A New Term Weighting Scheme for Text Classification using the Odds of Positive and Negative Class Probabilities

Youngjoong Ko
Computer Engineering, Dong-A University, Busan, 604-714 Korea
Youngjoong.ko@gmail.com

1. *Corresponding author* : Youngjoong Ko
2. *Corresponding address* : Department of Computer Engineering, Dong-A University, 840,
Hadan 2-dong, Saha-gu, Busan, 604-714, Korea
3. *Corresponding telephone number* : 82-51-200-7782
4. *Corresponding fax number* : 82-51-200-7783
5. *Corresponding Email address* : youngjoong.ko@gmail.com

Abstract

Text classification is a core technique for text mining and information retrieval. It has been applied to many applications in many different research and industrial areas. Term weighting schemes have to assign an appropriate weight to each term to obtain a high text classification performance. Although term weighting is one of the important modules for text classification, and text classification has different peculiarities from those in information retrieval, many term weighting schemes used in information retrieval, such as *tf.idf*, have also been utilized in text classification in the same manner. The peculiarity of text classification that differs most from information retrieval is the existence of class information. Therefore, this paper proposes a new term weighting scheme that utilizes class information using positive and negative class distributions. As a result, the proposed scheme, *log tf.TRR*, consistently performs better than other schemes using class information, as well as traditional schemes such as *tf.idf*.

Keywords: Term Weighting Scheme, Odds of Class Probabilities, Text Classification, Positive and Negative Classes

1. Introduction

Text Classification (TC) is the task of labeling natural language texts with thematic categories from a predefined set (Sebastiani, 2002). In text classification, text representation transforms the content of textual documents into a compact format allowing the documents to be recognized and classified using a classifier. In a vector space model, a document is represented as a vector in the term spaces, $d = (t_1, \dots, t_{|V|})$, where $|V|$ is the size of the vocabulary. The value of term t_i between $[0,1]$ represents how much t_i contributes to the semantics of document d . Text classification has borrowed traditional term weighting schemes from the field of information retrieval, such as *tf.idf* (Salton and Buckley, 1998) and its variants. In fact, the Probabilistic Ranking Principle (PRP) of information retrieval is based on the odds threshold, d is relevant iff $\frac{P(\text{relevant}|d,q)}{P(\text{nonrelevant}|d,q)} > 1$, where d is a document and q is a query (Manning et al., 2008). After Bayes rule and an independence assumption are applied to this equation for an odds threshold, the equation is decomposed into the odds of the term probabilities of t_i in document d (see section 3.2). In the case of information retrieval, these odds of the term probabilities cannot be applied to term weighting schemes directly because information retrieval systems cannot know which documents are relevant or irrelevant for a query in a practical sense. Thus, inverse document frequency (*idf*) weighting has been introduced to a term weighting scheme for information retrieval as *tf.idf*. However, training data for text classification contain both relevant and irrelevant information, which is generally annotated by hand. Therefore, this research starts with the question, “Can we develop a novel term-weighting scheme for specialized text classification better than those used in information retrieval?” We feel that text classification should utilize class information better than information retrieval because supervised-learning based text classification has the training data labeled as either positive or negative for each class. As mentioned above, inverse document frequency, *idf*, is simply a measure of the general importance of a term, and there is no use of class information. Therefore, this paper focuses on how text classification can be improved by effectively applying class information to a term weighting scheme. Since term weights used by the text classifier aim at distinguishing positive documents from negative documents, they are believed to be more effective by assigning more appropriate

weights using class information to the terms.

This paper proposes a novel term weighting scheme for multi-class text classification that uses the odds of the term probabilities estimated from positive and negative annotated data in each class. To estimate the term probabilities from positive and negative data, two different methods are employed: maximum likelihood estimation (MLE) for positive and negative data, and the expectation value of the MLE probabilities for each document in the data. The latter was developed to reflect the different term distributions in each document. Additional experiments were also conducted to resolve the problems incurred when test documents are represented; in the proposed term weighting scheme, although a test document does not have any class information, class information has to be utilized to represent a test document. As a result, the proposed scheme performs better than traditional term weighting schemes such as *tf.idf*, as well as other previous studies attempting to utilize class information on all datasets and classifiers.

The remainder of this paper is organized as follows. Section 2 presents other studies related to this work. Section 3 describes in detail the proposed term weighting scheme using class information. Section 4 presents the experiment design and results. Finally, Section 5 provides some concluding remarks regarding this research.

2. Related Work

TC is a supervised learning task because it makes use of the class information of training documents in contrast to information retrieval, and this known information has been widely used for the feature selection (Yang and Pedersen, 1997) and construction of text classifiers. Term weighting schemes using class information have been studied in several different ways. One is estimating term weights using feature selection metrics such as χ^2 , information gain, the gain ratio and the odds ratio (Debole and Sebastiani, 2003; Deng et al., 2004). Another approach is based on statistical confidence intervals that rely on prior knowledge of the statistical information in the labeled training data (Soucy and Mineau, 2005). Because these methods consider the document distribution, they are naturally expected to be superior to traditional term weighting methods. Comparisons with traditional term weighting methods have been conducted by Debole and Sebastiani (2003) and Soucy and Mineau (2005), but with mixed results. In our experiments,

a term weighting scheme with a χ^2 feature selection metric was re-implemented, and did not perform well.

Researchers have also recently attempted using class information for term weighting. There are two representative term weighting schemes: *Delta tf.idf* (Martineau and Finin, 2009; Paltoglou and Thelwall, 2010) and *rf* (*relevance frequency*) (Lan et al., 2006; Lan et al., 2009). The first, *Delta tf.idf*, provides a solution to the use of class information for sentiment classification by localizing the estimation of *idf* to documents of one or the other class, and subtracting the two values. However, this approach is limited to classification problems with only two classes, as in sentiment classification. The second, *rf*, was proposed to use the ratio of term occurrences of the positive and negative classes for calculating the term weights. The basic idea of this research is quite simple: the more concentrated a high frequency term is in the positive class than in the negative class, the more contributions it makes in discriminating positive samples from the negative samples (Lan et al., 2009). In this paper, we comprehensively compared my proposed schemes with their work (*rf* and *Delta tf.idf*), traditional *tf.idf* and a feature selection metric based term weighting scheme in the experiment section. In addition, the previous studies did not clearly explain how their schemes create a representation of test documents. If a test document does not have any prior class information, it becomes difficult to be represented using class information. In other words, whereas a traditional *tf.idf* scheme does not require any class information for creating a test document vector as an input for a text classifier, term weighting schemes that need to utilize class information for a vector representation, such as *rf*, *Delta tf.idf*, as well as the scheme proposed herein, may have several options for representing test documents using the expected class information. Thus, section 3.3 of this paper presents several methods for representing test documents as a vector using class information. The new term weighting scheme first appeared in (Ko, 2012). Whereas our previous work simply provided experimental evidence on the effectiveness of the newly proposed scheme, the current paper formally explains why this scheme was invented based on additional analyses, and adds additional data set and experiments including comparisons with a term weighting scheme using an χ^2 feature selection metric and another test document representation scheme using each class label as a hypothesis.

In addition, there have been several studies on solving binary classification problems for learning an unbalanced classifier, where the number of documents in each of the two classes,

positive and negative, is extremely unbalanced (Yu et al. 2003). Yu et al.'s method exploits the natural gap between distributions of positive and negative data, which also provides a chance to remove noisy data from both positive data and unlabeled data. And Han et al. (2007) reported that their method for removing noisy data is effective in a one-vs-rest setting for binary classification. Jiang and Zhai (2007) presented a domain adaptation approach to train statistical classifiers, from source domains where labeled examples are available, to a target domain where no labeled example is available. These studies showed us other alternatives about how to use positive and negative distributions in terms of the utilization of unlabeled data, removing noisy data and the adaptation of a new domain.

