Multi-label Text Categorization with Model Combination based on F_1-score Maximization

Akinori Fujino, Hideki Isozaki, and **Jun Suzuki**
NTT Communication Science Laboratories
NTT Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan 619-0237
{a.fujino,isozaki,jun}@cslab.kecl.ntt.co.jp

Abstract

Text categorization is a fundamental task in natural language processing, and is generally defined as a multi-label categorization problem, where each text document is assigned to one or more categories. We focus on providing good statistical classifiers with a generalization ability for multi-label categorization and present a classifier design method based on model combination and F_1-score maximization. In our formulation, we first design multiple models for binary classification per category. Then, we combine these models to maximize the F_1-score of a training dataset. Our experimental results confirmed that our proposed method was useful especially for datasets where there were many combinations of category labels.

1 Introduction

Text categorization is a fundamental task in such aspects of natural language processing as information retrieval, information extraction, and text mining. Since a text document often belongs to multiple categories in real tasks such as web pages and international patent categorization, text categorization is generally defined as assigning one or more predefined category labels to each data sample. Therefore, developing better classifiers with a generalization ability for such multi-label categorization tasks is an important issue in the field of machine learning.

A major and conventional machine learning approach to multi-label categorization is based on binary classification. With this approach, we assume the independence of categories and design a binary classifier for each category that determines whether or not to assign a category label to data samples. Statistical classifiers such as the logistic regression model (LRM), the support vector machine (SVM), and naive Bayes are employed as binary classifiers (Joachims, 1998).

In text categorization, the F_1-score is often used to evaluate classifier performance. Recently, methods for training binary classifiers to maximize the F_1-score have been proposed for SVM (Joachims, 2005) and LRM (Jansche, 2005). It was confirmed experimentally that these training methods were more effective for obtaining binary classifiers with better F_1-score performance than the minimum error rate and maximum likelihood used for training conventional classifiers, especially when there was a large imbalance between positive and negative samples. In multi-label categorization, macro- and micro-averaged F_1-scores are often used to evaluate classification performance. Therefore, we can expect to improve multi-label classification performance by using binary classifiers trained to maximize the F_1-score.

On the other hand, classification frameworks based on classifier combination have also been studied in many previous works such as (Wolpert, 1992; Larkey and Croft, 1996; Ting and Witten, 1999; Ghahramani and Kim, 2003; Bell et al., 2005; Fumera and Roli, 2005), to provide better classifier systems. In the classifier combination research field, it is known that weighted linear combinations of multiple classifiers often provide better classification performance than individual classifiers.

We present a classifier design method based on the combination of multiple binary classifiers to improve multi-label classification performance. In our framework, we first train multiple binary classifiers for each category. Then, we combine these binary classifiers with weights estimated to maximize micro- or macro-averaged F_1-scores, which are often used for evaluating multi-label classifiers. To estimate combination weights, we extend the F_1-score maximization training algorithm for LRM described in (Jansche, 2005). Using three real text datasets, we show experimentally that our classifier design method is more effective than the conventional binary classification approaches to multi-label categorization.

Our method is based on a binary classification approach. However, Kazawa et al. (2005) proposed a method for modeling a map directly from data samples to the combination of assigned category labels, and confirmed experimentally that the method outperformed conventional binary classification approaches. Therefore, we also compare our method with the direct mapping method experimentally.

2 F_1-score Maximization Training of LRM

We first review the F_1-score maximization training method for linear models using a logistic function described in (Jansche, 2005). The method was proposed in binary classification settings, where classifiers determine a class label assignment $y \in \{1, 0\}$ for a data sample represented by a feature vector $\boldsymbol{x}$. Here, $y^{(n)} = 1\ (= 0)$ indicates that the class label is assigned (unassigned) to the nth feature vector $\boldsymbol{x}^{(n)}$.

