Multi-label Patent Classification at NTT Communication Science Laboratories

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

Abstract

We design a multi-label classification system based on the combination of binary classifications for classification subtask at NTCIR-6 Patent Retrieval Task. In our system, we design a binary classifier per F-term that determines the assignment of the F-term to patent documents. Hybrid classifiers are employed as binary classifiers so that the multiple components of patent documents are used effectively. The hybrid classifiers are constructed by combining component generative models with weights based on the maximum entropy principle. Using a test collection of Japanese patent documents, we confirmed that our system provided good ranking of F-terms as regards assigning them to patent documents.

Keywords: *binary classification, hybrid classifier, multiple components, naive Bayes model, maximum entropy principle.*

1 Introduction

Classification subtask at NTCIR-6 Patent Retrieval Task is to develop *multi-label* classification systems that assign multiple F-terms to Japanese patent documents. The F-terms were developed by the Japanese Patent Office, to search effectively for relevant patent documents in the patent database. At present, hundreds of F-terms are defined per theme, which constitutes just one of many technological fields. Therefore, a multi-label classification system dealing with a *massive* number of class labels is required for the classification subtask.

We design a multi-label patent classification system based on the combination of binary classifications [4, 7] to deal with a massive number of class labels. In our formulation, we assume the independence of class labels and design a binary classifier per class label that determines whether or not to assign the class label to a data sample. We employ a *hybrid classifier* [8, 3] as a binary classifier designed for each class label.

Hybrid classifiers can deal with multiple components contained by data samples such as the abstract, claim, and main text in a patent document. To construct hybrid classifiers, we design a generative model for each component and combine all these generative models with weights provided by discriminative training. Namely, each component is modeled on the basis of a generative approach, while the classifier is constructed on the basis of a discriminative approach. Each hybrid classifier constructed per class label provides the probability of assigning a class label to a data sample. Using the probability provided by the hybrid classifiers, we obtain rankings of class labels that should be assigned to data samples.

To enable us to apply the binary hybrid classifiers to patent documents, we employed naive Bayes (NB) models as their component generative models, using a bag of words (BOW) representation for each component. We used five components, namely title (T), author and affiliation names (AA), abstract (AB), claim (C), and main text (MT). Using a test collection given by NTCIR-6 organizers, we show that our multi-label classification system provides good F-term ranking performance for patent documents in terms of mean average precision (MAP).

2 Multi-label Classification System based on Binary Hybrid Classifiers

For classification subtask at NTCIR-6 Patent Retrieval Task, we design a multi-label classification system that provides mapping from a feature vector $\boldsymbol{x}$ to a class label vector $\boldsymbol{y} = (y_1, \ldots, y_k, \ldots, y_K)$. Here, K is the number of class labels. $\boldsymbol{y}$ is a binary bit-vector that represents the assignment of class labels to a feature vector, where $y_k = 1$ if the kth class label is assigned to $\boldsymbol{x}$, and $y_k = 0$ otherwise. In the classification subtask, a feature vector $\boldsymbol{x}$ represents a patent document and a class label k represents an F-term.

In our multi-label classification system, we design K binary classifiers each of which provides the probability $P(y_k = 1|\boldsymbol{x})$ of assigning a kth class label

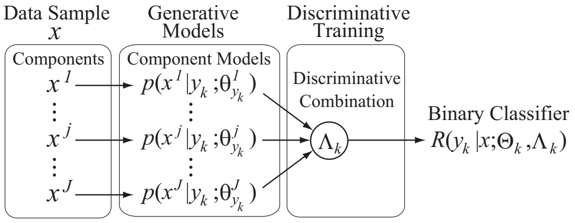
Data Sample
x
Components
x^1
x^j
x^J
Generative Models
Component Models
$p(x^1|y_k;\theta^1_{y_k})$
$p(x^j|y_k;\theta^j_{y_k})$
$p(x^J|y_k;\theta^J_{y_k})$
Discriminative Training
Discriminative Combination
Λ_k
Binary Classifier
$R(y_k|x;\Theta_k,\Lambda_k)$

Figure 1. Outline of binary hybrid classifiers.

to a feature vector $\boldsymbol{x}$, using training data set $D = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$. Then, we rank K class labels for a data sample $\boldsymbol{x}$ based on the values of $\{P(y_k = 1|\boldsymbol{x})\}_{k=1}^K$ provided by the binary classifiers. We employ hybrid classifiers [8, 3] as the binary classifiers. In this section, we describe a method for constructing hybrid classifiers for the binary classification of patent documents.

2.1 Outline of Hybrid Classifiers

A patent document consists of multiple components including a title, claim, and main text. For the binary classification of patent documents, we use hybrid classifiers to deal with multiple components effectively and thus acquire good generalization performance.