In particular, as big data analysis continues to be a major research trend, unsupervised or semi-supervised learning is playing an important role in text classification (Gliozzo et al., 2005; Ha-Thuc and Renders, 2011; Ko and Seo, 2009). Many new research issues remain for text classification. However, these issues are also based on the term weighting schemes of supervised-learning based text classification. Therefore, the current paper focuses on only term weighting schemes for using labeled data, and leaves other advanced research issues such as the utilization of the proposed term weighting scheme for unsupervised learning, noisy data removal, and domain adaptation as future work; these are significant issues for text classification and require a large number of experiments.

3. New Term Weighting Scheme

In this section, a brief review of traditional term weighting methods, i.e., *tf.idf* and its variants, is provided. In addition, the newly proposed term weighting scheme using the odds of positive and negative class probabilities are introduced. A review of the *rf* scheme and a comparison with the proposed scheme are given.

3.1. Brief review of traditional term weighting methods: tf.idf

There are three factors for term weighting in information retrieval (Salton and Buckley, 1988). Salton and Buckley (1998) considered three factors of assignment for an appropriately weighted single term in information retrieval. First, the term frequencies (*tf*) in a document appear to

represent closely the content of the document. Second, inverse document frequency (*idf*) has been proposed to increase a term's discriminating power to pick up all relevant documents from other irrelevant documents. These two factors are generally combined through a multiplication operation and are thought to improve the performance of information retrieval systems. Third, a cosine normalization factor is applied to equalize the length of the documents. Table 1 summarizes the components of *tf.idf* weighting methods.

Table 1. Components of *tf.idf*, where $tf_{t,d}$ is the frequency of term t in document d , df_t is the number of documents where t occurs, N is the total number of documents, and w_i is the weight of term i

Term frequency (<i>tf</i>)	Inverse document frequency (<i>idf</i>)	Normalization
n (natural) $tf_{t,d}$	t $\log \frac{N}{df_t}$	n (no normalization)
l (logarithm) $1 + \log(tf_{t,d})$		c (cosine) $\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_n^2}}$
a (augmented) $0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$		

The most popular way to combine a word's term frequency and inverse document frequency into a single weight is as follows (Manning and Schütze, 1999):

$$tf.idf_t = \begin{cases} (1 + \log(tf_t)) \log \frac{N}{df_t} & \text{if } tf_t \geq 1 \\ 0 & \text{if } tf_t < 1 \end{cases} \quad (1)$$

This equation is denoted by *ltc* in Table 1; its weight is normalized by the cosine normalization in *ltc*.

Although this *tf.idf* scheme was borrowed from information retrieval, it has also performed well in text classification. In this current study, two formats are used: *tf.idf* as a multiplication (*ntc*) of raw term frequency (*tf*) and $\log(N/df)$, denoted by *tf.idf*; and a multiplication (*ltc*) of logarithm term frequency ($\log tf$) and $\log(N/df)$, denoted by $\log tf.idf$. Because other variants share the same idea of *tf.idf*, and their formats are essentially similar, they are not used for the comparison of the proposed scheme described herein.

3.2. Probability Ranking Principle (PRP) in information retrieval

In information retrieval, the obvious order for presenting documents to the user is to rank them by their estimated probability of relevance with respect to the information need using a probabilistic model: $P(R = 1|d, q)$. This is the basis of the Probability Ranking Principle (PRP) (Gordon and Lenk, 1991, 1992; van Rijsbergen, 1979), in which all documents are simply ranked in decreasing order of $P(R = 1|d, q)$. Rather than estimating this probability directly, we work with some other quantities that are easier to compute and give the same document ordering. In particular, we can rank documents based on their odds of relevance because the odds of relevance are monotonic with the probability of relevance. This makes things easier because we can ignore a common denominator as follows:

$$O(R|\vec{d}, \vec{q}) = \frac{P(R=1|\vec{d}, \vec{q})}{P(R=0|\vec{d}, \vec{q})} = \frac{\frac{P(R=1|\vec{q})P(\vec{d}|R=1, \vec{q})}{P(\vec{d}|\vec{q})}}{\frac{P(R=0|\vec{q})P(\vec{d}|R=0, \vec{q})}{P(\vec{d}|\vec{q})}} = \frac{P(R=1|\vec{q})}{P(R=0|\vec{q})} \cdot \frac{P(\vec{d}|R=1, \vec{q})}{P(\vec{d}|R=0, \vec{q})} \quad (2)$$

The left-hand term, $\frac{P(R=1|\vec{q})}{P(R=0|\vec{q})}$, which is the rightmost expression of equation (2), is a constant for a given query, and does not need to be estimated when ranking documents. After the Naïve Bayes conditional independence assumption is applied to the right-hand term, $\frac{P(\vec{d}|R=1, \vec{q})}{P(\vec{d}|R=0, \vec{q})}$, we can obtain the following equation (3):

$$\frac{P(\vec{d}|R=1, \vec{q})}{P(\vec{d}|R=0, \vec{q})} = \prod_i^{|V|} \frac{P(t_i|R=1, \vec{q})}{P(t_i|R=0, \vec{q})}, \quad (3)$$

where $\vec{d} = [t_1, \dots, t_{|V|}]$, $|V|$ is the size of the vocabulary.

We can now define the Relevance Status Value (RSV) of document, d , and class, cl , as $RSV(d, c)$, for text classification using PRP and equation (3). In $RSV(d, c)$ for text classification, a query is replaced by a class label. That is, we want to estimate how relevant a document is to a given class to determine whether the document should be assigned to that class.

$$RSV(d, c) = \log \left(\frac{P(\vec{d}|R=1, cl)}{P(\vec{d}|R=0, cl)} \right) = \log \prod_i^{|V|} \frac{P(t_i|R=1, cl)}{P(t_i|R=0, cl)} \quad (4)$$

In multiclass classification, a single text classifier is generally trained per class to distinguish that class (positive class) from all other classes (negative class); this strategy is called one-vs-all (OvA) or one-vs-rest (OvR) (Aphinyanaphongs et al. 2014; Galar et al., 2011; Han et al., 2007). In this case, a positive class is denoted by cl , and a negative class is denoted by $\bar{cl}$. Using these notations, equation (4) can be reconstructed as follows:

$$RSV(d, c) = \log \left(\frac{P(\bar{d}|cl)}{P(\bar{d}|\bar{cl})} \right) = \log \prod_i^{|V|} \frac{P(t_i|cl)}{P(t_i|\bar{cl})} = \sum_i^{|V|} \log \frac{P(t_i|cl)}{P(t_i|\bar{cl})}. \quad (5)$$

The rightmost expression, $\log \frac{P(t_i|cl)}{P(t_i|\bar{cl})}$, for a single term, t_i , is called the Term Relevance Ratio (TRR), and the *idf* part of *tf.idf* is replaced with the TRR in the proposed term weighting scheme. $P(t_i|cl)$ and $P(t_i|\bar{cl})$ of the TRR can be estimated through the following two ways. The first way is based on maximum likelihood estimation (MLE), as shown in equation (6),

$$P(t_i|cl) = \frac{\sum_{j=1}^{|T_{cl}|} tf_{t_{ij}}}{\sum_{k=1}^{|V|} \sum_{j=1}^{|T_{cl}|} tf_{t_{kj}}}, \quad P(t_i|\bar{cl}) = \frac{\sum_{j=1}^{|T_{\bar{cl}}|} tf_{t_{ij}}}{\sum_{k=1}^{|V|} \sum_{j=1}^{|T_{\bar{cl}}|} tf_{t_{kj}}}, \quad (6)$$

where V denotes the vocabulary set of all training data, and T_{cl} is the document set of the positive class, cl , and $T_{\bar{cl}}$ is the document set of the negative class, $\bar{cl}$.