The discriminative function of a binary classifier based on a linear model is often defined as

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{\theta}_1^t \boldsymbol{x} + \theta_0, \tag{1}$$

where $\boldsymbol{\theta} = (\theta_0, \boldsymbol{\theta}_1^t)^t$ is a model parameter vector, and $\boldsymbol{\theta}_1^t \boldsymbol{x}$ implies the inner product of $\boldsymbol{\theta}_1$ and $\boldsymbol{x}$. A binary classifier using $f(\boldsymbol{x}; \boldsymbol{\theta})$ outputs a predicted class label assignment $\hat{y}$ for $\boldsymbol{x}$ as $\hat{y}^{(n)} = 1\ (= 0)$ when $f(\boldsymbol{x}^{(n)}; \boldsymbol{\theta}) \geq 0\ (< 0)$.

An LRM is a binary classifier that uses the discriminative function $f(\boldsymbol{x}; \boldsymbol{\theta})$. In this model, the class posterior probability distribution is defined by using a logistic function:

$$g(z) = \{1 + \exp(-z)\}^{-1}. \tag{2}$$

That is, $P(y = 1|\boldsymbol{x}; \boldsymbol{\theta}) = g(f(\boldsymbol{x}; \boldsymbol{\theta}))$ and $P(y = 0|\boldsymbol{x}; \boldsymbol{\theta}) = 1 - P(y = 1|\boldsymbol{x}; \boldsymbol{\theta}) = g(-f(\boldsymbol{x}; \boldsymbol{\theta}))$. The LRM determines that $y^{(n)} = 1\ (= 0)$ when $P(y = 1|\boldsymbol{x}^{(n)}; \boldsymbol{\theta}) \geq 0.5\ (< 0.5)$, since $g(0) = 0.5$. The model parameter vector $\boldsymbol{\theta}$ is usually estimated to maximize the likelihood of $P(y|\boldsymbol{x}; \boldsymbol{\theta})$ for training dataset $D = \{\boldsymbol{x}^{(m)}, y^{(m)}\}_{m=1}^M$ and the prior probability density of $\boldsymbol{\theta}$:

$$J_R(\boldsymbol{\theta}) = \sum_{m=1}^{M} \log P(y^{(m)}|\boldsymbol{x}^{(m)}; \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}). \tag{3}$$

In this paper, the classifier design approach that employs this training method is called LRM-L.

By contrast, in the training method proposed by (Jansche, 2005), the discriminative function $f(\boldsymbol{x}; \boldsymbol{w})$ is estimated to maximize the F_1-score of training dataset D. This training method employs an approximate form of the F_1-score obtained by using a logistic function.

The F_1-score is defined as $F_1 = 2(1/PR + 1/RE)^{-1}$, where PR and RE represent precision and recall defined as $PR = C/A$ and $RE = C/B$, respectively. Here, C represents the number of data samples whose true and predicted class label assignments, $y^{(n)}$ and $\hat{y}^{(n)}$, respectively, correspond to 1. A represents the number of data samples for which $\hat{y}^{(n)} = 1$. B represents the number of data samples for which $y^{(n)} = 1$. C, A, and B are computed for training dataset D as $C = \sum_{m=1}^M y^{(m)} \hat{y}^{(m)}$, $A = \sum_{m=1}^M \hat{y}^{(m)}$, and $B = \sum_{m=1}^M y^{(m)}$.

In (Jansche, 2005), $\hat{y}^{(m)}$ was approximated by using the discriminative and logistic functions shown in Eqs. (1) and (2) as

$$\hat{y}^{(m)} \approx g(\gamma f(\boldsymbol{x}^{(m)}; \boldsymbol{\theta})),\ \gamma > 0, \tag{4}$$

because $\lim_{\gamma \to \infty} g(\gamma f(\boldsymbol{x}^{(m)}; \boldsymbol{\theta})) = \hat{y}^{(m)}$. Then, an approximate distribution of the F_1-score for training dataset D was provided as

$$\tilde{F}_1(\boldsymbol{\theta}) = \frac{2 \sum_{m=1}^M g(\gamma f(\boldsymbol{x}; \boldsymbol{\theta})) y^{(m)}}{\sum_{m=1}^M y^{(m)} + \sum_{m=1}^M g(\gamma f(\boldsymbol{x}; \boldsymbol{\theta}))}. \tag{5}$$

The $\boldsymbol{\theta}$ estimate for the discriminative function $f(\boldsymbol{x}; \boldsymbol{\theta})$ can be computed to maximize $J_F(\boldsymbol{\theta}) = \log \tilde{F}_1(\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})$ around the initial $\boldsymbol{\theta}$ value by using a gradient method. In this paper, the classifier design approach that uses this training method is called LRM-F.

