Multiple Strategies for NTCIR-8 Patent Mining at BCMI

Gang Jin, Qi Kong, Jian Zhang, Xiaolin Wang, Cong Hui, Hai Zhao, and Bao-Liang Lu*
Center for Brain-Like Computing and Machine Intelligence
Department of Computer Science and Engineering
MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems
Shanghai Jiao Tong University 800 Dong Chuan Rd., Shanghai 200240, China
{zhaohai, blu}@cs.sjtu.edu.cn

ABSTRACT

This paper describes our system for the NTCIR-8 patent mining task which creates technical to map a research papers into IPC taxonomy. Our focus was upon the Japanese patent collection, and we applied three kinds of methods. One is based on the K-NN algorithm, we extended its similarity and ranking policy. The second is a hierarchical SVMS tree, that every node of the tree is a SVM classifier. At last we constructed a general framework called M^3 for handling huge training data set, based on the idea of divide-and-conquer. The evaluation results indicated that the extended K-NN has a better performance on both accuracy and time-consuming. And a combination strategy of re-ranking could improve the result slightly.

Categories and Subject Descriptors

H.3 [**Information Storage And Retrieval**]: Miscellaneous

General Terms

Algorithms and Experimentation

Keywords

Patent Mining, K-NN, M^3, Hierarchical SVMs

1. INTRODUCTION

The NTCIR-8 [10, 11] patent mining task aims to develop techniques to map a research paper into IPC taxonomy. The task is a standard multi-label classification issue. In the multi-label classification field, there are two major jobs: multi-label classification(MLC) and label ranking(LR), and the methods of processing the task grouped into two categories proposed in [13], one is problem transformation, and the other is algorithm adaptation. The first group of methods transform the learning task into one or more single-label classification tasks and the second group extend specific learning algorithm to handle multi-label data directly.

Based on the two groups of methods, we have developed three approaches for the Japanese patent mining subtask. And we think one of the barriers of this task is that the writing style of the research query is different from the ones used in the patent documents, and the other is the huge scale of the patent training data set. So our system focus on the feature space and the performance of training part.

The rest of this paper is organized as follows. Section 2 shows an overview of our system. And in section 3 and section 4, we describe our system in detail. Section 5 illustrates the experiments and submitted results. Section 6 is the discussion and section 7 concludes the paper.

*Corresponding Author

2. SYSTEM OVERVIEW

The patent mining task is treated as a label rangking problem in our system, and some specific algorithms are extended to produce the final ranked label directly. Our system has three basic parts illustrated in Fig 1.

The preprocess part is to convert the training documents and test samples into the vector of VSM, then we trained the classifier on the data set and gave a predict IPC list through the ranking module based on results of the predict model.

3. DATA AND PREPROCESSING

The training set we used is the Japanese patent documents from 1993 to 2002, each patent is generally assigned to one or multiple IPC codes that indicate the related technical fields.

Table 1: Statistics for Japanese patents

Data set	#Instatnces	#Attributes	#Labels
Japanese Patent	3496137	1037871	50042

The preprocess part includes three steps, first is parser. We used ChaSen to parsed the research paper and patent document. ChaSen is a morphological parser for the Japanese language. This tool for analyzing morphemes was developed at the Matsumoto laboratory, Nara Institute of Science and Technology. All of the terms have been chosen.

Second considered the different writting styles of research paper and patent document, we constructed kinds of feature space for both training set and test set. As a structured patent document, each patent has four fields: title, abstract,