This type of estimation misses the term distribution in a document. Thus, the second estimation method is a reformation of the first method as follows:

$$P(t_i|cl) = \sum_{k=1}^{|T_{cl}|} P(t_i|d_k) \cdot P(d_k|cl), \quad (7)$$

$$P(t_i|\bar{cl}) = \sum_{k=1}^{|T_{\bar{cl}}|} P(t_i|d_k) \cdot P(d_k|\bar{cl}), \quad (8)$$

where $P(t_i|d_k)$ is estimated through MLE as in equation (6), and $P(d_k|cl)$ and $P(d_k|\bar{cl})$ are estimated as having a uniform distribution.

The final term weighting scheme using the TRR uses the following equation.

$$\log \text{tf. TRR} = (\log \text{tf}_{t_i} + 1) \cdot \log \frac{P(t_i|cl)}{P(t_i|\bar{cl})} \quad (9)$$

3.3. Test document representation method using the TRR

The next issue is how to represent the test documents. A document representation method needs to be developed because the test documents do not have any class information, which the proposed term weighting scheme requires. A test document can first be represented as $|C|$ different vectors using the TRR of each class, which then has to be represented as one vector that describes well the document in the proposed vector space, where $|C|$ is the number of classes. Three solutions for using a single vector representation are listed in Table 2, along with a fourth solution in which the class label of a test document is regarded as a hypothesis. That is, a single vector is not created for each test document; instead, all of the different $|C|$ vectors created from the TRR of each class are used because a test document can have any class label. Section 4.3 describes the experimental evidence for the use of such test document representation methods.

Table 2. Four representation methods for a test document

<i>W-Max</i> (Word Max)	The term weight of each word is chosen based on the maximum value among $ C $ estimated term weights.
<i>D-Max</i> (Document Max)	The sum of all term weights in each vector is first calculated, and one vector with the maximum sum value is then selected as a representative vector.
<i>D-TMax</i> (Documents Two Max)	The sum of all term weights in each vector is calculated, and two vectors with the highest and second highest sum values are then selected. A vector is then created by choosing the term weight with the higher score between the two term weights of the selected vectors for each term.
<i>Hypo</i> (Hypothesis)	Each class label of the test document is considered as a hypothesis. That is, we assume that a test document can have any class label. Then, $ C $ vectors are generated with the term weights of each class, and a text classifier is used to calculate the prediction scores for all of them. Eventually, the class label with the highest prediction score is selected as the predicted class label.

3.4. Term discrimination power analysis through a comparison with the relevance frequency

Lan et al. (2009) proposed a new supervised weight factor relevance frequency (*rf*), the equation

of which can be expressed as follows:

$$rf = \log(2 + \frac{a}{c}). \quad (10)$$

The *idf* part of *tf.idf* is also replaced by *rf*, and the weight of term, t_i , is defined as follows:

$$tf.rf = tf_{t_i} \cdot \log(2 + \frac{a}{c}). \quad (11)$$

In equations (10) and (11), a and c denote the number of documents in different cases, *positive* (a) and *negative* (c), in OvA. Lan et al. (2009) detailed a , b , c and d descriptively using Figure 1, which is provided to fully compare and analyze *rf* against a traditional term weighting scheme (*tf.idf*) and feature selection methods (χ^2 , information gain, gain ratio and odds ratio).

Figure 1. Examples of a , b , c , and d values in different distributions of positive and negative classes in a whole collection

- a is the number of documents in the positive class that contain this term

- b is the number of documents in the positive class that do not contain this term
- c is the number of documents in the negative class that contain this term
- d is the number of documents in the negative class that do not contain this term
- N is the number of documents in the whole collection, and is the sum of a , b , c and d .

Using these notations, we can represent the *idf* factor as follows:

$$\text{idf} = \log \frac{N}{a+c}. \quad (12)$$

Because Lan et al. provided some evidence on why their new *rf* method is superior to both feature selection and *if.idf* methods, we conducted a comparison analysis of the proposed and *rf* schemes herein, as shown in Figure 1. In summary, d is generally much larger than a , b and c in most cases, as Figure 1 illustrates. This fact lessens the discrimination power of feature selection methods, all of which use the d factor in their equations. Both the proposed term weighting scheme (TRR) and *rf* do not use the d factor, but the weakness of the *rf* method as related to the normalization problem should be pointed out. First, the distribution of the number of documents over classes is in practice uneven in most cases. Thus, the ratio of document numbers of positive and negative classes under the OvA method for each class may vary, and some normalization methods should be applied to term weighting schemes in common application areas. As can be seen in Figure 1, t_1 , t_2 and t_3 have the same *rf* values, although they all have different distributions of positive and negative classes. Second, although the numbers of documents in each class are even, the number of documents in the negative class can also vary greatly according to the total number of classes under the OvA method. This affects the c values of the *rf* equation, which create the different values of the *rf* part of the term weighting according to the data size. Eventually, this can create a bias in the final *tf.rf* weights, i.e., if the *rf* value is small, the *tf* value contributes more to the *tf.rf* weight; otherwise, the *rf* value contributes more to the *tf.rf* weight. This means the *tf.rf* weight scheme can have a different discrimination power on different sized datasets. In practice, we observed the a , c and a/c values of several terms in Reuters datasets with different sizes (10 and 90 classes). As can be seen in Table 3, most of the terms naturally have

larger c values in the Reuters dataset with 90 classes, which diminishes the effect of the rf part of the term weighting in comparison with tf .

Table 3. Comparison of the a , c and a/c values of the four terms in the *acq* and *earn* classes on different sizes of Reuters datasets

class	term	Reuters datasets with 10 classes			Reuters datasets with 90 classes		
		a	c	a/c	a	c	a/c
acq	assets	189	154	1.23	189	202	0.94
	commission	206	162	1.27	206	297	0.69
earn	gains	123	55	2.24	123	131	0.94
	costs	150	119	1.26	150	255	0.59

In addition, the a , b , c and d notations by Lan et al. (2009) in Figure 1 are calculated not by the term frequency but by the document frequency. This can generate biased term weights particularly when the number of documents in a class is very small. Thus, we estimate the probabilities using the term frequency in positive/negative classes instead of the document frequency. In particular, if a , b , c and d notations are used, our estimation equation (TRR) can be approximated through equation (13), and can be compared using equation (6).

$$P(t_i|cl) = \frac{a}{a+b}, \quad P(t_i|\bar{cl}) = \frac{c}{c+d} \quad (13)$$

4. Experiments

We tested the proposed term weighting scheme on two widely used datasets, Reuters 21578 and 20 Newsgroups, as well as the Korean UseNet dataset. In this section, we present the results from several experiments demonstrating the effectiveness of our proposed technique and summarize the results.

4.1. Test collection and experimental settings

To test the proposed term weighting scheme, we used two newsgroup datasets, 20 Newsgroups and the Korean UseNet dataset, written in two different languages, English and Korean, and the Reuters 21578 dataset.