Let $\boldsymbol{x} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^j, \ldots, \boldsymbol{x}^J)$ represent a feature vector of a data sample consisting of multiple components, where $\boldsymbol{x}^j$ is a feature vector of the jth component. In the formulation of hybrid classifiers, we first model the probability density of the jth component for data samples assigned with the kth class label, $p(\boldsymbol{x}^j|y_k = 1)$, and also model that for unassigned samples, $p(\boldsymbol{x}^j|y_k = 0)$. Then, we combine these component generative models of $\{p(\boldsymbol{x}^j|y_k)\}_{j=1}^J$ based on the maximum entropy (ME) principle. Namely, we construct a binary hybrid classifier $R(y_k|\boldsymbol{x})$ by combining the component generative models with weights provided by the discriminative training. We outline the binary hybrid classifiers in Fig. 1.

2.2 Component Generative Models

For binary hybrid classifiers, we design individual component generative models, $p(\boldsymbol{x}^j|y_k = 1; \boldsymbol{\theta}^j_{y_k=1})$ and $p(\boldsymbol{x}^j|y_k = 0; \boldsymbol{\theta}^j_{y_k=0})$, for the jth component, where $\boldsymbol{\theta}^j_{y_k=1}$ ($\boldsymbol{\theta}^j_{y_k=0}$) is a model parameter set for the jth component of data samples assigned (not assigned) the kth class label.

We employ naive Bayes (NB) models [7] as component generative models for patent documents, using an independent word-based representation, known as the Bag-of-Words (BOW) representation. Let $\boldsymbol{x}^j = (x^j_1, \ldots, x^j_i, \ldots, x^j_{V_j})$ be the word-frequency vector of the jth component of a data sample, where x^j_i denotes the frequency of the ith word in the jth component and V_j denotes the number of vocabulary words included in the jth component. In the NB model, the probability distribution of $\boldsymbol{x}^j$ for data samples assigned the kth class label is regarded as a multinomial distribution:

$$p(\boldsymbol{x}^j|y_k = 1; \boldsymbol{\theta}^j_{y_k=1}) \;=\; \prod_{i=1}^{V_j} \left(\theta^j_{i,y_k=1}\right)^{x^j_i}. \quad (1)$$

Here, $\theta^j_{i,y_k=1} > 0$ and $\sum_{i=1}^{V_j} \theta^j_{i,y_k=1} = 1$. $\theta^j_{i,y_k=1}$ represents the probability that the ith word appears in the jth component of data samples assigned the kth class label. $p(\boldsymbol{x}^j|y_k = 0; \boldsymbol{\theta}^j_{y_k=0})$ is also given the same distribution form as $p(\boldsymbol{x}^j|y_k = 1; \boldsymbol{\theta}^j_{y_k=1})$.

Model parameter set $\Theta^j_k = \{\boldsymbol{\theta}^j_{y_k}\}_{j,y_k}$ is computed by maximizing the posterior $p(\Theta^j_k|D)$ (MAP estimation). According to the Bayes rule, $p(\Theta^j_k|D) \propto p(D|\Theta^j_k)p(\Theta^j_k)$, the objective function for the MAP estimation of component generative models is given by

$$\begin{aligned} J(\Theta^j_k) &= \sum_{n=1}^N \log p(\boldsymbol{x}^j_n|y_{nk}; \boldsymbol{\theta}^j_{y_{nk}}) \\ &+ \sum_{y_k=0}^1 \log p(\boldsymbol{\theta}^j_{y_k}). \end{aligned} \quad (2)$$

Here, $p(\boldsymbol{\theta}^j_{y_k})$ is a prior probability distribution of $\boldsymbol{\theta}^j_{y_k}$. We use the following Dirichlet prior

$$p(\boldsymbol{\theta}^j_{y_k}) \;\propto\; \prod_{i=1}^{V_j} \left(\theta^j_{i,y_k}\right)^{\xi^j_{y_k}-1}, \quad (3)$$

where $\xi^j_{y_k} (> 1)$ represents a hyperparameter. Using feature vectors $\{\boldsymbol{x}^j_n\}_{n,j}$, we compute the estimate of θ^j_{i,y_k} to maximize the objective function $J(\Theta^j_k)$ as

$$\hat{\theta}^j_{i,y_k} = \frac{\sum_{n=1}^N I_{y_k}(y_{nk}) x^j_{ni} + \xi^j_{y_k} - 1}{\sum_{i=1}^{V_j} \sum_{n=1}^N I_{y_k}(y_{nk}) x^j_{ni} + V_j(\xi^j_{y_k} - 1)}. \quad (4)$$

Here, $I_{y_k}(y_{nk})$ is an indicator function where $I_{y_k}(y_{nk}) = 1$ if $y_{nk} = y_k$, and $I_{y_k}(y_{nk}) = 0$ otherwise. For our system, we used normalized feature vectors such as $\sum_{i=1}^{V_j} x^j_{ni} = 1$.

