Efficient Classification Method for Complex Biological Literature Using Text and Data Mining Combination

Yun Jeong Choi and Seung Soo Park

Department of Computer Science & Engineering,
Ewha Womans University, Seoul 127-150, Korea
cris@ewhain.net, sspark@ewha.ac.kr

Abstract. Recently, as the size of genetic knowledge grows faster, the automated analysis and systemization into high-throughput database has become a hot issue. In bioinformatics area, one of the essential tasks is to recognize and identify genomic entities and discover their relations from various sources. Generally, biological literatures containing ambiguous entities, are laid by decision boundaries. The purpose of this paper is to design and implement a classification system for improving performance in identifying entity problems. The system is based on reinforcement training and post-processing method and supplemented by data mining algorithms to enhance its performance. For experiments, we add some intentional noises to training data for testing the robustness and stability. The result shows significantly improved stability on training errors.

1 Introduction

As the advanced computational technology and systems have been developed, the amount of new biomedical knowledge and their scientific literature has been increased exponentially. Consequently, the automated analysis and systemization in high- throughput system has become a hot issue. Most of biological and medical literatures have been published online, such as journal articles, research reports, and clinical reports. These literatures are invaluable knowledge source for researchers. When we perform knowledge discovery from large amount of biological data, one essential task is to recognize and identify genomic entities and discover their relations. Recently, many effective techniques have been proposed to analyze text and documents. Yet, accuracy seems to be high only when the data fits the proposed model well. We explain the motivation and issues to be solved in this section.

1.1 Automated Analysis of Biological Literature and Identification Problem

Biological literature contains many ambiguous entities including biological terms, medical terms and general terms, and so on. Genes and their transcripts often share the same name, and there are plenty of other examples of the multiplicity of meanings. The task of annotation can be regarded as identifying and classifying the terms that appear in the texts according to a pre-defined classification. However, disambiguated annotation is hard to achieve because of multiplicity of meanings and types. Generally,

E. Corchado et al. (Eds.): IDEAL 2006, LNCS 4224, pp. 688 – 696, 2006.

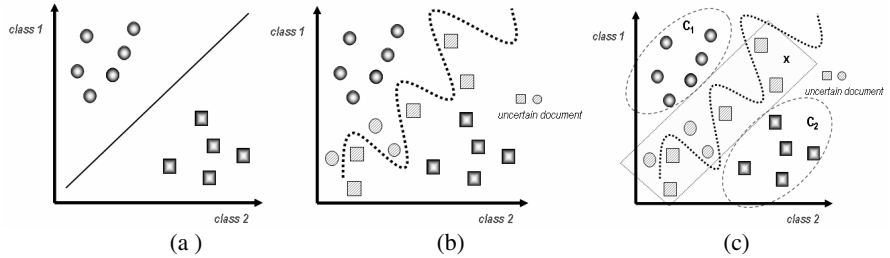
documents containing ambiguous entities are laid by decision boundaries and it is not easy for a machine to perform a reasonable classification(Fig.1). These problems reduce the accuracy of document retrieval engines and of information extraction system. Most of classifiers ignore the semantic aspects of the linguistic contents.

1.2 Classification Algorithms and Evaluation of the Performance

Automated text classification is to classify free text documents into predefined categories automatically, and whose main goal is to reduce the considerable manual process required for the task. Generally, when you evaluate the performance of automated text classification, you simply consider what kind of classifier and how many documents have been used. Traditionally, classification approaches are either statistical methods or those using NLP(Natural Language Processing) methods. Simple statistical approaches are efficient, and fast but usually lack deep understanding, and hence prone to ambiguity errors. Knowledge based NLP techniques, however, are very slow even though the quality of the result is usually better than that of statistical approaches[1,2]. Also, there are tons of classifiers based on rule base model, inductive learning model, information retrieval model, etc. Some classifiers such as Naïve Bayesian and Support Vector Machines(SVMs) is based on inductive learning based model. These classifiers have pros and cons.

1.3 Classification Problem in Complex Data

As the data size and its complexity grow fast, finding optimal line to classify is more difficult. Fig.1 shows the example of documents represented in vector. It displays the difficulty in automated classification of complex documents. A set of documents which has simple contents with lower complexity, are represented as (a). Complex documents which have multiple concepts are represented as (b). Usually, the documents located around decision boundary have multiple subjects and features. This is the area where our research is focused on.