First, two widely used datasets were used as the benchmark as follows:

The Reuters 21578 Distribution 1.0 dataset (*Reuters*) consists of 12,902 articles and 90 topic categories from the Reuters newswire (Aphinyanaphongs et al. 2014; Debole and Sebastiani, 2003; Gliozzo et al., 2005; Ke, 2012; Sun et al. 2003; Sun et al., 2009; Yang and Liu, 1999; Yu et al., 2003). Following other studies by Nigam (2001) and Joachims (1998), we built binary classifiers for each class to identify the news topic. Since the documents in this dataset can have multiple class labels, each class is traditionally evaluated using a binary classifier. To split the training/test data, we follow a standard *ModApte* split (Lewis 1997). The standard *ModApte* training/test split divides the articles by time, and the top-ten largest classes were used in the experiments. All words inside the title and body were used, along with a stop-list and no stemming.

The Newsgroups dataset (*NG*), collected by Ken Lang, contains about 20,000 documents evenly divided among 20 UseNet discussion groups (Banerjee and Basu, 2007; Gliozzo et al., 2005; Ke, 2012; McCallum et al., 1998; Sun et al., 2009; Yoon et al. 2006). For a fair evaluation, we evaluated our scheme using the five-fold cross-validation method. All results of the experiments using this dataset are the averages of five runs. After removing words that occur only once or are on a stop-word list, the vocabulary from the training data included 51,018 words (with no stemming).

The second dataset (*KNG*) was gathered from the Korean UseNet group (Ko et al., 2004). This dataset contains a total of 10,331 documents and 15 categories. In total, 3,107 documents (30%) were used for test data, and the remaining 7,224 documents (70%) were used for the training data. The resulting vocabulary from the training data has 69,793 words. This dataset is uneven, as shown in Table 4.

All parameter values of our experiments are from a validation set, which was randomly chosen to be 20% of the training data from each dataset.

Table 4. Constitution of Korean newsgroup dataset (*KNG*)

Class	Training data	Test data	Total
han.arts.music	315	136	451
han.comp.database	198	86	284
han.comp.devtools	404	174	578
han.comp.lang	1,387	595	1,982
han.comp.os.linux	1,175	504	1,679

han.comp.os.window	517	222	739
han.comp.sys	304	131	435
han.politics	1,469	630	2,099
han.rec.cars	291	126	417
han.rec.games	261	112	373
han.rec.movie	202	88	290
han.rec.sports	130	56	186
han.rec.travel	102	45	147
han.sci	333	143	476
han.soc.religion	136	59	195
Total	7,224	3,107	10,331

These datasets have very different characteristics. *Reuters* has a skewed class distribution, and many of the documents have two or more class labels, whereas *NG* has a uniform class distribution, and its documents have only one class label. On the other hand, *KNG* does not have a skewed class distribution but rather a non-uniform one, and its documents have only one class label written in another language, Korean. Thus, two different measures were used to evaluate various term weighting schemes on the two classifiers and three different datasets for our experiments. For *Reuters*, we used the Break Even Point (*BEP*) measure, which is a standard information retrieval measure for binary classification, and for *NG* and *KNG*, the performances are reported using an F_1 measure. Herein, we followed the standard definition of recall (r), precision (p), and F_1 measure ($2rp/(r+p)$). To evaluate the average performance across categories, we used the *micro-averaging* and *macro-averaging* methods (Ko and Seo. 2009; Yang, 1999). The recall, precision, and F_1 measure can first be computed for individual categories, and then averaged over the categories as a global measure of the average performance; this method of averaging is called *macro-averaging*. An alternative method, *micro-averaging*, is used to count the decisions for all categories in a joint pool and compute the global recall, precision, and F_1 values for that global pool. Herein, we report the statistical significance for the observed differences in F_1 values of category pairs for *Reuters* and *KNG* and fold pairs for *NG* when the p value is sufficiently small ($p < 0.01$ or $p < 0.05$) based on a one-sided paired t -test¹ (Smucker et al., 2007; Yang and Liu, 1999).

¹ For *Reuters* and *KNG*, we tested statistical significance on differences in F_1 values of category pairs by Macro t -test (Yang and Liu 1999). Because the evaluation on *NG* is based on the five-fold cross validation, we did it on differences in F_1 values of fold pairs. In particular, we obtained the same results by Macro t -test on *NG*.

We used χ^2 statistics for the statistical feature selection (Yang et al., 1997; Aggarwal and Zhai, 2012). This metric was also used for a term weighting scheme (*log tf.chi*) instead of *idf* (Debole and Sebastiani, 2003). In our study, it was chosen as one of the other term weighting schemes for a comparison with our own. The χ^2 statistics values are calculated through the following equation.

$$\chi^2(t_k, cl_i) = \frac{[P(t_k, cl_i)P(\bar{t}_k, \bar{cl}_i) - P(t_k, \bar{cl}_i)P(\bar{t}_k, cl_i)]^2}{P(t_k)P(\bar{t}_k)P(cl_i)P(\bar{cl}_i)}, \quad (14)$$

where $P(\bar{t}_k, cl_i)$ indicates the probability that, for a random document x , term t_k does not occur in x , which belongs to class cl_i , and is estimated based on the maximum likelihood. As a globalization function, $f_{\max}(t_k) = \max_{i=1}^{|C|} f(t_k, cl_i)$ is selected for the feature selection and term weighting.

To evaluate our term weighting scheme, two promising learning algorithms, *kNN* and *SVM*, which have shown a better performance than the other algorithms, were chosen for the experiments. The value of k in *kNN* was set to 30 based on a previous parameter optimization learned using the validation set. For *SVM*, we used the linear models offered by *SVM^{light}* (Joachims, 1998).

In our experiments, all of the term weighting schemes using class information (*log tf.chi*, *log tf.rf*, *Delta tf.idf* and *log tf.TRR*), which replace the *idf* part of *tf.idf* in equation (1) with *chi*, *rf*, *Delta idf* and *TRR*, were applied to generate document vectors and normalize them using *ltc* in Table 1. Feature selection was then employed to observe the performance changes of each scheme according to the number of features based on the ordering from the χ^2 statistics. Here, all of the features were a unigram of terms, and the document representation was a bag-of-words representation. The document vectors created through this process were equally used for *kNN* and *SVM*.

4.2. Comparison of the proposed schemes, *idf*, *chi* and *rf*

First, the proposed term weighting schemes using *TRR* are compared with the traditional *idf*, *chi* and *rf* schemes. Here, none of these schemes used any *tf* information. As a result, two versions of the proposed scheme, *Cat-MLE* and *Doc-MLE*, achieved better performances than the *idf* and *rf*

schemes. Here, the results of TRR need to be discussed because TRR has two different estimation methods (*Cat-MLE* by equation (6), and *Doc-MLE* by equations (7) and (8)), which show different performance aspects in each dataset; *Cat-MLE* obtained the best performances for *Reuters*, while *Doc-MLE* obtained the best performances for *NG* and *KNG*. This may be caused by the skewed distribution of the document length in *Reuters*; many documents in *Reuters* consist of small numbers of sentences, such as two or three, which creates poor probability estimation for *Doc-MLE*. As can be seen in Figure 2, TRR achieved better performances than the other schemes over all intervals of the number of features, from 1,000 to 10,000. The number of features in this experiment was limited by the ranking from the feature selection, i.e., χ^2 statistics.