3 Proposed Method

We propose a framework for designing a multi-label classifier based on the combination of multiple models. In our formulation, multiple models are combined with weights estimated to maximize the F_1-scores of the training dataset. In this section, we show our formulation for model combination and training methods for combination weights.

3.1 Combination of Multiple Models for Multi-label Categorization

Multi-label categorization is the task of selecting multiple category labels from K pre-defined category labels for each data sample. Multi-label classifiers provide a map from a feature vector $\boldsymbol{x}$ to a category label assignment vector $\boldsymbol{y} = (y_1, \ldots, y_k, \ldots, y_K)^t$, where $y_k^{(n)} = 1\ (= 0)$ indicates that the kth category label is assigned (unassigned) to $\boldsymbol{x}^{(n)}$.

In our formulation, we first design multiple models for binary classification per category and obtain $J \times K$ discriminative functions, where J is the number of models. The discriminative function of the jth model for the kth category is denoted by $f_{jk}(\boldsymbol{x}; \boldsymbol{\theta}_{jk})$, where $\boldsymbol{\theta}_{jk}$ represents the model parameter vector. Let $\Theta = \{\boldsymbol{\theta}_{jk}\}_{j,k}$ be a model parameter set. We train model parameter vectors individually with each model training algorithm and obtain the estimate $\hat{\Theta} = \{\hat{\boldsymbol{\theta}}_{jk}\}_{jk}$. Then, we define the discriminative function of our multi-label classifier by combining multiple models such as

$$f_k(\boldsymbol{x}; \hat{\Theta}, \boldsymbol{w}) = \sum_{j=1}^{J} w_j f_{jk}(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_{jk}) + w_0,\ \forall k, \quad (6)$$

where $\boldsymbol{w} = (w_0, w_1, \ldots, w_j, \ldots, w_J)^t$ is a weight parameter vector and is independent of k. w_j provides the combination weight of the jth model, and w_0 is the bias factor for adjusting the threshold of the category label assignment.

We estimate the $\boldsymbol{w}$ value to maximize the micro-averaged F_1-score (F_μ), which is often used for evaluating multi-label categorization performance. The F_μ-score of training dataset $D = \{\boldsymbol{x}^{(m)}, \boldsymbol{y}^{(m)}\}_{m=1}^{M}$ is calculated as

$$F_\mu = \frac{2\sum_{m=1}^{M}\sum_{k=1}^{K} y_k^{(m)} \hat{y}_k^{(m)}}{\sum_{m=1}^{M}\sum_{k=1}^{K} y_k^{(m)} + \sum_{m=1}^{M}\sum_{k=1}^{K} \hat{y}_k^{(m)}}, \quad (7)$$

We provide an approximate form of the F_μ-score of the training dataset, $\tilde{F}_\mu(\hat{\Theta}, \boldsymbol{w})$, by using the approximation:

$$\hat{y}_k^{(m)} \approx g(\gamma f_k(\boldsymbol{x}^{(m)}; \hat{\Theta}, \boldsymbol{w})),\ \gamma > 0, \quad (8)$$

as shown in Eq. (4). In our proposed method, $\boldsymbol{w}$ is estimated to maximize $\tilde{F}_\mu(\hat{\Theta}, \boldsymbol{w})$.

