Automated Classification of PubMed Texts for Disambiguated Annotation Using Text and Data Mining

Yun Jeong Choi[1] **Seung Soo Park**[2]
[1,2]*Department of Computer Science &Engineering, Ewha Institute of Science & Technology, Ewha Womans University, Seoul,Korea*
Email : cris@ewhain.net, sspark@ewha.ac.kr

ABSTRACT:
Recently, as the size of genetic knowledge grows faster, automated analysis and systemization into high-throughput database has become hot issue. One essential task is to recognize and identify genomic entities and discover their relations. However, ambiguity of name entities is a serious problem because of their multiplicity of meanings and types. So far, many effective techniques have been proposed to analyze documents. Yet, accuracy is high when the data fits the model well. The purpose of this paper is to design and implement a document classification system for identifying entity problems using text/data mining combination, supplemented by rich data mining algorithms to enhance its performance. we propose *RTPost system* of different style from any traditional method, which takes fault tolerant system approach and data mining strategy. This feedback cycle can enhance the performance of the text mining in terms of accuracy. We experimented our system for classifying RB-related documents on PubMed abstracts to verify the feasibility.

1 INTRODUCION

Genes and their transcripts often share the same name and there are plenty of other examples of the multiplicity of meanings in bioinformatics area. The task of annotation can be explained as identifying and classifying the terms that appear in the text according to a predefined classification.

Automated text classification is to classify free text documents into predefined categories automatically and whose main goals is to reduce the considerable manual process required for the task. Traditionally, classification of texts is done either statistically or using NLP(Natural Language Processing), information retrieval and machine learning.

Simple statistical approaches are efficient and fast but usually lack deep understanding, and are hence prone to ambiguity errors[2,4,5]. Knowledge based NLP techniques, however, are very slow even though the quality of the result is usually better than that of statistical approaches[1]. Most techniques are based on some typical models but, classification accuracy will be high when the data fits the model well.

In this paper, we propose a new approach based on a reinforcement training method and text/data mining combination. We show that we do not need to change the classification techniques itself to improve accuracy and flexibility. For this purpose, we built a simple conceptual model of substances and sources to define the knowledge distance[13,14,15,16]. Based on this ontology, the names of protein, DNA, RNA, source and other molecular that appear in the abstract can be tagged accordingly. These names are considered to be relevant to the description of biological processes, and recognition of such names is crucial for understanding higher level 'event/ interaction' knowledge.

2 DESIGN AND METHOD

In this paper, we present a refinement system to improve classification performance of documents laid by decision boundary nearby. The proposed system was designed in a different style from any traditional method, which takes a fault tolerant system approach and data mining strategy. The two important parts points of the system are reinforcement training and post-processing parts. First, main point of the training method deals with the problem of defining categories to be classified before selecting training sample. Second, the post-processing method deals with the problem of assigning category, not performance of classification algorithms.

2.1 Category Design and Definition

Most training algorithms have dealt with some problems based on selection and number of training documents under a fixed condition of target category. We expand the problem into design and definition of category.

We defined some types of class for classification purpose.

- *definition 1.* $\mathbf{C} = \{c_1, c_2, .., c_n\}$ is a set of final target categories, where c_i and c_j are disjoint each other.$(i \neq j)$
- *definition 2.* $\mathbf{c_n'} = \{ c_{n1}, c_{n2}, \ldots, c_{nk} \}$ 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. The data located around decision boundary belongs to X. Also, unclassified documents are denoted by X, meaning special category for the documents in need of assignment to target category in later.
- *definition 4.* $\mathbf{L_i} = [l_{i1}, l_{i2}, \ldots, l_{im}]$ is a list of candidate categories which is list ranked by score in classification result of input document D_i.Where, l_{i1} is

the highest candidate category of input document D_i, we simply mention L_{i1}. l_{ij} is ordered pair of (c, s), where $c \in C \cup X$, s is real number between 0 and 1, given by system. A value of pruning parameter, m, must define larger then number of target category, n.

- *definition 5.* **P** is a pivot category. It denotes the highest intermediate category in L_i. If intermediate categories lies in a row, we merge them.

- *definition 6.* $\mathbf{Tr(c_i)} = \cup_k Tr(c_{ik})$ is a set of training documents for target category c_i.

- *definition 7.* $\mathbf{T} = Tr(c_i) \cup Tr(x_j)$ is a set of training documents for input documents D, which represents classification goals about D

Figure 1 shows the outline of defined categories. We added intermediate category to assign documents laid by a decision boundary. These texts generally lead to poor performance and contain multiple topics and multiple features in similar frequency; such as junk mail and various unrelated business letters. These are typical cases which induce false positive error and lower accuracy. The actual training is performed on a set of subcategories and intermediate categories, which is illustrated by figure 2.