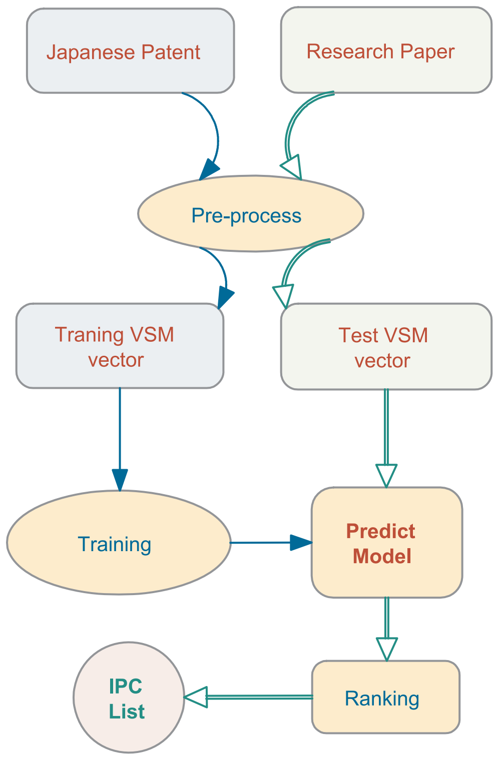

Figure 1: System Processing Flow

claim and description, and we thus expect that the individual field may has different impact for the task. So we select from these fields and combine them by some weights based on the experiments. Table 2 is the combination we chose for training set. There are four different versions for every patent document.

Table 2: Structure of the training patent data set

combination	title	abstract	claim	description
whole+title	1	1	1	1
whole+3title	3	1	1	1
part+title	1	1	0	0
part+3title	3	1	0	0

On the other hand, for the test set, a K-NN is run to find the most similar documents to the research paper and the research paper is re-constructed by the words of the neighbors we found. That is to say, we re-fixed the position of the research paper in the patent documents space. The idea is shows in Fig 2

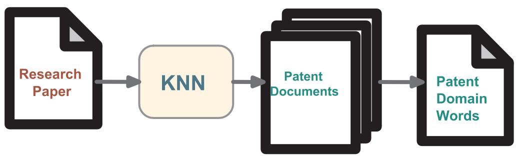

Figure 2: Re-construction of the Research Paper

The third step is indexing. We tried three different methods to index. The first method for indexing is

$$TFIDF : a_{ij} = f_{ij} \times log(\frac{N}{n_i}) \tag{1}$$

The lengths of documents are different. In order to reduce the influence of this different, we normalize it as following.

$$TFC : a_{ij} = \frac{f_{ij} \times log(\frac{N}{n_i})}{\sqrt[2]{\sum_{k=1}^{M}(f_{kj} \times log(\frac{N}{n_k}))^2}} \tag{2}$$

Noticed that the differences in word frequencies, and some words with low word frequencies are also very important roles in the classification, we tried the third method as

$$ITC : a_{ij} = \frac{log(f_{ij}+1) \times log(\frac{N}{n_i})}{\sqrt[2]{\sum_{k=1}^{M}(log(f_{kj}+1) \times log(\frac{N}{n_k}))^2}} \tag{3}$$

where f_{ij} is the frequency of $word_i$ in doc_j, N is the number of documents, M is the number of the words in all documents and n_i is number of documents which include the $word_i$.

4. CLASSIFIER MODEL

For characteristic of the data set we used, the task can be summarized as a large-scale imbalance multi-label text classification problem. The major challenges are that how to deal with the large-scale imbalance patent documents and how to handle the huge IPC taxonomy. Considered the challenges, three kinds of methods are proposed in our system.

4.1 K-NN based method

For the challenges mentioned above, the great deal of memory space and time cost make sophisticated machine learning method not worked well. In contrast, the K-NN method has the nature trait to deal with these challenges. Its idea is only based on the extracting similar documents and no training process is required. Besides, the K-NN' neighbors are a ranking and it's easily applied on the IPC codes.

So we developed some similarity distance functions and ranking methods.

4.1.1 Similarity

There are two kinds of similarity used in our system, one is the traditional similarity and the other is an IR similarity.

The traditional similarity commonly used in K-NN contains Cosine, Euclid and Set distance functions. We chose the Cosine distance function based on our experiments before.

Given two document vectors $\vec{v_1}$ and $\vec{v_2}$, the similarity is computed as:

$$Sim_{cosine}(\vec{v_1}, \vec{v_2}) = \frac{\vec{v_1} \cdot \vec{v_1}}{\| \vec{v_1} \| \| \vec{v_2} \|} \tag{4}$$

In our method, each feature term t_j of a document vector $\vec{v_i} = (w_{i,1}, w_{i,2}, \cdots, w_{i,m})$ is weighted by the indexing methods we mentioned in section 3.

And the IR similarity we use is $BM25$ [2] which is widely used by search engines in information retrieval. It is based on the probabilistic retrieval framework. In the $BM25$ weighting scheme, the input document is treated as a query.