2.3 Hybrid Classifier Construction

We provide the probability of assigning a kth class label to a data sample $\boldsymbol{x}$, $R(y_k = 1|\boldsymbol{x})$, based on the weighted combination of the component generative models in a discriminative manner. More specifically, we design the distribution of $R(y_k = 1|\boldsymbol{x})$ by combining component generative models based on the maximum entropy (ME) principle [1].

The ME principle is a framework for obtaining a probability distribution, which prefers the most uniform models that satisfy any given constraints. Providing the constraints for component generative models and the probability of assigning the kth class label as shown in [3], we can obtain a probability distribution according the ME principle as

$$\begin{aligned} & R(y_k = 1|\boldsymbol{x}; \hat{\Theta}_k, \Lambda_k) \\ = & \frac{1}{1 + \exp\left\{-\mu_k - \sum_{j=1}^{J} \lambda_k^j f_k^j(\boldsymbol{x})\right\}}, \end{aligned} \tag{5}$$

where $f_k^j(\boldsymbol{x}) = \log\{p(\boldsymbol{x}^j|y_k = 1; \hat{\boldsymbol{\theta}}^j_{y_k=1})/p(\boldsymbol{x}^j|y_k = 0; \hat{\boldsymbol{\theta}}^j_{y_k=0})\}$, and $\Lambda_k = \{\{\lambda_k^j\}_{j=1}^J, \mu_k\}$ is a set of Lagrange multipliers. λ_k^j provides the combination weight of the jth component, and μ_k provides an assignment bias for the kth class label. The probability of not assigning the kth class label is provided as $R(y_k = 0|\boldsymbol{x}; \hat{\Theta}_k, \Lambda_k) = 1 - R(y_k = 1|\boldsymbol{x}; \hat{\Theta}_k, \Lambda_k)$. The distribution $R(y_k|\boldsymbol{x}; \hat{\Theta}_k, \Lambda_k)$ gives us the formulation of a binary hybrid classifier based on a discriminative combination of component generative models.

According to the ME principle, the solution of Λ_k in Eq. (5) is equal to the Λ_k value that maximizes the log likelihood for $R(y_k|\boldsymbol{x}; \hat{\Theta}_k, \Lambda_k)$ of training samples $(\boldsymbol{x}_n, y_{nk}) \in D$ [1, 6]. However, D is also used to estimate Θ_k. Using the same training samples for Λ_k as Θ_k may lead to a bias estimation of Λ_k. Thus, a leave-one-out cross-validation of the training samples is used for estimating Λ_k [8]. Let $\hat{\Theta}_k^{(-n)}$ be the generative model parameter estimated by using all the training samples except $(\boldsymbol{x}_n, y_{nk})$. The objective function of Λ_k then becomes

$$\begin{aligned} J(\Lambda_k) &= \sum_{n=1}^{N} \log R(y_{nk}|\boldsymbol{x}_n; \hat{\Theta}_k^{(-n)}, \Lambda_k) \\ &+ \log p(\Lambda_k), \end{aligned} \tag{6}$$

where $p(\Lambda_k)$ is a prior probability distribution of Λ_k. We use a Gaussian prior [2] as

$$\begin{aligned} p(\Lambda_k) &\propto \prod_{j=1}^{J} \exp\left\{-\frac{(\lambda_k^j - 1)^2}{2\sigma^2}\right\} \\ &\times \exp\left(-\frac{\mu_k^2}{2\rho^2}\right), \end{aligned} \tag{7}$$

where σ and ρ are hyperparameters. We can compute an estimate of Λ_k to maximize $J(\Lambda_k)$ by using the L-BFGS algorithm [5], which is a quasi-Newton method. We summarize the algorithm for estimating these model parameters in Fig. 2.

```
   Given training sample set: $D = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$
1. Compute $\hat{\Theta}_k$ using Eq. (2).
2. Compute $\hat{\Theta}_k^{(-n)}$, $\forall n$ by applying Eq. (2) to training
   samples except $(\boldsymbol{x}_n, y_{nk})$.
3. Compute $\Lambda_k$ using Eq. (6) under fixed $\hat{\Theta}_k^{(-n)}$.
4. Output a classifier $R(y_k|\boldsymbol{x}; \hat{\Theta}_k, \hat{\Lambda}_k)$.
```

Figure 2. Algorithm for estimating model parameters of binary hybrid classifiers.

3 Experiments

3.1 Test Collections

For classification subtask at NTCIR-6 Patent Retrieval Task, Japanese patent documents submitted to the Japanese patent office from 1993 to 1999 were given us for an evaluation of the multi-label classification systems that assign F-terms to Japanese patent documents. The 21606 patent documents that related to one of 108 themes and were submitted in 1998 and 1999 were selected as text data by the NTCIR-6 organizers. We designed a multi-label classification system per theme and trained it by using patent documents submitted from 1993 to 1997.