(a) (b) (c)

Fig. 1. Finding decision rule or line for classification : A set of documents which has simple contents and lower complexity, are represented as(a). Complex documents which have multiple concepts are represented as(b). Usually, documents located around decision boundaries have multiple subjects.

In this paper, we propose a new approach based on a reinforcement training method and text and data mining combination. We have designed and implemented a

text classification system, *RTPost*, for identifying entity based on reinforcement training and post-processing method. We show that we do not need to change the classification techniques itself to improve accuracy and flexibility. This paper is organized as follows. We describe our proposed method in section 2. Section 3 presents the experimental results on the newsgroup domain. Finally, section 4 concludes the paper.

2 Method

Our goal is to maximize the classification accuracy while minimizing training costs using a refined training method and post-processing analysis. Specifically, we focus our attention to complex documents. Most of them can be misclassified, which is one of the main factors to reduce the accuracy. In this section, we present a RTPost system, which is designed in a different style from traditional methods, in the sense that it takes a fault tolerant system approach as well as a data mining strategy. We use text classification system based on text mining as a front-end system, which performs clustering and feature extraction basically. The output of the text mining, then, is fed into a data mining system, where we perform automated training using a neural net based procedure. This feedback loop can be repeated until the outcome is satisfactory to the user. In this section we describe our propose method focusing on refinement training and post-processing.

2.1 Training : Category Design and Definition

Most of the training algorithms deal with the selection problem under a fixed condition of target category. We expand the problem into designing and definition of more categories. We add a new category, X, in addition to the target category, C, to generate the initial classification results, L ,based on probabilistic scores. We define some types of class for classification purpose.

Definition 1. $\mathbf{C} = \{c_1, c_2, \ldots, c_n\}$ is a set of final target categories, where c_i and c_j are disjoint each other.($i \neq j$)

Definition 2. $\boldsymbol{SC_n} = \{c_{n1}, c_{n2}, \ldots, c_{nk}\}$ is a set of subcategories of target category c_i , where each c_{nj} are disjoint.

Definition 3. $\mathbf{X} = \{x_1, x_2, \ldots, x_{n-1}\}$ is set of intermediate categories to analyze the relevance among target classes. The data located around decision boundary belong to X. Also, unclassified documents are denoted by X, meaning special category for the documents to be assigned to target categories later.

Fig.2 shows the outline of the defined categories. Generally, the documents located along the decision boundary, lead to poor performance as they contain multiple topics and multiple features in similar frequencies. These are the typical cases which induce false positive errors and lower accuracies. We simply select and construct training samples in each class by collecting obviously positive cases. If we define a set of target categories as $C = \{c_1, c_2\}$, and number of subcategory = 2, the actual training is performed on, $T = \{c_{11}, c_{12}, x_1, x_2, c_{21}, c_{22}\}$, where x_1's are intermediate categories. The decision of the final target categories of complex documents, class x_1, and x_2, is done by the computation of distance function in the post-processing step[11].

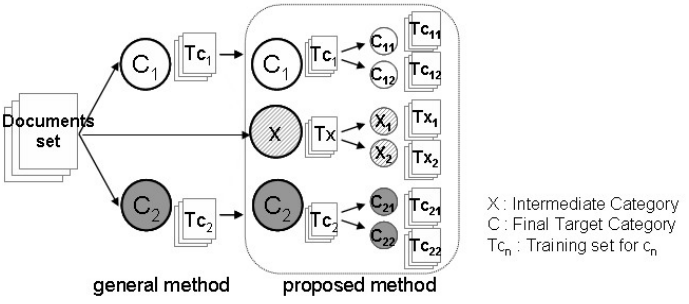


Fig. 3. Organizing method of training data : In complex documents decision boundary is not a line but a region. The data in this region is predicted as false positive. We separate the training set into target and intermediate category.

2.2 Reinforcement Post-processing Method in *RTPost* System

The main goal is to overcome these problems and limitations of traditional methods using the data mining approach. The main feature of our system is the way that we assign complex documents to the corresponding classes. We combine data mining and text mining so that they can complement each other. It is based on the structural risk minimization principle for error-bound analyses. This post–processing method consists of two stages. The front part is to assign a category to a document using the initial score calculated from the text classification result. Then, the second part is to make feedback rules to give guidelines to the previous step.