Given a query Q, containing keywords $q_1, \cdots, q_n$, the $BM25$ score of a document D is:

$$BM25(D,Q) = \sum_{i=1}^{n} w_i . \frac{f(q_i, D).(k_1+1)}{f(q_i, D) + k_1.(1 - b + b.\frac{|D|}{avgdl})} \quad (5)$$

where $f(q_i, D)$ is q_i's term frequency in the document D, $|D|$ is the length of the document D in words, w_i is the IDF value of (q_i) and $avgdl$ is the average document length in the text collection from which documents are drawn. k_1 and b are free parameters, usually chosen as $k_1 = 2.0$ and $b = 0.75$. w_i is the IDF weight of the query term q_i, which is calculated as follows,

$$w_i = log(\frac{N - df_i + 0.5}{df_i + 0.5}) \quad (6)$$

where the df_i is the number of documents containing i.

4.1.2 Ranking

First the system extracts the top-k documents with the highest similarities through the K-NN method. After that the system produces the IPC score list by the ranking scheme.

We choose two ranking schemes based on the experiments before.

The first is simple vote scheme, in this method, score is calculated by summing up the similarities of all the extracted documents containing IPC code, as follows,

$$Score_{vote}(c) = \sum_{i=1}^{k} occurs(c, d_i). Sim(q, d_i) \quad (7)$$

where c is the IPC code and the occurs fucntion is defined as follows

$$occurs(c,d) = \begin{cases} 1 & \text{if ipc code c occurs in document d} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

and q is the input document, $Sim(q, d_i)$ is the similarity between q and d_i.

The second method is listweak, which is to emphasize the patents ranked in the frontier part of list.

$$Score_{listweak}(c) = \sum_{i=1}^{k} occurs(c, d_i). Sim(q, d_i). r_1^i \quad (9)$$

where r_1 is a parameter ranging in (0,1) and r_1^i can be regarded as a penalty whose ranks are lower.

4.2 Hierarchical SVMS

In this method, a hierarchical network of support vector machines(SVMs) is built, the structure of which is isomorphism with IPC (Fig. 3). One SVM is trained for each internal node of IPC by the training documents belonging to that node [1, 5]. This method is also called top-down method [12, 16].

To classify a test instance, it is first sent to the root classifier, which predicts its scores on section labels. The top-n sections are accepted, where n is a predefined number. Then the instance is sent to the classifiers of accepted sections, which further predict the scores on class labels. In this way the instance walk down the network of SVMs until it reaches the buttom levels. The number of accepted labels, which we set in this task, are (2,3,3,5,10) at levels from section to subgroup. The final rank of candidate subgroup labels are based on the sum of scores at all levels.

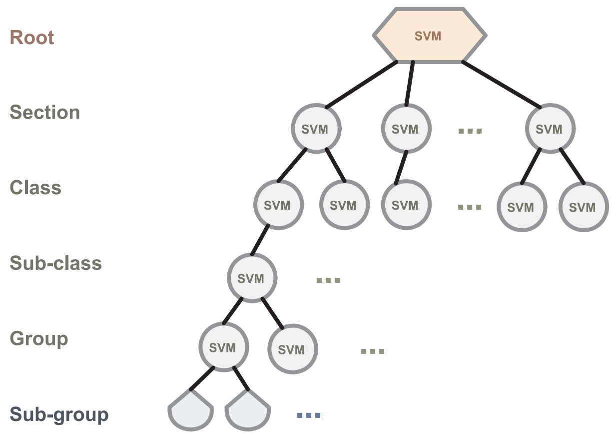

Figure 3: Hierarchical SVMS

4.3 M^3 framework

M^3 is short for Min-Max Modular Network proposed in [7, 6]. It's a framework that is capable of solving large-scale pattern classification problems in a parallel way based on the conquer-and-divide idea.

The framework has been used in many applications in [4, 14, 15, 9, 17, 3, 8].

M^3 include two major parts described as following.

