 collection. From this result, we can assume that there is no global optimal value of α, but each class has its own optimal α. In our experiments, many of the classes have the highest F1 value when α is about 0.8 or 0.9 except some classes such as `corn` class which shows the highest F1 value at $\alpha = 0.3$. Similar results are obtained in the KoreanNews2002 collections.

Table 2 and 3 shows the MicroF1 and MacroF1 values of the unigram model classifiers and our PNB on the two collections, where PNB(min) and PNB(max) are the highest and lowest values at different α. In any cases, PNB is superior to UM.

4.3 Feature Weighting : PNB-{IG,CHI,PrR}

We have fixed the interpolation parameter α at 0.8, and evaluated the following feature weighting methods: PNB-IG with information gain, PNB-CHI with χ^2-statistic, and PNB-PrR with probability ratio. In these experiments, some important behaviors of feature weighted PNB classifiers are observed from the results. In order to explain the phenomenon, we have grouped the classes into the bins according to

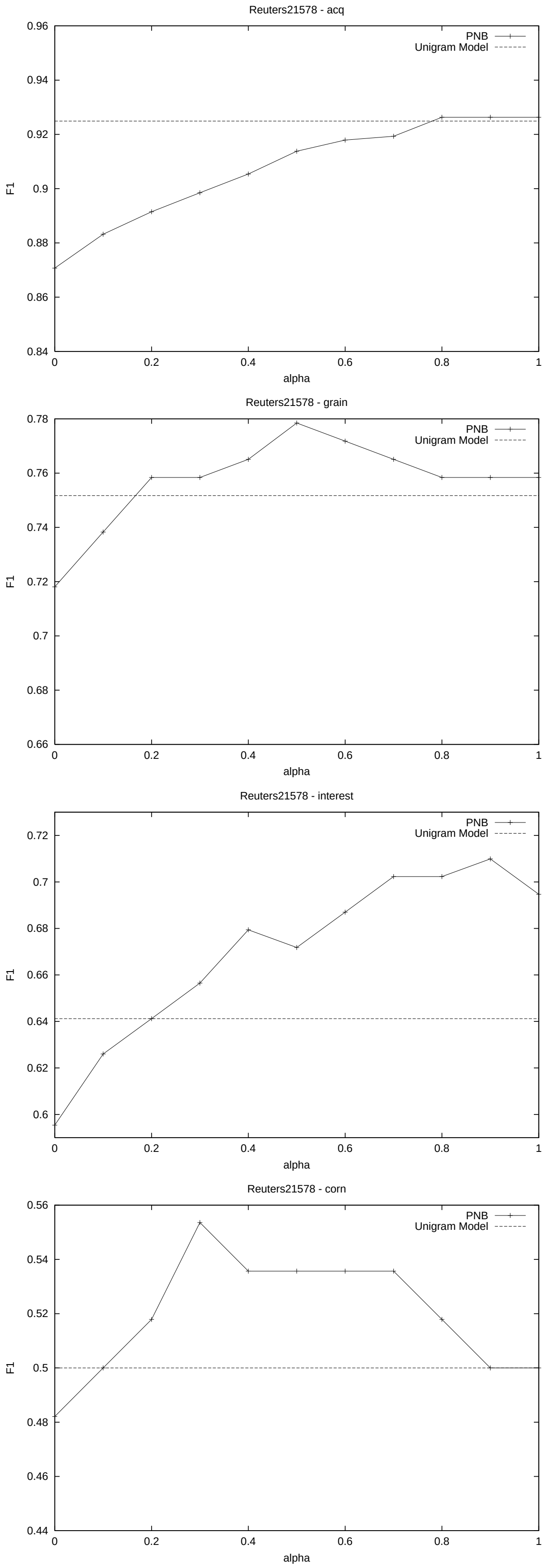

Figure 2: Performances for 4 categories in Reuters21578 according to interpolation parameter α for estimating λ and μ(without feature weighting)

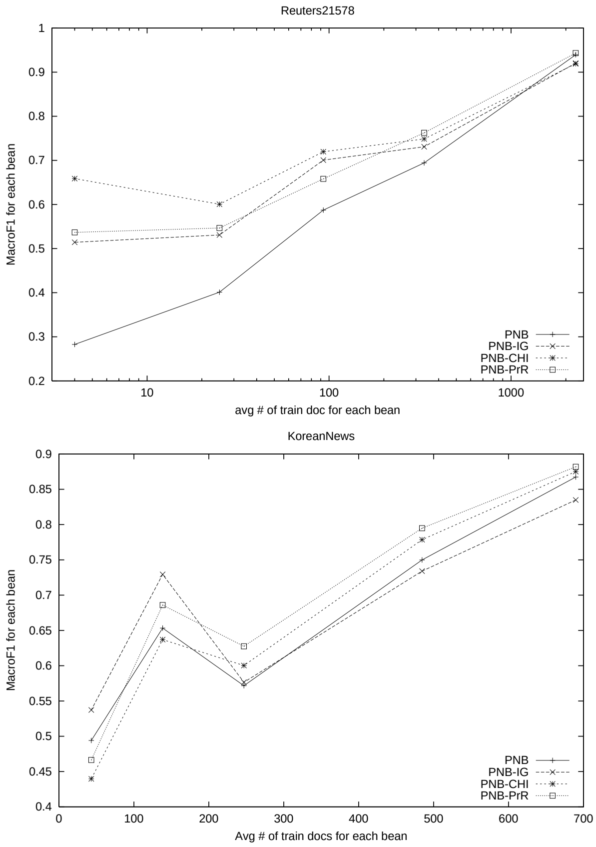

Figure 3: MacroF1 performances of the bins on Retures21578 and KoreanNews2002

the number of training documents for each class. 5 bins are generated in both Reuters21578 and KoreanNews2002 collection.