Table 5. Comparison of TRR (Cat-MLE and Doc-MLE), *idf*, and *rf*

		<i>idf</i>	<i>chi</i>	<i>rf</i>	<i>Cat-MLE</i>	<i>Doc-MLE</i>
<i>Reuters</i> (<i>BEP</i>)	<i>kNN</i>	92.86	86.7	93.29	94.22	93.93
	SVM	94.11	90.03	94.47	94.69	94.29
<i>NG</i> (<i>F₁</i>)	<i>kNN</i>	86.39	69.95	85.75	87.59	88.43
	SVM	87.61	80.45	87.69	87.87	88.34
<i>KNG</i> (<i>F₁</i>)	<i>kNN</i>	82.91	65.32	76.83	83.98	84.33
	SVM	85.66	75.87	82.09	85.02	85.66

(a) Changes in performance according to the number of features in Reuters using *kNN*

(b) Changes in performance according to the number of features in Reuters using SVM

(c) Changes in performance according to the number of features in NG using kNN

(d) Changes in performance according to the number of features in NG using SVM

(e) Changes in performance according to the number of features in KNG using kNN

(f) Changes in performance according to the number of features in KNG using SVM

Figure 2. Comparison of *TRR* (Cat-MLE and Doc-MLE), *idf* and *rf* based on changes in performance according to the number of features

4.3. Document representation methods applied to test data

Table 6 shows the performances of four different document representation methods for the test documents: *Hypo*, *W-Max*, *D-Max* and *D-TMax*. *D-TMax* achieved the best performance for *Reuters*, whereas *D-Max* achieved the best performance for *NG* and *KNG*. These results seem natural because many of the documents in *Reuters* have two or more labels. For example, if a document with two labels is represented using only one class distribution between two labels, the classifiers may have some difficulty classifying it in the other class.

Table 6. Comparison of document representation schemes for the test data

			<i>Hypo</i>	<i>W-Max</i>	<i>D-Max</i>	<i>D-Tmax</i>
<i>Reuters</i> (<i>BEP</i>)	<i>kNN</i>	<i>Cat-MLE</i>	92.96	94.33	94.51	<u>94.90</u>
		<i>Doc-MLE</i>	92.86	94.54	93.97	<u>94.72</u>
	<i>SVM</i>	<i>Cat-MLE</i>	92.79	94.83	94.83	<u>95.30</u>
		<i>Doc-MLE</i>	93.18	94.72	94.65	<u>95.12</u>
<i>NG</i> (<i>F₁</i>)	<i>kNN</i>	<i>Cat-MLE</i>	70.26	84.19	<u>87.15</u>	86.01
		<i>Doc-MLE</i>	68.68	84.30	<u>87.75</u>	86.36
	<i>SVM</i>	<i>Cat-MLE</i>	83.75	86.60	<u>87.93</u>	87.54
		<i>Doc-MLE</i>	83.71	86.24	<u>88.31</u>	87.75
<i>KNG</i> (<i>F₁</i>)	<i>kNN</i>	<i>Cat-MLE</i>	66.92	80.01	<u>84.26</u>	82.69
		<i>Doc-MLE</i>	70.17	80.47	<u>84.36</u>	82.85
	<i>SVM</i>	<i>Cat-MLE</i>	80.16	83.99	<u>85.18</u>	84.76
		<i>Doc-MLE</i>	80.61	84.12	<u>85.70</u>	85.18

4.4. Comparison of the proposed and other term weighting schemes

Here, we add another benchmark term weighting scheme, *Delta tf.idf* (Martineau and Finin, 2009; Paltoglou and Thelwall, 2010). This term weighting is calculated through the following equation.

$$\text{Delta tf.idf} = tf_t \cdot \left\{ \log \frac{N_{cl}}{df_{t,cl}} - \log \frac{N_{\bar{cl}}}{df_{t,\bar{cl}}} \right\}, \quad (15)$$

where N_{cl} and $N_{\bar{cl}}$ are the total number of training documents in the positive class, cl , and negative class, $\bar{cl}$, respectively, and $df_{t,cl}$ and $df_{t,\bar{cl}}$ are the numbers of training documents in positive and negative classes respectively.

Tables 7, 8 and 9 show the final experiment results, a comparison of the proposed and other schemes in each class, and the overall performances after micro-averaging and macro-averaging. The proposed scheme achieved a better performance than the other schemes. In particular, the proposed method showed the best performance for most categories of all three datasets. Figure 3 shows the performance changes based on the number of features, from 1,000 to 10,000. The proposed *log tf.TRR* method achieved the best performance for all of the datasets and classifiers in each interval. Note that *tf.idf* used the raw term frequency, whereas *log tf.idf* used $\log(tf)+1$, as in equation (1).

Table 7. Performance comparison on the Reuters dataset

Category (# of training documents)	<i>kNN (BEP)</i>						<i>SVM (BEP)</i>					
	<i>tf.idf</i>	<i>log</i>	<i>log</i>	<i>tf.rf</i>	<i>Delta</i>	<i>log</i>	<i>tf.idf</i>	<i>log</i>	<i>log</i>	<i>tf.rf</i>	<i>Delta</i>	<i>log</i>
		<i>tf.idf</i>	<i>tf.chi</i>		<i>tf.idf</i>	<i>tf.TRR</i>		<i>tf.idf</i>	<i>tf.chi</i>		<i>tf.idf</i>	<i>tf.TRR</i>
Acq (1650)	95.27	96.66	90.54	96.1	95.27	97.07	97.91	97.49	95.68	97.77	96.24	97.35
Corn (181)	80.35	82.14	69.64	78.57	85.71	85.71	85.71	87.5	94.46	83.92	91.07	92.85
Crude (389)	88.88	88.35	87.83	90.47	87.3	92.06	92.06	91	90.47	92.06	82.01	91.53
Earn (2877)	97.88	97.97	96.5	97.88	97.05	98.43	98.62	98.62	98.16	98.06	98.16	98.71
Grain (433)	91.27	93.95	89.93	83.89	88.59	94.63	95.3	96.64	93.95	95.3	95.3	97.31
Interest (347)	81.67	82.44	75.57	81.67	80.15	87.78	82.44	82.44	79.38	84.73	72.51	83.96
Money-fx (538)	82.12	84.91	79.88	87.15	84.91	87.15	85.47	83.79	82.12	86.03	82.12	85.47
Ship (197)	82.02	82.02	77.52	85.39	86.51	87.64	87.64	91.01	88.76	89.88	88.76	89.88
Trade (369)	88.13	86.44	84.74	86.44	90.67	89.83	90.67	90.67	89.83	88.98	86.44	92.37
Wheat (212)	69.01	74.64	83.09	67.6	73.23	84.5	83.09	85.91	85.91	84.5	87.23	87.32
macro-avg. BEP	85.66	86.95	83.52	85.52	86.94	90.48	89.89	90.51	89.89	90.12	88.13	91.68
micro-avg. BEP	92.46	93.29	89.99	92.68	92.46	94.9	94.87	94.86	93.83	94.76	93	95.3