4.3.1 Task decomposition

Let $\mathcal{T}$ be the training set of a K-class classification problem and the K classes are represented by $C_1, C_2, \ldots, C_K$, respectively.

$$\mathcal{T} = \{(X_l, Y_l)\}_{l=1}^{L} \quad (10)$$

where $X_l \in R^d$ is the input vector, $Y_l \in R^K$ is the desired output, and L is the number of training data. Suppose the K training input sets, $\mathcal{X}_1, \ldots, \mathcal{X}_K$ are expressed as

$$\mathcal{X}_i = \{X_l^i\}_{l=1}^{L_i} \quad for \quad i = 1, \ldots, K \quad (11)$$

where L_i is the number of training inputs in class C_i, X_l^i is the l-th sample belong to class C_i and all of $X_l^{(i)} \in \mathcal{X}_i$ have the same desired outputs and $\sum i = 1^K L_i = L$. According to the m3 network, a K-class problem can be divided into $K \times (K-1)$ two-class problem that are trained independently, each of which is given by,

$$\mathcal{T}_{ij} = \{(X_l^{(i)}, +1)\}_{l=1}^{L_i} \cup \{(X_l^{(j)}, -1)\}_{l=1}^{L_j}$$
$$for \quad i = 1, \ldots, K-1 \quad and \quad j = i+1, \ldots, K \quad (12)$$

If these two-class problems are still in large-scale or imbalanced, the can be further decomposed into relatively smaller two-class problems.

Assume that the input set $\mathcal{X}_i$ is further partitioned into N_i subsets in the form of

$$\mathcal{X}_{ij} = \{X_l^{(ij)}\}_{l=1}^{L_i^{(j)}} \quad for \quad j = 1, \ldots, N_i \quad (13)$$

where L_i^j is the number of training inputs included in $\mathcal{X}_{ij}$ and $\cup_j^{N_i} \mathcal{X}_{ij} = \mathcal{X}_i$. After dividing the training input set $\mathcal{X}_i$

into N_i subsets $\mathcal{X}_{ij}$, the training set for each of the smaller and simpler two class problem can be given by

$$\mathcal{T}_{ij}^{(u,v)} = \{_l^{(iu),+1}\}_{l=1}^{L_i^{(u)}} \cup \{X_l^{(iu),-1}\}_{l=1}^{L_j^{(v)}} \quad for \quad u = 1, \ldots, N_i, v = 1, \ldots, N_j, \quad i = 1, \ldots, K-1 \quad and \quad j = i+1, \ldots, K \tag{14}$$

where $X_l^{(iu)} \in \mathcal{X}_{iu}$ and $X_l^{(jv)} \in \mathcal{X}_{jv}$ are the input vectors belonging to class C_i and class C_j, respectively, $\sum_{u=1}^{N_i} L_i^{(u)}$ and $\sum_{v=1}^{N_j} L_j^{(v)}$.

4.3.2 Module combination

After these smaller two-class problems $\mathcal{T}_{ij}^{(u,v)}$ have been trained, they will be integrated according to the minimization principle and maximization principle, respectively, as follows:

$$\mathcal{T}_{ij}^{u}(x) = \min_{v=1}^{N_j} \mathcal{T}_{ij}^{(u,v)}(x) \tag{15}$$

$$\mathcal{T}_{ij}(x) = \max_{u=1}^{N_i} \mathcal{T}_{ij}^{(u)}(x) \tag{16}$$

A linear-M^3 network is illustrated in Fig 4

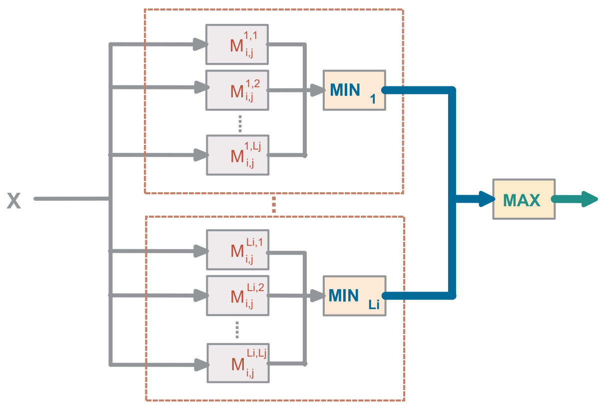

Figure 4: The M^3 network consists of $L_i \times L_j$ individual network modules, L_i MIN units, and one MAX unit

In our system, we has implemented the framework of M^3, and in the task decomposition part, we chose the one-vs-one scheme to transform multi-label problem to single-label task, and if the single-label problem is still imblanced, we use a random split scheme to blance the task by setting each training set less than 2000. And the classifer we used is a normal K-NN algorithm.