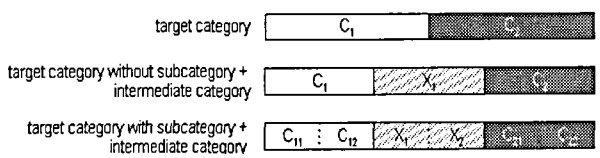

Figure 1. outline of arrangement training data in *RTPost* System

2.2 Text/Data Mining Combination based on Reinforcement Training and Post-Processing : *RTPost* System

The research on improving the text categorization performance in recent years, has focused on enhancing existing classification models and algorithms itself, however, their range has been limited by feature based statistical methodology. In this paper, we propose *RTPost* system of a different style from any traditional method, which takes a fault tolerant system approach and data mining strategy. The two important parts of *RTPost* system are reinforcement training and post-processing.

The main feature of our system is the way 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 analysis. We use text classification based on text mining method 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. The output, in turn, provides a guideline to the text mining system. 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.

2.2.1 Target Category Definition for Training

Figure 2 shows a basic idea of proposed training method. The separate line among target categories, C_1 and C_2, seem semantically certain. However, decision boundary is not line but region even though pre-labeled documents, the data in the region will be predicted as false positive. Thus we separate the training set into target and intermediate.

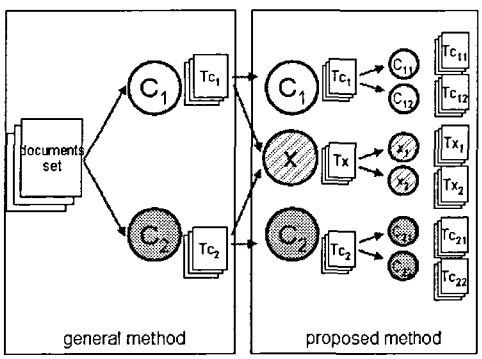

Figure 2. Organizing method of training data

2.2.2 Post-Processing

Since the statistical information derived from those existing classification methods is usually unreliable and serious mismatching between the selected candidate class by system and actual class tends to occur.

Input : *Document D_i, Candidate category list L_i normalized and resorted by descend order*

Step1 : **for** i= 0 to N(= number of input documents) {
If ($D_i^{size} \geq$ min_support)&&((L_i^1 score $\geq$ min_value)||
(L_i^1 score - L_i^2 score $\geq$ diff_value))
then assign D_i to L_i^1 **else** assign D_i to X
}

Step2 : **for** n=0 to N (= number of unassigned documents in step1) {
for n=0 to N (= number of target category) {
Calculate distance of category between P, c_{nk}

$$Dist(P, c_n) = \sum RD(P, c_{nk}) * w_m$$

} assign D_i to more closer side c_n
}

Figure 3. Step 1 and step 2 in post-processing: assignment a category to documents using initial scores

The main goal of post-processing method in *RTPost* system is to overcome the problems and limitations of traditional methods with the mining process approach which is focused on risk minimization analysis. The proposed method consists of two stages. The front part is to assign a category to documents using initial scores from text classification result. Then, the second to make feedback rules to give guideline to the previous step is following. Figure 3 is the pseudo code for step 1~2, which performs comparisons

using rank scores and total scores. We define a simple rule to assign a category using the initial scores. When the result satisfies the threshold value, it is fixed as the final result. In step 1, min_support, max_value and diff_value are parameters given by the user.

In step 2, we compare category distance between pivot category and subcategory. For the rest of the documents in previous step, we define two functions using category distance between class P, c_n. The result of step 2 is computed through (1), then, is determined as c_n which minimizes the value $Dist(P,c_n)$, like (3). Figure 4 shows computation examples between pivot category, P and candidate categories, c_n and the right side of arrow presents the final target category.

$$Dist(P, c_n) = \sum RD(P, c_{nk}) * w_m \quad (1)$$

$c_{nk} \in L_i$: list of candidate category,
m= rank order of c_{nk}

RD(x, y) means the value of the simple ordinal rank distance between x and y, if x and y are adjacent, the value is 1.

$$w_m = log\ (\sqrt{(m + \alpha)}) \quad (2)$$

weight in rank of c_{nk},
α : control parameter

$$\text{Assign} : \underset{n}{Min}\{Dist(P, c)\} \quad (3)$$

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. Figure 4 shows how computation is done in each candidate lists based on actual experimental data. As input values, rank scores and difference of category distance are used. We perform data mining analysis with these uncommon patterns, then, we gets valuable rules which is made of previous candidates pattern as condition.