Table 8. Performance comparison on the Newsgroup dataset

Category (# of training documents)	<i>kNN (F_1)</i>						<i>SVM (F_1)</i>					
	<i>tf.idf</i>	<i>log</i>	<i>log</i>	<i>tf.rf</i>	<i>Delta</i>	<i>log</i>	<i>tf.idf</i>	<i>log</i>	<i>log</i>	<i>tf.rf</i>	<i>Delta</i>	<i>log</i>
		<i>tf.idf</i>	<i>tf.chi</i>		<i>tf.idf</i>	<i>tf.TRR</i>		<i>tf.idf</i>	<i>tf.chi</i>		<i>tf.idf</i>	<i>tf.TRR</i>
Atheism (800)	70.87	77.43	60.45	70.71	76.86	77.35	76.95	77.45	69.39	76.91	75.38	76.56
Graphics (800)	77.17	80.90	58.41	75.51	79.55	81.83	81.80	82.52	71.88	81.78	80.51	83.94
Windows.misc (800)	74.84	78.23	68.66	77.48	82.31	84.99	83.83	84.80	74.81	84.20	84.51	87.20
Pc.hardware (800)	68.65	73.98	56.20	69.88	73.65	77.49	76.61	77.88	68.61	76.92	75.09	79.09
Mac.hardware (800)	76.47	83.24	73.45	81.44	87.86	89.64	87.03	87.86	81.98	87.95	88.47	90.30
Windows.x (800)	83.56	87.62	71.92	86.88	89.30	90.77	88.69	89.64	82.05	88.90	89.46	91.10
Forsale (800)	69.41	77.88	72.82	76.89	79.07	83.33	83.67	85.00	78.59	82.79	77.92	84.10
Autos (800)	86.30	90.33	76.06	87.08	91.89	92.17	91.44	91.92	83.08	91.24	91.62	92.46
Motorcycles (800)	90.52	94.31	85.37	94.59	96.53	96.9	95.73	95.74	89.60	95.97	96.42	96.57
Baseball (800)	94.45	96.11	77.27	94.42	96.12	97.21	96.15	96.34	91.24	96.22	96.42	96.83
Hockey (800)	93.91	95.76	84.89	96.33	96.79	97.48	96.62	96.79	93.67	97.13	97.08	97.23
Crypt (800)	92.73	94.69	87.67	95.15	94.81	96.98	95.11	95.81	91.85	95.58	95.37	96.48
Electronics (800)	74.65	79.96	46.61	72.79	81.95	82.47	83.30	83.99	68.53	83.69	82.71	85.24
Med (800)	88.49	91.13	75.38	88.47	93.60	93.65	91.40	91.71	86.44	91.72	93.07	93.75
Space (797)	91.10	93.23	76.82	91.89	94.73	95.3	94.43	94.89	87.03	94.04	94.56	94.88
Christian (800)	87.29	91.36	94.32	93.13	95.82	94.89	97.11	97.98	96.54	98.50	98.81	98.36
Guns (800)	83.34	86.54	70.40	84.80	86.09	88.30	85.63	87.00	76.54	86.79	85.91	87.87

Midest (800)	90.93	93.69	86.24	92.94	93.11	94.30	93.13	93.69	91.62	93.30	92.40	93.24
Politics.misc (800)	70.71	75.04	55.56	70.61	73.44	76.35	75.10	75.51	65.85	74.99	72.98	74.54
Religion.misc (800)	57.29	61.18	52.17	58.24	60.85	64.04	61.83	62.74	61.57	63.91	58.36	61.96
macro-avg. F_1	81.13	85.13	71.53	82.96	86.22	87.78	86.78	87.46	80.54	87.13	86.35	88.09
micro-avg. F_1	81.2	85.17	71.55	82.79	86.01	87.85	87.07	87.74	81.41	87.37	86.64	88.31

Table 9. Performance comparison on the Korean Newsgroup dataset

Category (# of training documents)	$kNN (F_1)$						$SVM (F_1)$					
	$tf.idf$	log $tf.idf$	log $tf.chi$	$tf.rf$	Δ $tf.idf$	log $tf.TRR$	$tf.idf$	log $tf.idf$	log $tf.chi$	$tf.rf$	Δ $tf.idf$	log $tf.TRR$
han.arts.music (315)	85.05	84.73	68.25	79.01	82.3	87.59	87.64	87.5	79.06	87.59	82.81	88.21
han.comp.database (198)	65.35	68.42	61.72	55.81	67.05	71.14	72.61	76.02	65.88	74.07	74.86	79.01
han.comp.devtools (404)	59.77	58.13	35.8	42.55	66.44	69.47	74.61	75.37	58.27	73.01	71.68	78.46
han.comp.lang (1387)	79.45	79.24	69.63	73.53	82.8	83.34	85.69	85.78	77.83	84.34	82	87.1
han.comp.os.linux (1175)	74.66	75.73	62.23	73.69	80.2	81.61	82.32	82.73	70.54	80.86	80.24	83.24
han.comp.os.window (517)	68.44	70.41	48.08	62.2	69.03	74.19	73.46	74.45	64.27	73.75	69.67	74.15
han.comp.sys (304)	57.56	62.13	34.25	45.55	60.63	69.86	73.72	74.01	57.02	70.24	63.93	76.19
han.politics (1469)	93.44	94.45	83.96	90.72	93.3	95.71	95.53	95.44	91.38	92.65	92.03	94.6
han.rec.cars (291)	89.71	89.71	83.19	82.4	93.06	92.3	95.08	94.21	88.88	93.72	94.65	94.6
han.rec.games (261)	65.54	68.93	51.14	64.28	68.37	76.57	72.64	76.44	64.62	72.88	65.75	75.43
han.rec.movie (202)	88.26	91.62	80.61	84.61	88.34	89.88	91.11	91.62	88.17	87.35	89.24	87.86
han.rec.sports (130)	87.85	86.79	75.72	67.44	87.71	92.3	87.5	85.21	83.76	85.14	88.69	92.45
han.rec.travel (102)	79.01	83.33	64.36	75	87.05	88.63	84.09	84.09	71.57	80	78.65	89.12
han.sci (333)	80.46	81.56	40.55	70.49	78.78	83.13	83.73	83.44	69.71	77.5	79.57	78.52
han.soc.religion (136)	94.21	94.91	90.59	81.18	90.9	92.85	94.91	95.86	91.66	91.07	92.43	91.07
macro-avg. F_1	77.92	79.34	63.34	69.9	79.73	83.24	83.64	84.14	74.84	81.61	80.41	84.67
micro-avg. F_1	79.69	80.66	66.77	74.54	81.69	84.36	85.12	85.47	76.84	83.34	81.83	85.7

(a) Changes in performance according to the number of features in Reuters using kNN

(b) Changes in performance according to the number of features in Reuters using SVM

(c) Changes in performance according to the number of features in NG using kNN

(d) Changes in performance according to the number of features in NG using SVM

(e) Changes in performance according to the number of features in KNG using kNN

(f) Changes in performance according to the number of features in KNG using SVM

Figure 3. Comparison of $\log tf.TRR$ with other term weighting methods based on changes in performance according to the number of features

4.5. Summary of the experiment results

Table 10 summarizes the final results of the term weighting scheme used in this experiment. The results show that the proposed scheme, $\log tf.TRR$, achieved the best performance over all of the datasets and classifiers. The performance of $\log tf.TRR$ outperforms the traditional term weighting schemes ($tf.idf$ and $\log tf.idf$), $\log tf.chi$, $\Delta tf.idf$ and $tf.rf$. As expected, $tf.rf$ showed an unstable performance according to the dataset characteristics such as the variation in class size. In particular, $\log tf.TRR$ obtained a better performance for categories with a small number of documents (see Tables 7 and 9). This may lead to more improvements in macro-averaging than in micro-averaging. Note that the scores inside the parentheses denote the relative improvements over the best improvement among the other term weighting schemes. The statistical significance tests on kNN show that the differences from $\log tf.TRR$ to other schemes for all of the datasets are statistically significant ($p < 0.01$), and the statistical significance tests on SVM indicate that the differences in the Newsgroup dataset are all statistically significant ($p < 0.01$); the differences in Reuters are statistically significant ($p < 0.05$), except for $tf.rf$; and the differences in the Korean Newsgroup dataset are statistically significant ($p < 0.01$), except for $tf.idf$ and $\log tf.idf$.