D_i \ L_i	1 (w_m = 0.02)	2 (w_m = 0.15)	3 (w_m = 0.25)	4 (w_m = 0.31)	5 (w_m = 0.35)	D_i^{size}	Step1	Step2	Assign	Actual Class
1	C_{21} .98	C_{11} .01	X_1 .01	X_2 .01	C_{12} .00	726.33	$C_{21} \rightarrow C_2$	-	C_2	C_2
2	C_{21} .39	C_{12} .20	X_2 .17	C_{11} .13	X_1 .10	31.6	X	C_2	C_2	C_2
3	X_2 .29	C_{11} .28	C_{12} .17	C_{21} .15	X_1 .01	514.42	X	C_1	C_1	C_1
4	X_1 .28	C_{21} .23	X_2 .17	C_{11} .16	C_{12} .15	287.12	X	C_2	C_2	C_2

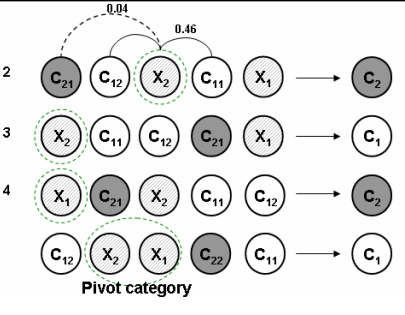


Fig. 4. Assignment examples by computation of distance between pivot category and candidate categories defined categories and experimental condition. It shows how computation is done in each candidate lists based on actual experimental data.

As a limitation of pages we simply explain about step 1 and step 2, which performs comparisons using rank scores given by the text classification result. This work is

well-presented in previous study[11]. In step 1, *min_support*, *min_value* and *diff_value* are parameters given by the user, *min_support* means the minimum support values, and *min_value* represents the minimum score to be considered the best candidate category. And *diff_value* is the difference of scores to be considered they are different.

In step 3, we make another training data for pattern analysis using the results of step 1 and step 2, which is useful in uncommon cases. Fig.3 shows how computation is done in each candidate lists based on actual experimental data. Finally, we use text mining as a preprocessing tool to generate formatted data to be used as input to the data mining system. The output of the data mining system is used as feedback data to the text mining to guide further categorization

In step 4, we analyze a whole process until classifying of document D_i is done. As input values, integrated results of previous steps are used. The goal is to minimize classification error in *RTPost* system and maintain stability in a fault tolerant manner. Fault tolerant system is designed to automatically detect faults and correct a fault effect concurrently at the cost of either performance degradation or considerable hardware or software overhead.

Table 1. Evaluation matrix for effectiveness by variance of results

Location / Input Data	Result from each step (feedback time =1)						
	$C_n^{1}.step1$	$C_n^{1}.step2$	C_n^{2}	$C_{n+1}^{1}.step1$	$C_{n+1}^{1}.step2$	C_{n+1}^{2}	
d_1	X	1	1	1	-	-	Good
d_2	X	0	1	1	-	-	Good
d_3	X	0	0	1	-	-	Poor
d_4	1	1	1	1	-	-	Fair
d_5	1	0	1	1	-	-	Fair
d_6	1	1	0	1	-	-	Poor
d_7	1	0	0	1	-	-	Poor
d_8	0	1	1	1	-	-	Good
d_9	0	0	1	1	-	-	Good
d_{10}	0	1	0	1	-	-	Poor
	✓	✓	✓	✓	-	-	

where $C_n^{e}.process$ *,*
$n = 0,1,2 \ldots$ *: feedback time,*
e : a type of input data, 1= documents 2=candidate lists of documents
process= step1, step2

In our system, the types of faults are classified to design error, parameter error and training error. We integrated results from each steps and make evaluation matrix like table 1. Table 1 is evaluation table to observe classification progress and to catch out the errors Where $C_n^{e}.process$, n refers to feedback time, and e is a type of input date; '1'=documents, '2' = candidate lists of documents, *process* refers to step1 and step2. We denote 1 when each predicted value is true, and we denote X when the document was unclassified. We can expect the location that the error is occurred as analysis of these variances in the matrix. In step 1, it is caused by parameters and category scheme, and in step2, computation of distance between pivot category and target categories is a important factor. Based on this table, we define effectiveness function

to assess how the process works well. We divide result into 3 states: *good, fair, poor* and simply make an effectiveness function, like (1).