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 the entire process until classification of document D_i is complete. As values are input the 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. The 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

progress	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
	√	√	√	√	-	-	

In our system, the types of faults include design errors, parameter errors and training errors. We integrated the results from each steps and made evaluation matrix such as the one seen in table 1. Table 1 is evaluation table of the classification progress, designed 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.

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 = 0.33)		D_i's c_{nk}	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

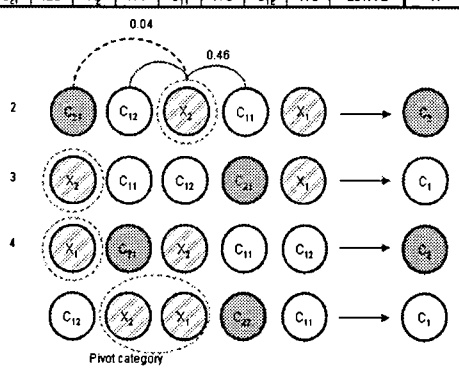

Figure 4. Assignment examples by computation of distance between pivot category and candidate categories defined categories and experimental condition in step 3

We denote 1 when each predicted value is true, and we denote X when the document was unclassified. We can expect the location of the error which occurred within the analysis of these variances in the matrix. In step 1, the errors are caused by parameters and the category scheme, and in step2, computation of distance between pivot category and target categories is an important factor.

Based on this table, we define an effectiveness function to assess how well the process performs. We classify results into 3 states: *good, fair, poor* and simply make an effectiveness function, like (4).

$$E(RTPost) = \frac{1}{N}\left[\sum Good(d_i) \times benefit + \frac{1}{N}\sum Fair(d_i) - \frac{1}{N}\sum Poor(d_i) \times penalty\right] \quad (4)$$

$$benefit = log\ (n) + 1.0 \quad (5)$$

$$penalty = log\ (n) + 1.5 \quad (6)$$

If documents d_i is located around the decision boundary and the result value in step1 is true, then we regard it as a '*good*' case, this means that the *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 a 'poor' case, this means that there were problems within the entire process. So we deduct a penalty. Also, if d_i is not located around the decision boundary and the result value in step1 is true, then we regard it as a 'fair' case, this means that there is no critical problem in the process. (6) and (7) are weight values for '*good*' states and '*poor*' states. For inctance, the range of *E(RTPost)* is $-4.5 < E < 4$, when 1000 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 the *E(RTPost)* score is lower than the defined reasonable value, we need to assess that there are critical problems underlying the entire process.

3 EXPERIMENTS

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 Rb-related 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).

3.1 Classification for Disambiguation of 'RB'

We took these above 5 categories(C, L, P, G, I) as the target Category. Our goal is to identify the words 'Rb' or 'retinoblastoma', in the Rb-related documents through the classification task. The examples of the successful tagging is as follows :

(1) P130 i 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 Setup

In RB-related documents, most documents are connected with protein(P), gene(G) and cancer(C). Hence, there are a few documents connected with ion(I) and which are very small. Therefore, we must first classify a set of categories, C={I, others}, then perform a one-against-one classification about C={P, G, D}. In this paper, we test 3 classes and define categories as seen in table 2. Each target category was equally divided into two parts and added two intermediate categories. Finally, we perform classification on the set of candidate category, S={P1, P2, X1, G1, G2, X2, D1, D2}.

For the experiments, we collected approximately 20000 abstracts from PubMed and we verified our results using a test sample of 200 abstracts. Each training set consists of 30 documents containing incorrect documents for evaluation.

We carefully selected 100 of documents to verify the system. In actual fact, these documents caused high classification errors, mainly since these have many ambiguous features and their contents are very intricate. The classification results of these documents showed good input patterns for step 2 and good training data and test data.

Table 2. Defined categories and experimental condition

Definition of category			**Number of training documents (correct + incorrect)**		
Target category (C)	Candidate category (S)	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	

Table 3. Result : existing method and *RTPost* method using training data with correct documents

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

Table 4. Result : existing method and *RTPost* method using training data containing incorrect documents

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

Incorrect documents were deliberately added to the training data for testing purposes. The aim of the experiments was to compare the stability and consistency in training sample error and a special quality of classifiers.

Two classification techniques, Naïve Bayesian and Support Vector Machines(SVM), have been tested in the proposed method as the base classifier in text mining. And we use Neural Network as pattern classifier in data mining. SVM has been proven to be a superior classifier in binary classification.

However, SVM is more sensitive to training sample distribution and does not generate substantial training error. For comparison, we use Neural Network getting low explanatory power because it is one of superior classifiers.

We defined parameter values for assigning documents in text classification, as shown by figure 3, min_support=100 (bytes), min_value=0.6, diff_value=0.2. Analysis was performed based on effectiveness factor, 0.5 and one-time feedback.