Table 10. Final experiment results

			<i>tf.idf</i>	<i>log tf.idf</i>	<i>log tf.chi</i>	<i>tf.rf</i>	<i>Delta tf.idf</i>	<i>log tf.TRR</i>
<i>Reuters</i> (<i>BEP</i>)	<i>kNN</i>	<i>macro-averaging</i>	85.66	86.95	83.52	85.52	86.94	90.48 (+4.06)
		<i>micro-averaging</i>	92.46	93.29	89.99	92.68	92.46	94.90 (+1.72)
	<i>SVM</i>	<i>macro-averaging</i>	89.89	90.51	89.89	90.12	88.13	91.68 (+1.29)
		<i>micro-averaging</i>	94.87	94.86	93.83	94.76	93.00	95.30 (+0.45)
<i>NG</i> (<i>F₁</i>)	<i>kNN</i>	<i>macro-averaging</i>	81.13	85.13	71.53	82.96	86.22	87.78 (+1.81)
		<i>micro-averaging</i>	81.20	85.17	71.55	82.79	86.01	87.75 (+2.02)
	<i>SVM</i>	<i>macro-averaging</i>	86.78	87.46	80.54	87.13	86.35	88.09 (+0.72)
		<i>micro-averaging</i>	87.07	87.74	81.41	87.37	86.64	88.43 (+0.78)
<i>KNG</i> (<i>F₁</i>)	<i>kNN</i>	<i>macro-averaging</i>	77.92	79.34	63.34	69.9	79.73	83.24 (+4.40)
		<i>micro-averaging</i>	79.69	80.66	66.77	74.54	81.69	84.36 (+3.27)
	<i>SVM</i>	<i>macro-averaging</i>	83.64	84.14	74.84	81.61	80.41	84.67 (+0.62)
		<i>micro-averaging</i>	85.12	85.47	76.84	83.34	81.83	85.70 (+0.27)

4.6. Analysis of the experiment results

We here try to analyze the experiment results in terms of how the proposed scheme works well on categories with ambiguity and low performances. We expect that the proposed scheme, *log tf.TRR*, can achieve better performance than other schemes in the categories with ambiguity and low performances. We first observe how much difference of F_1 scores is between categories with some ambiguity and the other categories in Table 11. The hierarchical information of *NG* and *KNG* data sets is exploited in order to select ambiguous categories. The five subcategories related computer (Graphics, Windows.misc, PC.hardware, Mac.hardware, Windows.x) are chosen from *NG* as categories with some ambiguity, and the six subcategories related computer (han.comp.database, han.comp.devtools, han.comp.lang, han.comp.os.linux, han.comp.os.window, han.comp.sys) and the five subcategories related recreation (han.rec.cars, han.rec.games, han.rec.movie, han.rec.sports, han.rec.travel) are from *KNG*. As a result, the proposed scheme made more score differences on categories with ambiguity. In addition, when the score differences between the bottom 30% and the top 30% of categories sorted by F_1 scores of *log tf.idf* are compared in Table 12, the similar result is observed. The categories with low performance show much more difference in F_1 scores than ones with high performance. In particular, the score differences of SVM on unambiguous categories or categories with high performance are small, and even when comparing to *tf.idf* and *log tf.idf*, they have minus

differences in *KNG*. Eventually, the proposed scheme highly outperforms other schemes on difficult categories to be classified, whereas it is not expected to get high improvement on easy categories.

Table 11. Comparison between categories with some ambiguity (Ambi) and the other categories (Others) in the averages of F_1 score differences from the proposed scheme to other schemes

		<i>kNN (Average of Differences)</i>						<i>SVM (Average of Differences)</i>					
		<i>tf.idf</i>	<i>log</i> <i>tf.idf</i>	<i>log</i> <i>tf.chi</i>	<i>tf.rf</i>	<i>Delta</i> <i>tf.idf</i>	<i>Average</i>	<i>tf.idf</i>	<i>log</i> <i>tf.idf</i>	<i>log</i> <i>tf.chi</i>	<i>tf.rf</i>	<i>Delta</i> <i>tf.idf</i>	<i>Average</i>
<i>NG</i>	Ambi	8.81	4.15	19.22	6.71	2.41	8.26	2.73	1.79	10.46	2.38	2.72	4.02
	Others	5.92	2.14	15.25	4.18	1.27	5.75	0.83	0.23	6.57	0.49	1.4	1.90
<i>KNG</i>	Ambi	6.63	4.89	19.96	14.62	3.47	9.91	2.22	1.6	11.27	3.86	5.23	4.84
	Others	1.53	0.91	18.98	9.47	3.5	6.88	-2.35	-2.46	5.15	0.9	1.39	0.53

Table 12. Comparison between the bottom 30% (Low 30%) and the top 30% (High 30%) of categories in the averages of F_1 score differences from the proposed scheme to other schemes

		<i>kNN (Average of Differences)</i>						<i>SVM (Average of Differences)</i>					
		<i>tf.idf</i>	<i>log</i> <i>tf.idf</i>	<i>log</i> <i>tf.chi</i>	<i>tf.rf</i>	<i>Delta</i> <i>tf.idf</i>	<i>Average</i>	<i>tf.idf</i>	<i>log</i> <i>tf.idf</i>	<i>log</i> <i>tf.chi</i>	<i>tf.rf</i>	<i>Delta</i> <i>tf.idf</i>	<i>Average</i>
<i>Reuters</i>	Low 30%	8.82	6.35	9.20	8.76	4.13	7.45	1.92	1.54	3.11	0.50	4.46	2.31
	High 30%	1.90	0.52	4.39	4.09	3.07	2.79	0.51	0.21	1.86	0.75	1.22	0.91
<i>NG</i>	Low 30%	8.51	3.19	19.08	7.06	2.56	8.08	1.30	0.52	9.70	0.88	2.71	3.02
	High 30%	4.59	1.99	11.50	2.09	0.74	4.18	0.76	0.34	4.58	0.41	0.36	1.29
<i>KNG</i>	Low 30%	8.91	6.64	26.05	18.17	5.94	13.14	3.24	1.39	14.64	3.86	7.47	6.12
	High 30%	1.91	1.11	9.79	11.34	1.95	5.22	-1.59	-1.66	3.44	0.79	1.04	0.40

Moreover, we pointed out that the ratio of document numbers of positive and negative classes under the OvA method for each category may vary in practice due to uneven size of categories so the proposed method is superior to *rf* in section 3.4. Table 13 shows the comparison results of F_1 score differences between *rf* and the proposed scheme on categories with small number of documents and the other categories. As a result, the score differences on categories with small number of documents are more than ones on the other categories. This is evidence that *rf* suffers more from the size difference of positive and negative classes in small categories. In addition, we observed the standard deviation values of F_1 score differences on all of the data sets because *NG*

has even size of categories but *Reuters* and *KNG* have uneven size. As Table 13 illustrates, the variation of score differences on NG with even size of categories is more stable than ones on *Reuters* and *KNG*.

Table 13. Comparison between the bottom 30% of categories with respect to small number of documents (Small 30%) and the other categories (Others) in the averages of F_1 score differences from the proposed scheme to *tf.rf*, and comparison among data sets in the average and standard deviation (St.Dev) values of F_1 score differences from the proposed scheme to *tf.rf*

		<i>kNN</i> (Average of Differences)	<i>SVM</i> (Average of Differences)		<i>kNN</i> (Differences)		<i>SVM</i> (Differences)	
		<i>tf.rf</i>	<i>tf.rf</i>		Average	St.Dev	Average	St.Dev
<i>Reuters</i>	Small 30%	8.76	3.92	<i>Reuters</i>	4.96	5.13	1.56	2.84
	Others	3.34	0.54	<i>NG</i>	4.81	2.43	0.95	1.17
<i>KNG</i>	Small 30%	14.15	4.38	<i>KNG</i>	13.34	6.63	3.06	2.73
	Others	12.94	2.39					

5. Conclusions and Future Work

In this paper, we utilized class information for term weighting for text classification. The odds of positive and negative class probabilities are applied to term weighting for text classification to utilize class information. As a result, the proposed schemes performed consistently well on the three benchmark datasets used, as well as on the *kNN* and *SVM* classifiers, and outperformed the new recently reported term weighting schemes, *tf.rf* and Delta *tf.idf*, as well as traditional term weighting schemes, *tf.idf* and *log tf.idf*.