However, training dataset D is also used to estimate Θ. Using the same training data samples for both Θ and $\boldsymbol{w}$ may lead to a bias estimation of $\boldsymbol{w}$. Thus, we used an n-fold cross-validation of the training data samples to estimate $\boldsymbol{w}$ as in (Wolpert, 1992). Let $\hat{\Theta}^{(-m)}$ be the model parameter set estimated by using $n-1$ training data subsets not containing $\{\boldsymbol{x}^{(m)}, \boldsymbol{y}^{(m)}\}$. Then, using

$$\tilde{F}_\mu = \frac{2\sum_{m,k} y_k^{(m)} g(\gamma f_k(\boldsymbol{x}; \hat{\Theta}^{(-m)}, \boldsymbol{w}))}{\sum_{m,k} y_k^{(m)} + \sum_{m,k} g(\gamma f_k(\boldsymbol{x}; \hat{\Theta}^{(-m)}, \boldsymbol{w}))}, \quad (9)$$

we provide the objective function of $\boldsymbol{w}$ such that

$$J_\mu(\boldsymbol{w}) = \log \tilde{F}_\mu + \log p(\boldsymbol{w}), \quad (10)$$

where $p(\boldsymbol{w})$ is a prior probability density of $\boldsymbol{w}$. We use a Gaussian prior (Chen and Rosenfeld, 1999) with the form as $p(\boldsymbol{w}) \propto \prod_{j=0}^{J} \exp\{-(w_j - \rho_j)^2/2\sigma_j^2\}$, where σ_j, and ρ_j are hyperparameters in the Gaussian prior. We compute an estimate of $\boldsymbol{w}$ to maximize $J_\mu(\boldsymbol{w})$ around the initial $\boldsymbol{w}$ value by using a quasi-Newton method. In this paper, this formulation is called *model combination by micro-averaged F_1-score maximization* (MC-F_μ).

3.2 Other Training Methods

In multi-label categorization problems, the macro-averaged F_1-score (F_M) is also used to evaluate classifiers. Moreover, the average labeling F_1-score (F_L) has been used to evaluate the average labeling performance of classifiers for data samples (Kazawa et al., 2005). These F_1-scores are computed for training dataset D as

$$F_M = \frac{1}{K}\sum_{k=1}^{K} \frac{2\sum_{m=1}^{M} y_k^{(m)} \hat{y}_k^{(m)}}{\sum_{m=1}^{M} y_k^{(m)} + \sum_{m=1}^{M} \hat{y}_k^{(m)}}, \quad (11)$$

$$F_L = \frac{1}{M}\sum_{m=1}^{M} \frac{2\sum_{k=1}^{K} y_k^{(m)} \hat{y}_k^{(m)}}{\sum_{k=1}^{K} y_k^{(m)} + \sum_{k=1}^{K} \hat{y}_k^{(m)}}. \quad (12)$$

Using Eq. (8), we can also obtain the approximate forms, $\tilde{F}_M(\hat{\Theta}, \boldsymbol{w})$ and $\tilde{F}_L(\hat{\Theta}, \boldsymbol{w})$, of the F_M-

and F_L-scores, and then present similar objective functions to that for the F_μ-score. Therefore, in the next section, we examine experimentally the performance of classifiers obtained by estimating $\boldsymbol{w}$ to maximize $\tilde{F}_M(\hat{\Theta}, \boldsymbol{w})$ and $\tilde{F}_L(\hat{\Theta}, \boldsymbol{w})$. In this paper, these model combination methods based on F_M- and F_L-scores are called MC-F_M and MC-F_L, respectively.

4 Experiments

4.1 Test Collections

To evaluate our proposed method empirically, we used three test collections: Reuters-21578 (Reuters), WIPO-alpha (WIPO), and Japanese Patent (JPAT) datasets. Reuters and WIPO are English document datasets and have often been employed for benchmark tests of multi-label classifiers.

The Reuters dataset contains news articles from the Reuters newswire and consists of 135 topic categories. Following the setup in (Yang and Liu, 1999), we extracted 7770 and 3019 articles as training and test samples, respectively. A subset consisting of the training and test samples contained 90 topic categories. We removed vocabulary words included either in the stoplist or in only one article. There were 16365 vocabulary words in the dataset.

The WIPO dataset consists of patent documents categorized using the International Patent Classification (IPC) taxonomy (Fall et al., 2003). The IPC taxonomy has four hierarchical layers: *Section*, *Class*, *Subclass*, and *Group*. Using patent documents belonging to *Section D* (textiles; paper), we evaluated classifiers in a task that consisted of selecting assigned category labels from 160 groups for each patent document. Following the setting provided in the dataset, we extracted 1352 and 358 patent documents as training and test samples, respectively. We removed vocabulary words in the same way as for Reuters. There were 45895 vocabulary words in the dataset.