4.4 Re-ranking

In this module, we combine h different IPC codes lists outputs from different classification schemes. We calculate the score of each IPC code again according the equation as follows,

$$Score_{combination}(c) = \sum_{i=1}^{h} \frac{\lambda_i}{rankinlist(c, l_i)} \tag{17}$$

where rankinlist(c,l_i) stands for the rank of IPC code c in the code list l_i and λ_i denotes the weight for a list. Here we adopt the MAP value of different classification schemes obtained on the dry run dataset as weight.

5. EVALUATION

The measure method used in the task is Mean Average Precision(MAP), which is the most frequently used summary measure of a ranked retrieval run, computed as,

$$AveP = \frac{\sum_{r=1}^{N}(P(r) \times rel(r))}{number \quad of \quad relevatnt \quad documents} \tag{18}$$

where r is the rank, N is the number retrieved, $rel()$ is a binary function on the relevance of a given rank, and $P(r)$ is precision at a given cut-off rank, defined as follows,

$$P(r) = \frac{|T|}{r} \tag{19}$$

where T means the relevant retrieved documents of rank r or less.

The first set of experiments on the dry-run data set is carried out to find the best combination of the options in the data preprocess part. Then based on the results of dry-run, we submitted some results for the formal-run. At last, we tried some additional experiments and got a better MAP value.

5.1 Dry Run

The traning set used in the dry-run experiments is the Japanese patent documents, and the test samples are the research papers provided by the organizers. There are 50 test samples in total and each sample includes title and abstract.

The first experiment we set up is to test which ranking method is better, we use the K-NN classifier and run the experiments with K from 100 to 9000. Fig 5 shows that the listweak ranking method got a better result, and the simple vote strategy can a get maximal value when K is 30.

Then the next experiment shows the different effect of the indexing methods. From Fig 6, it's concluded that the ITC indexing has a better performance.

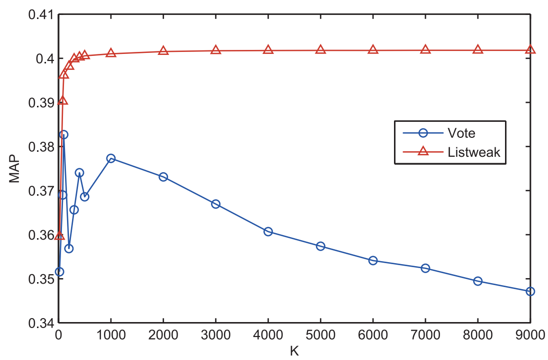

Figure 5: Ranking method Experiment Test

And the third experiment indicates the impact of combination of different fields in the patent document.

We see that the part+3title combination of the patent fields could give more information for the patent topic. Claim and description make little contribution.

The replace method, which convert the research paper words into patent words we methioned in the section 2, didn't make any progress. The space conversion is not accurate.

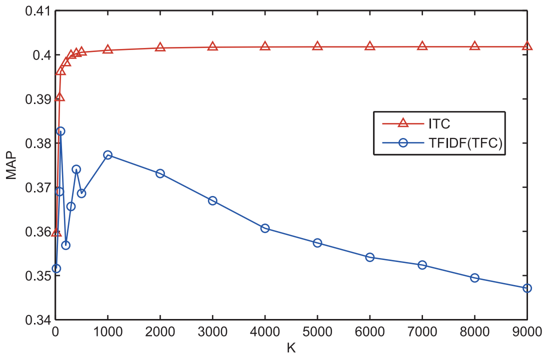

Figure 6: Indexing method Experiment Test

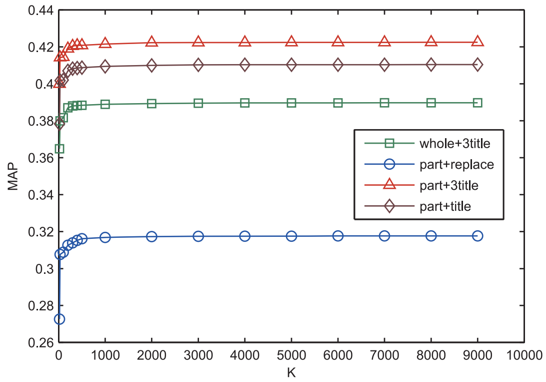

Figure 7: Combination of different fileds of patent

At last, an experiment is set to evluate the similarity for the K-NN alrightm. The $BM25$ similarity got a much better MAP value.