In the future, we would like to apply these schemes to other applications such as question classification in cQA systems. In addition, since term weighting is the most basic component of other text mining techniques and text classification, we expect that our term weighting scheme can be integrated into various text-mining tasks such as a sentiment analysis and text summarization.

References:

Aggarwal, C. C. and Zhai, C. (2012). A Survey of Text Classification Algorithms. *Chapter 6, Mining Text Data*.

- Aphinyanaphongs, Y., Fu, L.D., Li, Z., Peskin, E.R., Efstathiadis, E., Aliferis, C.F. and Statnikov, A. (2014). A comprehensive empirical comparison of modern supervised classification and feature selection methods for text categorization. *Journal of the American Society for Information Science and Technology*, Article first published online: 1 March 2014.
- Banerjee, A. and Basu, S. (2007). Topic Models over Text Streams: A Study of Batch and Online Unsupervised Learning. In *Proceedings of Conference on Data Mining*. pp. 431-436.
- Debole, F. and Sebastiani, F. (2003). Supervised term weighting for automated text categorization. In *Proceedings of the 2003 ACM symposium on applied computing*. pp. 784-788, 2003.
- Deng, Z.H., Tang, S.W., Yang, D.Q., Zhang, M. Li, L.Y. and Xie, K.Q. (2004). A comparative study on feature weight in text categorization, In *Proceedings of APWeb 2004, LNCS 3007*, pp. 588-597, 2004.
- Galar, M., Fernández, Alberto, Barrenechea, E. Bustince, H. and Herrera F. (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, Elsevier Science, Vol. 44, No. 8, pp. 1761-1776, 2011.
- Gliozzo, A. Strapparava, C. and Dagan, I. (2005). Investigating Unsupervised Learning for Text Categorization Bootstrapping. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 129-136.
- Gordon, M.D. and Lenk, P. (1991). A utility theoretic examination of the probability ranking principle in information retrieval. *Journal of the American Society for Information Science and Technology*, Vol. 42, No. 10, pp. 703-714.
- Gordon, M.D. and Lenk, P. (1991). When is the probability ranking principle suboptimal? *Journal of the American Society for Information Science and Technology*, Vol. 43, No. 1, pp. 1-14.
- Han, H, Ko, Y and Seo, J. (2007). Using the Revised EM Algorithm to Remove Noisy Data for Improving the One-against-the-rest Method in Binary Text Classification. *Information Processing and Management, Pergamon-Elsevier Science*, Vol. 43, No. 5, pp. 1281-1293.
- Ha-Thuc, V. and Renders, J.-M., (2011). Large-Scale Hierarchical Text Classification without Labelled Data. In *Proceedings of the fourth International Conference on Web Search and Data Mining*. pp. 685-694.

- Jiang, J. and Zhai, C. (2007). A Two-Stage Approach to Domain Adaptation for Statistical Classifiers. *Proceedings of the 16th ACM International Conference on Information and Knowledge Management*, pp. 401-410.
- Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In *Proceedings of European Conference on Machine Learning (ECML)*, pp. 137-142.
- Ke, W. (2012). Least Information Document Representation for Automated Text Classification. In *Proceedings of the American Society for Information Science and Technology (ASIST 2012)*, Vol. 49, No. 1, pp. 1-10.
- Ko, Y. (2012). A study of term weighting scheme using class information for text classification. In *proceedings of the 35th Annual International ACM SIGIR Conference (SIGIR 2012)*, pp. 1029-1030, in Portland Oregon, USA, August 2012.
- Ko, Y., Park, J. and Seo, J. (2004). Improving text categorization using the importance of sentences. *Information Processing and Management, Pergamon-Elsevier Science*, Vol. 40, No. 1, pp. 65-79.
- Ko, Y. and Seo J. (2009). Text classification from unlabeled documents with bootstrapping and feature projection techniques. *Information Processing and Management, Pergamon-Elsevier Science*, Vol. 45, No. 1, pp. 70-83, January 2009.
- Lan, M., Tan, C.L. and Low, H.B. (2006). Proposing a new term weighting scheme for text categorization. In *proceedings of 21st National Conference on Artificial Intelligence, AAAI-2006*, pp. 16-20.
- Lan, M., Tan, C.L., Su, J. and Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 31, No. 4, pp. 736-741, April 2009.
- Lewis, D.D. (1997). Reuters-21578 text categorization test collection. ReadMe file of Distribution 1.0.
- Manning, C.D., and Raghavan, P. and Schütze H. (2008). *Introduction to Information Retrieval*, Cambridge University Press.
- Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT press.
- Martineau, J. and Finin, T. (2009). Delta TFIDF: an improved feature space for sentiment analysis. In *Proceedings of 3rd AAAI International Conference on Weblogs and Social Media*, May 2009.

- McCallum, A. and Nigam, K. (1998). A comparison of event models for naive Bayes text classification. *AAAI '98 workshop on Learning for Text Categorization*. pp. 41-48.
- Nigam, K. P., (2001). Using unlabeled data to improve text classification, *the dissertation for the degree of Doctor of Philosophy*.
- Paltoglou, G. and Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 1386 - 1395, 2010.
- Salton, G. and Buckley, C. (1998). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, Vol. 24, No. 5, pp. 513-523.
- Sebastiani, F. (2002). Machine learning in automatic text retrieval. *ACM Computing Surveys (CSUR)*, Vol. 34, No. 1, pp. 1-47.
- Smucker, M., Allan, J. and Carterette, B. (2007). A Comparison of Statistical Significance Tests for Information Retrieval Evaluation, In *Proceedings of the ACM Sixteenth Conference on Information and Knowledge Management (CIKM 2007)*, pp. 623-632.
- Soucy, P. and Mineau, G.W. (2005). Beyond tfidf weighting for text categorization in the vector space model. In *Proceedings of International Joint Conference on Artificial Intelligence, IJCAI 2005*, pp. 1130-1135, 2005.
- Sun, A., Lim, E.-P. and Liu, Y. (2009). On Strategies for Imbalanced Text Classification Using SVM: A Comparative Study. *Decision Support Systems*, Vol. 48, No. 1, pp. 191-201.
- Sun, A., Lim, E.-P. and Ng, W.-K. (2003). Performance Measurement Framework for Hierarchical Text Classification. *Journal of the American Society for Information Science and Technology*, Vol. 54, No. 11, pp. 1014-1028.
- Van Rijsbergen, C.J. (1979). *Information Retrieval*, 2nd edition. Butterworths.
- Yang Y. and Pedersen J.P. (1997). Feature selection in statistical learning of text categorization. In *Proceedings of The Fourteenth International Conference on Machine Learning*, pp. 412-420.
- Yang Y. (1999). An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, Vol 1, No. 1/2, pp. 67-88.
- Yang Y. and Liu X. (1999). A re-examination of text categorization methods. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pp. 42-49.

- Yoon, Y., Lee, C. and Lee, G.G. (2006). An Effective Procedure for Constructing a Hierarchical Text Classification System. *Journal of the American Society for Information Science and Technology*. Vol. 57, No. 3, pp. 431-442.
- Yu, H., Zhai, C. and Han, J. (2003). Text Classification from Positive and Unlabeled Documents. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, pp. 232-239.