The different average F1 performance of each bin is shown in Figure 3. The clear observation from this result is that feature weighting is highly effective in the bins of the classes with a small number of training documents, but hardly contributes the performances for the bins of the classes with sufficiently many training documents. In the bins with enough training documents, simple PNB classifiers show the similar performances to the PNB with feature weighting methods. This tendency is more clearly captured in the Reuters21578 collection, where a third of the classes have fewer than 10 training documents. In contrast, two thirds of the classes in the KoreanNews2002 collection have more than a hundred of training documents.

Among the feature weighting methods, PNB-PrR performs stably than PNB-IG and PNB-CHI. PNB-IG or PNB-CHI somewhat degrades the performance in the classes with the large number of training documents, while PNB-PrR maintains the good performances in those classes on both of the collections. On the other hand, PNB-IG and PNB-CHI considerably improve the performances in the rare categories though the improvement is somewhat different from the two collections. For example, PNB-CHI significantly improves the simple PNB on the Reuters21578 collection while PNB-IG is very effective on the KoreanNews2002 collection. Thus, we can realize that the proper feature weighting method depends on the characteristics of the collection, and different feature weighting strategies should be adopted to improve the naive Bayes text classification.

From these observations, we tested another classifier PNB* which employ different feature weighting method for each bin to obtain the near optimal performances. Table 4 and 5 show the summary of the performances including PNB* on the both collections. Our proposed model with feature weighting methods are very effective compared to the baseline UM method. Moreover, the performance of bin-optimized PNB* in Reuters21578 collection shows that Poisson naive Bayes with feature weighting methods can achieve the state-of-the-art performances achieved by SVM or kNN which are reported in (Yang and Liu, 1999; Joachims, 1998).

5 Conclusion and Future Work

In this paper, we propose a Poisson naive Bayes text classification model with feature weighting. Our new model uses the normalized and smoothed term frequencies for each document, and Poisson parameters are calculated by weighted averaging the frequencies over all training documents. Experimental results show that the proposed model is quite useful to build probabilistic text classification systems without requiring any extra cost compared to the traditional simple naive Bayes or unigram language model classifiers.

Further improvement is achieved by a feature weighting technique. In our experiments, three measures including chi-square statistics, information gain, and newly introduced probability ratio are

Table 4: Summary of the performances on the Reuters21578 collection

	UM	PNB	PNB-IG	PNB-CHI	PNB-PrR	PNB*
MicroF1	0.7212	0.7690	0.7971	0.8167	**0.8190**(+13.56%)	0.8341
MacroF1	0.3414	0.4307	0.5800	**0.6601**(+93.35%)	0.5899	0.6645

Table 5: Summary of the performances on the KoreanNews2002 collection

	UM	PNB	PNB-IG	PNB-CHI	PNB-PrR	PNB*
MicroF1	0.6502	0.7056	0.7114	0.7122	**0.7409**(+13.95%)	0.7438
MacroF1	0.5208	0.5906	**0.6305**(+21.06%)	0.5748	0.6119	0.6662

adopted to weigh each term feature. The results show that feature weighting considerably improves the performances for the classes with a small number of training documents, but not for the classes with the sufficient training documents. Probability ratio also performs well, especially in the classes with the great number of training documents where other feature weighting methods show the unsatisfactory performances.

For the future work, we will try to develop some automatic methods of selecting proper feature weighting measures and determining the interpolation parameters for the different classes. Furthermore, we will explore applications of our approach in other tasks such as adaptive filtering and relevance feedback.

References

William S. Cooper, Fredric C. Gey, and Daniel P. Dabney. 1992. Probabilistic retrieval based on staged logsitic regression. *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 198–210.

Pedro Domingos and Michael J. Pazzani. 1997. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2/3):103–130.

Susan Dumais, John Plat, David Heckerman, and Mehran Sahami. 1998. Inductive learning algorithms and representation for text categorization. *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pages 148–155.

Thorsten Joachims. 1998. Text categorization with support vector machines: learning with many relevant features. *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142.

Karen Sparck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments - part 1. *Information Processing and Management*, 36(6):779–808.

David D. Lewis. 1992. *Representation and learning in information retrieval*. Ph.D. thesis, Department of Computer Science, University of Massachusetts, Amherst, US.

David D. Lewis. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 4–15.

Andrew K. McCallum and Kamal Nigam. 1998. Employing EM in pool-based active learning for text classification. *Proceedings of ICML-98, 15th International Conference on Machine Learning*, pages 350–358.

Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

Robert E. Schapire and Yoram Singer. 2000. BoosTexter: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Yiming Yang and Christopher G. Chute. 1994. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 12(3):252–277.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420.