Table 3: Similarity Experiment

K	20	80	100
Cosine	0.399937	0.414275	0.414511
BM25	0.498591	0.496554	0.496884

5.2 Formal Run

Based on the dry-run results, we submitted 10 results for the formal-run.The first two are come from the Hierarchical SVMs methods, the second group from 3 to 7 is based on the K-NN algorithm, 8 and 9 is the result of re-ranking module, and at last, the result is from the M^3 framework.

The table shows that the re-ranking module improves the MAP value by comibining some of different classifiers. And the K-NN algorithm has a better performance.

5.3 Additional

The first try is to use the whole patent document in the BM25 similarity method. And anther proposed is that, the patent's IPC code we used in experiment is only the primary IPC code, now we collect all the IPC codes belong to the patent, we found it can improve the MAP value much.

At last we found an encoding error of the English word,

Table 4: Formal Run Results

RunID	Description	MAP
SG_GBCMI1	Hierarchical SVM, part+3title	0.2994
SG_GBCMI2	Hierarchical SVM, part+title	0.2884
SG_GBCMI3	K-NN_BM25 part	0.3131
SG_GBCMI4	K-NN_COS, whole+title	0.2716
SG_GBCMI5	K-NN_COS, part+title	0.2883
SG_GBCMI6	K-NN_COS, part+3title	0.2788
SG_GBCMI7	K-NN_COS, part+replace	0.2705
SG_GBCMI8	re-ranking(3,4,5)	0.2914
SG_GBCMI9	re-ranking(3,4)	0.3414
SG_GBCMI10	M^3, part+3title	0.3131

which in the test document is normal English words in ASCII but in our training documents it's encoded in Shift_JIS, illustrated in Tabel 5, so we translate them into the same one, the result is improved.

Table 5: Error Encoding Samples

Test Set	Traning Set
CDMA	Ｃ Ｄ Ｍ Ａ
Fiber	Ｆ ｉ ｂ ｅ ｒ
GPS	Ｇ Ｐ Ｓ
GSM	Ｇ Ｓ Ｍ
GByte	Ｇ Ｂ ｙ ｔ ｅ

The following table shows the result based the SG_GBCMI3 method.

Table 6: Additional Result based on SG_GBCM3 method

RunID	description	MAP
SG_GBCMI3	baseline	0.3131
Additional-1	all fileds [abstract only]	0.3433
Additional-2	all IPC codes [primary IPC code]	0.4061
Additional-3	encoding error [English word]	0.4240

6. DISCUSSION

6.1 Data prepross

The data prepross plays an important role in the task. In the dry run experiments, the combinations of different fields of patent documents and the indexing methods impact the result much. The part+3title combination of the patent fields could give more information for the patent topic. Claim and description make little contribution. The space conversion convertting the research paper words into patent words was not accurate and didn't make any progress. And ITC indexing has a better performance.

How to characterise the patent documents is worth studying in our future work.

6.2 Classifier

Based on the experiments we set up, we found that the K-NN algorithm has a much better performance than the other two. Both Hierarchical SVMs and M^3 are suffered from the huge IPC codes, so we think the one-vs-rest or one-vs-one scheme is not suitable for the real world multi-label problem. In our experiments, the parallel K-NN may take twenty minites for 50 test samples and get a better MAP value but the other two may take several hours for training and testing.

So how to improve the performance of the Hierarchical SVMs, M^3 or other machine learning method for the these real world problems is need to study further.

6.3 Re-ranking

The best result is come from the re-ranking scheme, which re-ranked based on the different predicted IPC lists from several methods. One major reason why it works we think is that it combine the advantages of the different methods, just as an expert system, every method is an expert and re-raking module here is the vote scheme in the expert system. So the combination of different methods including their feature space, classifier or other characteristic may improve the MAP value in an IR problem.

7. CONCLUSIONS AND FUTURE WORK

We participated in the NTCIR-8 patent mining task, Particularly, our focus is on the Japanese patent subtask and we implemented a system to classify a research paper into the IPC taxonomy according to the patents database.