The JPAT dataset (Iwayama et al., 2007) consists of Japanese patent documents published between 1993 and 1999 by the Japanese Patent Office. These documents are categorized using a taxonomy consisting of *Themes* and *F-terms*. The themes are top-label categories, and the patent documents belonging to each theme are categorized by using F-terms. Using patent documents belonging to *Theme 5J104*, we evaluated classifiers in a task that consisted of selecting assigned category labels from 268 F-terms for each patent document. 1920 patent documents published between 1993 and 1997 were used as training samples, and 544 patent documents published between 1998 and 1999 were used test samples. We extracted Japanese nouns, verbs, and adjectives from patent documents by using a morphological analyzer named MeCab [1], and removed vocabulary words included in only one patent document. There were 21135 vocabulary words in the dataset.

	Reuters	WIPO	JPAT
N_{av}	1.17	1.28	10.5
N_{max}	15	6	40
K	90	160	268
N_{ds}	10789	1710	2464
N_{LC}	468	378	2430
N_{ds}/N_{LC}	23.1	4.52	1.01

Table 1: Statistical information of three datasets: N_{av} and N_{max} are the average and maximum number of assigned category labels per data sample, respectively. K and N_{ds} are the number of category labels and data samples, respectively. N_{LC} is the number of category label combinations appearing in each dataset.

Table 1 shows statistical information about the category label assignment of the data samples for the three datasets. The average numbers of assigned category labels per data sample, N_{av}, for Reuters and WIPO were close to 1 and much smaller than that for JPAT. The number of category label combinations, N_{LC}, included in JPAT was larger than those for Reuters and WIPO. These statistical information results show that JPAT is a more *complex* multi-label dataset than Reuters or WIPO.

4.2 Experimental Settings

For text categorization tasks, we employed word-frequency vectors of documents as feature vectors input into classifiers, using the independent word-based representation, known as the Bag-of-Words (BOW) representation. We normalized the L1-norms of the word-frequency vectors to 1, to mitigate the effect of vector size on computation. We did not employ any word weighting methods such as inverse document frequency (IDF).

[1] http://mecab.sourceforge.net/

We constructed three multi-label text classifiers based on our proposed model combination methods, MC-F_μ, MC-F_M, and MC-F_L, where LRM and SVM ($J = 2$) were employed as binary classification models combined with each method. We trained the LRM by using LRM-L described in Section 2, where a Gaussian prior was used as the prior probability density of the parameter vectors. We provided the SVM by using SVM^{light} [2] (SVM-L), where we employed a linear kernel function and tuned the C (penalty cost) parameter as a hyperparameter.

To evaluate our proposed method, we examined the micro- and macro-averaged, and average labeling F_1-scores (F_μ, F_M, and F_L), of test samples obtained with the three classifiers based on MC-F_μ, MC-F_M, and MC-F_L. We compared the performance of the three classifiers with that of two binary classification approaches, where LRM-L or SVM-L was used for binary classification.

We also examined two binary classification approaches using LRM-F and SVM-F. For LRM-F, we used a Gaussian prior and provided the initial parameter vector with a parameter estimate obtained with LRM-L. SVM-F is a binary classifier design approach that employs SVM^{perf} [3]. For SVM-F, we used a linear kernel function, set the L (loss parameter) parameter to maximize the F_1-score, and tuned the C (penalty cost) parameter as a hyperparameter.

Moreover, we examined the performance of the *Maximal Margin Labeling* (MML) method (Kazawa et al., 2005), which models the map from feature vectors to category label assignment vectors, because it was reported that MML provides better performance than binary classification approaches.

We tuned the hyperparameter of SVM-F for JPAT to provide good performance for test samples, because the computational cost for training was high. We tuned the other hyperparameters by using a 10-fold cross-validation of training samples.