$$E(RTPost) = \frac{1}{N}\left[\sum Good(d_i) \times benefit + \frac{1}{N}\sum Fair(d_i) - \frac{1}{N}\sum Poor(d_i) \times penalty\right] \quad (1)$$

$$benefit = log\ (n) + 1.0 \quad (2)$$

$$penalty = log\ (n) + 1.5 \quad (3)$$

If documents d_i is located around decision boundary and the result value in step1 is true, then we regard it as '*good*' case, it means *RTPost* system works very well. If d_i is not located around decision boundary and the result values in step1 and step2 are both false, then we regard it as 'poor' case, it means that there were problem in entire process. So we give penalty. Also, if d_i is not located around decision boundary and the result value in step1 is true, then we regard it as 'fair' case, it mean there is no critical problem in the process. (2) and (3) are weight values for '*good*' state and '*poor*' state. For example, the range of *E(RTPost) is* $-4.5 < E < 4$, when 1000 of test documents were used. At this time, there are above 30% of 'poor' cases without any '*good*' cases, then, *E(RTPost)* has the score below 0. If *E(RTPost)* score is lower that defined reasonable value, we need to assess that there are critical problems over the entire process.

3 Experiments

To measure the performance of our system, We experiment our system in a field where ambiguous words can cause errors in grouping and affect the result. In particular, we focused on the Rb(retinoblastoma)-related documents from the PubMed abstracts. The main difficulty of automatic classification of the documents is the ambiguity of the intended meaning of Rb, which can only be interpreted correctly when full context is considered. Possible interpretations include cancer(C), cell line(L), protein(P), gene(G), and ion(I). We perform the same experiments using Naïve Bayesian and SVM, with and without the post-processing steps, for two situations(with and without noise). We present the test conditions in Table 2 and report. Since the proposed system is developed by using a component based style using BOW toolkit[10] and C, it can be easily adapted to deal with other data or other data mining algorithms.

3.1 Classification for Disambiguation of 'RB'

Our goal is to identify the words 'Rb' or 'retinoblastoma' through the classification task. The examples of the successful tagging is as follows :

(1) P130I mediates TGF-beta-induced cell-cycle arrest inn **Rb** mutant HT-3 cells. *(gene)*

(2) The INK4alpha/ARF locus encodes p14(ARF) and p16(INK4alpha) , that function to arrest the cell cycle through the p53 and **RB** pathways, respectively. *(protein)*

(3) Many tumor types are associated with genetic changes in the **retinoblastoma** pathway, leading to hyperactivation of cyclin-dependent kinases and incorrect progression through the cell cycle. *(cancer)*

(4) The Y79 and WERI-Rb1 **retinoblastoma** cells, as well as MCF7 breast cancer epithelial cells, all of which express T-channel current and mRNA for T-channel subunits, is inhibited by pimozide and mibefradil with IC(50)= 8 and 5 microM for pimozide and mibefradil, respectively). *(cell line)*

3.2 Experimental Setting

In RB-related documents, most documents is connected with protein(P), gene(G) and cancer(C). Hence, there are a few documents connected with ion(I) and which size are very small. In this paper, we experimented with 3 classes by defined categories as shown in table 2. We equally divided each target category into two parts, and added two intermediate categories. Finally, we performed classification on the set of candidate categories, SC={P1, P2, X1, G1, G2, X2, D1, D2}. For experiments, we collected about 20,000 abstracts, and we verified our result using 200 abstracts. Especially, we put some intentional noises by adding incorrectly classified documents to target categories, which is about 10% of the total. Actually, these documents get high classification errors because these have many ambiguous features, and their contents are very intricate.

Table 2. Defined categories and Experimental Condition

Definition of category			**Number of training documents (correct + incorrect)**		
Target category (C)	Candidate category (SC)	Intermediate category(X)	Correct Documents	Incorrect documents (10%)	Total (300, 318)
Protein	P1		30	5	60(36)
	P2		30	1	
		X1	60	0	60
Gene	G1		30	3	60(36)
	G2		30	3	
		X2	60	0	60
Disease, Cancer	D1		30	6	60(36)
	D2		30	0	

We defined parameter values to assign documents in text classification, as shown in figure 3: min_support=100(bytes), min_value=0.6, diff_value=0.2. We performed analysis based on effectiveness factor, 0.5 and one-time feedback.

3.3 Experimental Result and Discussion

Table 3, 4 show the experimental results on the correct training data. According to the results, our method works very well when applied to the Naïve Bayesian or SVM classifiers. Especially, SVM and NB perform badly on the *protein* class, which is the fraction of protein-related documents that are with high complexity and multiplicity, which share multiple topics and features in the similar frequency.