Our group proposed three kinds of approaches for the task, one is based on the K-NN algrigthm, we tried several different similarities and ranking functions. Second we implemented a decision tree which every node of the tree is a SVM classifier. Then we constructed a framework called M^3 to deal with the huge data set classification problem. The evaluation result shows that the K-NN approach got a better perfomance.

In future, we plan to add more options to the M^3 framework. We also want to design some effective approaches to handle the issue raised by the different writting styles between the patent documents and research papers.

Acknowledgement

This work was partially supported by the National Natural Science Foundation of China (Grant No. 60903119, Grant No. 60773090 and Grant No. 90820018), the National Basic Research Program of China (Grant No. 2009CB320901), and the National High-Tech Research Program of China (Grant No.2008AA02Z315).

References

[1] M. Ceci and D. Malerba. Classifying web documents in a hierarchy of categories: a comprehensive study. *In Journal of* Intelligent Information Systems, 28:37–38, 2007.

[2] S. E.Robertoson, S. Walker, and M. Hancock-Beaulieu. Okapi at trec-7. *In Proceedings of the Seventh Text Retrieval Conference. Gaithersbrg, USA*, pages 253–264, November 1998.

[3] H. C. Lian and B. L. Lu. Age estimation using a min-max modular support vector machine. *in Proc. ICONIP 2005, 2005, Taipei.*, 2005.

[4] F. Y. Liu, K. Wu, H. Zhao, and B. L. Lu. Fast text categorization with min-max modular support vector machines. *Proc. IEEE International Joint Conference on Neural Networks, Montreal, Quebec, Canada , July 31-Aug*, 4:570–575, 2005.

[5] T. Liu, Y. Yang, Z. Wan, H., C. H.J., Z., and W. Ma. Support vector machines classification with a very large-scale taxonomy. *In* SIGKDD Explorations, 7(1):36–43, 2005.

[6] B. L. Lu and M. Ito. Task decomposition and module combination based on class relations: a modular neural network for pattern classification. *IEEE trans. Neural Networks*, 10:1244–1256, 1999.

[7] B. L. Lu and M. Ito. Task decomposition baed on class relations: a modular neural network arcitecture for pattern classification. *Lecture Notes in Computer Science*, 1240:330–339, 1999.

[8] B. L. Lu, J. Shin, and M. Ichikawa. Massively parallel classification of single-trial eeg signals using a min-max modular neural network. *IEEE Trans, Biomedical Engineering*, 51:551–558, 2004.

[9] J. Luo and B. L. Lu. Gender recognition using a min-max modular support vector machine with equal clustering. *ISNN 2006. LNC*, 3972:210–215, 2006.

[10] H. Nanba, A. Fujii, M. Iwayama, and T. Hashimoto. Overview of the patent mining task at the ntcir-8 workshop.

[11] H. Nanba, A. Fujii, M. Iwayama, and T. Hashimoto. Proceedings of the 8th ntcir workshop meeting on evaluation of information access technologies: Information retrieval, question, answering and cross-lingual information access, 2010.

[12] A. Sun and E.-P. Lim. Hierarchical text classification and evaluation. *In Proc. of* IEEE ICDM, pages 521–528, 2001.

[13] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook, O. Maimon, L. Rokach (Ed.), Springer, 2nd edition*, 2009.

[14] Y. M. Wen, B. L. Lu, and H. Zhao. Equal clustering makes min-max modular support vector machine more efficient. *Proc. 12th Interna- tional Conference on Neural Information Processing, Taipei, Taiwan, Oct. 30-Nov.*, 2:77–82, 2005.

[15] Y. M. Wen, B. L. Lu, and H. Zhao. Multi-view gender classification using local binary patterns and support vector machines. *ISNN 2006. LNCS*, 3972:202–209, 2006.

[16] G. Xue, D. Xing, Q. Yang, and Y. Yu. Deep classification in large-scale text hierarchies. *In Proc. of* ACM SIGIR, pages 619–626, 2008.

[17] Y. Yang and B. L. Lu. Prediction of protein subcellular multi-locations with a min-max modular support vector machine. *in Proc. International Symposium on Neural Networks 2006, Chengdu, China, May, 2006*, pages 667–673, 2006.