4.3 Results and Discussion

In Table 2, we show the classification performance obtained for three datasets with our proposed and other methods described in Section 4.2. We examined nine evaluation scores: the micro-averaged F_1-score (F_μ), precision (P_μ), and recall (R_μ), the macro-averaged F_1-score (F_M), precision (P_M), and recall (R_M), and the average labeling F_1-score (F_L), precision (P_L), and recall (R_L) of the test samples. F_M and P_M were calculated by regarding both the F_1-score and precision as zero for the categories where there were no data samples predicted as positive samples.

Method	F_μ (P_μ/R_μ)	F_M (P_M/R_M)	F_L (P_L/R_M)
MC-F_μ	87.0 (87.4/86.7)	51.3 (60.0/48.4)	90.0 (90.1/92.3)
MC-F_M	85.0 (80.8/89.5)	53.9 (54.9/58.4)	89.7 (88.5/94.1)
MC-F_L	86.3 (84.3/88.3)	53.4 (59.6/52.6)	90.0 (89.3/93.6)
LRM-L	85.2 (87.3/83.2)	46.1 (55.0/43.1)	86.9 (87.6/88.6)
LRM-F	85.2 (87.2/83.2)	47.4 (58.5/42.7)	87.0 (87.6/88.7)
SVM-L	87.1 (92.9/82.0)	48.9 (58.9/45.8)	88.1 (89.3/88.8)
SVM-F	82.4 (78.9/86.2)	51.4 (49.4/60.1)	87.4 (86.9/91.4)
MML	**87.8** (92.6/83.4)	**59.3** (62.6/60.0)	**91.2** (91.7/93.2)

(a) Reuters

Method	F_μ (P_μ/R_μ)	F_M (P_M/R_M)	F_L (P_L/R_M)
MC-F_μ	**51.4** (57.3/46.6)	30.4 (35.8/30.3)	46.9 (48.3/51.5)
MC-F_M	48.1 (46.1/50.4)	32.2 (33.8/36.0)	46.8 (46.3/56.0)
MC-F_L	48.6 (45.8/51.9)	**32.5** (33.4/36.5)	47.1 (46.4/56.8)
LRM-L	40.5 (68.0/28.9)	22.1 (33.7/17.9)	32.7 (36.5/32.0)
LRM-F	41.0 (68.6/29.2)	22.3 (34.0/18.1)	33.2 (37.0/32.4)
SVM-L	41.8 (61.9/31.5)	24.4 (34.2/21.0)	35.1 (38.8/35.3)
SVM-F	48.3 (53.8/43.8)	32.3 (37.4/31.8)	45.6 (47.9/49.6)
MML	48.6 (54.9/43.6)	30.8 (36.5/29.7)	**49.4** (56.2/48.4)

(b) WIPO

Method	F_μ (P_μ/R_μ)	F_M (P_M/R_M)	F_L (P_L/R_M)
MC-F_μ	41.8 (42.6/41.1)	17.5 (21.4/17.4)	40.2 (43.5/44.4)
MC-F_M	40.6 (35.8/46.7)	**20.2** (20.4/23.1)	39.4 (37.7/50.6)
MC-F_L	**42.1** (42.3/41.9)	17.6 (21.1/17.8)	**40.5** (43.2/45.2)
LRM-L	33.9 (44.4/27.4)	15.8 (20.9/14.0)	32.2 (46.5/29.9)
LRM-F	36.9 (44.6/31.5)	16.9 (22.9/14.7)	35.1 (47.3/34.1)
SVM-L	33.3 (39.6/28.7)	16.3 (20.9/14.6)	31.9 (42.4/31.6)
SVM-F	32.2 (28.6/36.8)	19.7 (15.0/38.4)	31.0 (30.7/40.0)
MML	32.7 (42.1/26.8)	14.7 (19.4/12.9)	32.2 (51.8/30.5)

(c) JPAT

Table 2: Micro- and macro-averaged, and average labeling F_1-scores (%) with our proposed and conventional methods.

LRM-F and SVM-F outperformed LRM-L and SVM-L in terms of F_M-score for the three datasets, respectively. The training methods of LRM-F and SVM-F were useful to improve the F_M-scores of LRM and SVM, as reported in (Jansche, 2005; Joachims, 2005). The F_μ- and F_L-scores of LRM-F were similar or better than those of LRM-L. LRM-F was effective in improving not only the F_M-score but also the F_μ- and F_L-scores obtained with LRM.