To apply binary hybrid classifiers, we extracted five components, *title* (T), *author and affiliation names* (AA), *abstract* (AB), *claim* (C), and *main text* (MT), from the patent documents. We extracted nouns, verbs, and adjectives from each component by using MeCab[1] and obtained word-frequency vectors as feature vectors of components. Vocabulary words included in only one patent document were removed from the feature vectors.

3.2 Evaluation Results

With the standard TREC-style evaluation method, we calculated recall and precision in an F-term ranking for each patent document, and we summarized these scores in terms of the mean average precision (MAP). We also collected confident class labels assigned to patent documents by classifiers, and then calculated the F-measure, recall, and precision of confidence for each patent document. These evaluation scores were averaged over all patent documents provided as test data.

Table 1 shows the recall-precision curve and MAP obtained with our system and the top two other teams' systems submitted to the NTCIR-6 organizers. These scores were extracted from the results returned by the organizers. Our system provided better MAP than the other systems. This result indicates that our system

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

Table 1. recall-precision curve and MAP with our and top two other systems.

Recall	Precision		
	Our System	SVM-based	kNN-based
0.0	0.8219	**0.8251**	0.7864
0.1	0.7965	**0.8013**	0.7594
0.2	0.7233	**0.7304**	0.6828
0.3	0.6365	**0.6432**	0.5969
0.4	0.5655	**0.5684**	0.5273
0.5	**0.5143**	0.5142	0.4770
0.6	**0.4272**	0.4180	0.3934
0.7	**0.3520**	0.3335	0.3244
0.8	**0.2961**	0.2702	0.2711
0.9	**0.2291**	0.1989	0.2054
1.0	**0.1979**	0.1679	0.1723
MAP	**0.4852**	0.4779	0.4518

Table 2. F-measure, recall, and precision of confidence with our and top two other systems.

System	F-measure	Recall	Precision
Ours (Fixed Threshold)	0.3289	0.2705	**0.5874**
Ours (Tuned Threshold)	0.4037	0.4846	0.4083
SVM-based	**0.4125**	0.4904	0.4075
kNN-based	0.3840	**0.5668**	0.3354

provided better F-term ranking for patent documents. The precision of our system was better when the recall was high. We confirmed that our system was useful especially when required to extract every F-term assigned to a patent document.

Table 2 shows F-measure, recall, and precision of confidence obtained with our system and the top two other teams' systems submitted to the NTCIR-6 organizers. In our system, two thresholds for $\{R(y_k = 1|\boldsymbol{x}; \hat{\Theta}_k, \hat{\Lambda}_k)\}_{k=1}^{K}$ provided by binary classifiers were used to assign F-terms to patent documents. First, we used a fixed threshold value of 0.5 for the assignment. Namely, we assigned the kth F-term to patent document $\boldsymbol{x}$ when $R(y_k = 1|\boldsymbol{x}; \hat{\Theta}_k, \hat{\Lambda}_k) \geq R(y_k = 0|\boldsymbol{x}; \hat{\Theta}_k, \hat{\Lambda}_k)$. Second, we tuned a threshold value per theme to maximize the leave-one-out cross-validation F-measure of the training data. As shown in table 2, the recall of our system was small when the threshold value was fixed at 0.5. This result indicates that our system with the fixed threshold value did not assign many of the F-terms that should be assigned to patent documents. However, the recall and F-measure for our system were improved by tuning the threshold value. We confirmed that tuning the threshold values of our system was useful in practice as regards obtaining better F-term assignment performance in F-measure.

4 Conclusion

We designed a multi-label classification system based on binary classifiers for classification subtask at NTCIR-6 Patent Retrieval Task. Each binary classifier was used for determining the assignment of an F-term to patent documents. As the binary classifiers, we employed hybrid classifiers dealing with multiple components of patent documents. The hybrid classifiers were constructed by combining component generative models with weights in a discriminative manner. Using a test collection provided by the NTCIR-6 organizers, we confirmed experimentally that our system provided good F-term ranking performance for patent documents in terms of mean average precision (MAP). We also confirmed that tuning the threshold value per theme for our system improved the F-term assignment performance in F-measure. Future work will involve training the multi-label classifier system with labeled and unlabeled (test) data, which are data with and without class labels.

References

[1] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] S. F. Chen and R. Rosenfeld. A Gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University, 1999.

[3] A. Fujino, N. Ueda, and K. Saito. A hybrid generative/discriminative approach to text classification with additional information. *Information Processing and Management*, 43(2):379–392, 2007.

[4] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML '98)*, pages 137–142, 1998.

[5] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Programming, Ser. B*, 45(3):503–528, 1989.

[6] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[7] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.

[8] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. Classification with hybrid generative/discriminative models. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.