Table 3. Experimental Result: Exising method and RTPost method with correct document

performance / method	Accuracy	Protein Predict Power	Gene Predict Power	Disease Predict Power	Misclassification rate
Naïve Baysian(NB)	0.69	51%	82%	74%	31%.
SVM	0.74	64%	83%	76%	29%
RTPost Algorithm(with NB)	0.89	81%	94%	92%	11%
RTPost Algorithm(with SVM)	0.91	88%	91%	94%	8%

Table 4. Experimental Result: Exising method and RTPost method with incorrect document

performance / method	Accuracy	Protein Predict Power	Gene Predict Power	Disease Predict Power	Misclassification rate
Naïve Baysian(NB)	0.45	52%	65%	17%	55%.
SVM	0.47	54%	61%	26%	64%
RTPost Algorithm(with NB)	0.85	84%	92%	75%	15%
RTPost Algorithm(with SVM)	0.87	87%	91%	81%	11%

Our system enhances both classifiers by relatively high rates. On the average, the refined classifiers are on average about 25% better the original. Especially, our method have high predict power about *gene* class consisting of '*Gene*', '*DNA*', '*mRNA*' as main features, and *cancer* class consisting of '*cancer*', '*disease*' and so on.

Table 4 shows the experimental result on the data containing incorrect training samples. According to the result, the accuracy of original method decreased 0.45 and 0.47. Generally, it is well known that Naïve Bayesian is less influenced by the training errors. However, it's predict power drops down to 17% in '*disease*' class. It clearly shows that the important features among the classes were generalized because of incorrect documents. Also, it reveals the assignment problem and the limitation of improving performance by reforming computation method based on probability models or vector models. Hence, our method significantly improved stability on training errors.

4 Conclusion

In this paper, we proposed a refinement method to enhance the performance of identifying entity using text and data mining combination. It provides a comparatively cheap alternative to the traditional statistical methods. We applied this method to analyze Rb-related documents in PubMed and got very positive results. We also have shown that our system has high accuracy and stability in actual conditions. It does not depend on some of the factors that have important influences to the classification power. Those factors include the number of training documents, selection of sample data, and the performance of classification algorithms. In the future research, we plan to simplify the effectiveness function without raising the running costs of the entire process.

References

1. Agrawal R., R. Bayardo, and R. Srikant. :Athena: Mining-based Interactive Management of Text Databases*, In Proc. of the International Conference on Extending Database Technology* (2000) 365-379
2. Koller D. and S. Tong.:Active learning for parameter estimation in Bayesian networks. *In Neural Information Processing Systems(* 2001)
3. Bing Liu, Haoran Wu and Tong Heng Phang :a Refinement Approach to Handling Model Misfit in Text Categorization, SIGKDD(2002)
4. Castillo M. D.,, J.L.Serrano:A Multistrategy Approach for Digital Text Categorization form Imbalanced Documents, SIGKDD, vol 6(2004) 70-79
5. Sheng Gao, Wen Wu, et al,:A MFoM Learning Approach to Robust Multiclass Multi-Label Text Categorization, *In Proceedings of the 21th* Intenational Conference on Machine Learning(2004)
6. Joachims T.,, :Text categorization with support vector machines: learning with many relevant features*. In Proceedings of ECML-98, 10th European Conference on Machine Learning(1998)* 137-142
7. Hasenager M.,.: Active Data Selection in Supervised and Unsupervised Learning. PhD thesis, Technische Fakultat der Universitat Bielefeld(2000)
8. Hatzivassiloglou, V., P.A. Duboue, and A.Rzhetsky. : Disambiguating Proteins, Genes and RNA in Text: a Machine Learning Approach. *Bioinformatics*(2001) Vol.17, S97-106
9. Lifeng Chen, Hongfang Liu and Caroal Friedman, : Gene Name Ambiguity of Eukaryotic Nomenclatures, Bioinformatics, Vol21, No.2, pages 248-256,. Jan 15, 2005 .
10. BOW toolkit : http://www.cs.cmu.edu/~mccallum/bow/
11. Choi, Y.J., Park, S.S. : Refinement Method of Post-processing and Training for Improvement of Automated Text Classification", *In Proc. Of the International Conference, ICCSA(2006)*