Let us evaluate our model combination methods.

[2] http://svmlight.joachims.org/

[3] http://svmlight.joachims.org/svm_perf.html

MC-F_μ provided better F_μ-scores than LRM-F and SVM-F. The F_M-scores of MC-F_M were similar or better than those of LRM-F and SVM-F. Moreover, MC-F_L outperformed LRM-F and SVM-F in terms of F_L-scores. The binary classifiers designed by using LRM-F and SVM-F were trained to maximize the F_1-score for each category. On the other hand, MC-F_μ, MC-F_M, and MC-F_L classifiers were constructed by combining LRM and SVM with weights estimated to maximize the F_μ-, F_M-, and F_L-scores, respectively. The experimental results show that our training methods for combination weights were useful for obtaining better multi-label classifiers.

MC-F_μ, MC-F_M, and MC-F_L outperformed MML as regards the three F_1-scores for JPAT. However, MML performed better for Reuters than MC-F_μ, MC-F_M, and MC-F_L, and provided a better F_L-score for WIPO. As shown in Table 1, there were more category label combinations for JPAT than for Reuters or WIPO. As a result, there were fewer data samples for the same category label assignment for JPAT. Therefore, MML, which learns the map directly from the feature vectors to the category label assignment vectors, would have been overfitted to the training dataset for JPAT. By contrast, our model combination methods employ binary classifiers for each category, which mitigates such an overfitting problem. Our model combination methods will be useful for *complex* datasets where there are many category label combinations.

5 Conclusion

We proposed a multi-label classifier design method based on model combination. The main idea behind our proposed method is to combine multiple models with weights estimated to maximize evaluation scores such as the micro- and macro-averaged, and average labeling F_1-scores. Using three real text datasets, we confirmed experimentally that our proposed method provided similar or better performance than conventional binary classification approaches to multi-label categorization. We also confirmed that our proposed method was useful for datasets where there were many combinations of category labels. Future work will involve training our multi-label classifier by using labeled and unlabeled samples, which are data samples with and without category label assignment.

References

David A. Bell, J. W. Guan, and Yaxin Bi. 2005. On combining classifier mass functions for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 17(10):1307–1319.

Stanley F. Chen and Ronald Rosenfeld. 1999. A Gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University.

C. J. Fall, A. Törcsvári, K. Benzineb, and G. Karetka. 2003. Automated categorization in the international patent classification. *ACM SIGIR Forum*, 37(1):10–25.

Giorgio Fumera and Fabio Roli. 2005. A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):942–956.

Zoubin Ghahramani and Hyun-Chul Kim. 2003. Bayesian classifier combination. Technical report, Gatsby Computational Neuroscience Unit, University College London.

Makoto Iwayama, Atsushi Fujii, and Noriko Kando. 2007. Overview of classification subtask at NTCIR-6 patent retrieval task. In *Proceedings of the 6th NTCIR Workshop Meeting on Evaluation of Information Access Technologies (NTCIR-6)*, pages 366–372.

Martin Jansche. 2005. Maximum expected F-measure training of logistic regression models. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP2005)*, pages 692–699.

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

Thorsten Joachims. 2005. A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning (ICML'05)*, pages 377–384.

Hideto Kazawa, Tomonori Izumitani, Hirotoshi Taira, and Eisaku Maeda. 2005. Maximal margin labeling for multi-topic text categorization. In *Advances in Neural Information Processing Systems 17*, pages 649–656. MIT Press, Cambridge, MA.

Leah S. Larkey and W. Bruce Croft. 1996. Combining classifiers in text categorization. In *Proceedings of the 19th ACM International Conference on Research and Development in Information Retrieval (SIGIR-96)*, pages 289–297.

Kai Ming Ting and Ian H. Witten. 1999. Issues in stacked generalization. *Journal of Artificial Intelligence Research*, 10:271–289.

David H. Wolpert. 1992. Stacked generalization. *Newral Networks*, 5(2):241–259.

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