(class)	Gene	Protein	Cancer
Gene	Positive predict power		
Protein		Positive predict power	
Cancer			Positive predict power

Figure 5. 3*3 matrix to obtain positive predict power rate.

We use positive predict power and error as the evaluation measures as 3*3 matrix, as shown in figure 5, which is much simpler and a more suitable method for this experiments. Error is the ratio of the sum of the numbers of false positives and false negatives to the total number of documents.

3.3 Experimental Result and Discussion

Table 2 shows 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 NP perform badly on *protein* class, which is the fraction of protein-related documents that have high complexity and multiplicity with sharing multiple topic and features in similar frequency.

Hence, our system is relatively successful as it enhances both classifiers with overwhelming improvement. The refined classifiers are on average about 25% better the original. More importantly, our method has high predicting power about *gene* class consisting of '*Gene*', '*DNA*', '*mRNA*' as its main features, and *cancer* class consisting of '*cancer*', '*disease*' and so on.

Table 3 shows the experimental results on containment of the incorrect training samples. According to the result, the accuracy of original method decreased 0.45 and 0.47. It is well known that Naïve Bayesian is less influenced by training error, however, it's predicting power drops down to 17% in '*disease*' class. This 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 methods based on probability models or vector models.

Hence, our method significantly improved stability on training error although accuracy was likewise decreased. Our system is on average almost 100% better than the original. In addition, our technique also reduces all the error rates substantially.

In previous research, we used newsgroup data collected from Usenet articles. We performed classification on their sub categories to test classification power of our method. There are four main categories, Computing(*comp*), Recreation(*rec*), Science(*sci*) and Talk(*talk*). There are no differences among the main categories, i.e. '*comp*' and '*sci*'. But, it made some difference in sub categories and was able to perform classification with a high success rate, also a remarkable results were achieved in the lowest sub categories, e.g. *comp.sys.mac.hardward*' and '*comp.sys.ibm.hardward*'.

We have shown that our system has high accuracy and stability in actual conditions. It wholly did not depend on some variables which are important influence to classification power such as number of training documents, selection of sample data and performance of classification algorithms.

4 CONSLUSION AND FEATURE WORK

In this paper, we proposed a refinement method to handle

the problem of identifying entity using text/data mining combination and training method. It provides a comparatively cheap alternative to the traditional statistical and NLP methods.

We applied this method to analyze Rb-related documents in PubMed and got a very positive result. Since the proposed system is developed in component based style, it can be easily expanded to deal with other documents, or other mining algorithms. We plan to extend our experiments, and apply our techniques to other applications, such as identifying new patterns as well as extracting new relations from bio literatures.

Generally, in machine learning area, which data to analyze and what method is operated are most successful key. In SVM, which kernel function is applied, how variables declared and which way is used for multi-classification problem are most critical. The causes which may be able to affect *RTPost* system are organization method of training data using intermediate categories, the maximum and minimum figure of parameters used to assign categories and effectiveness function. These elements are used to guarantee and supply for previous results.

In the future, we would like 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 Proceedings of the 7th International Conference on Extending Database Technology*, pages 365-379, 2000.

[2] Yiming Yang. "An Evaluation of Statistical Approaches to Text Categorization", *Journal of Information Retrieval*, Vol.1, No.1, pages 67-88, 1999.

[3] Zijian Zheng. "Naïve Bayesian Classifier Committees". *In Proceedings of European Conference on Machine Learning*, pages 196-207, 1998.

[4] Yiming Yang and J. O. Pedersen. "A Comparative Study on Feature Selection in Text Categorization". *In Proceedings of the 14th International Conference on Machine Learning*, pages 42-420, 1997.

[5] David D. Lewis and Jason Catlett. "Heterogeneous Uncertainty Sampling for Supervised Learning". *In Proceedings of the 11th international Conference on Machine Learning*, pages 148-156, 1994.

[6] Pedro Domingos and Michael Pazzani. "Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier", *In Proceedings of the 13th International Conference on Machine Learning*, pages 105-112, 1996.

[7] Bing Liu, Haoran Wu and Tong Heng Phang," a Refinement Approach to Handling Model Misfit in Text Categorization", SIGKDD'02 pages 207-216, July, 2002.

[8] Castillo M. D.,, J.L.Serrano, "A Multistrategy Approach for Digital Text Categorization form Imbalanced Documents", SigKdd, vol 6, issue 1, pages 70-79,2004

[9] Sheng Gao, Wen Wu, Chin-Hui Lee, and Tat-Seng Chua, "A MFoM Learning Approach to Robust Multiclass Multi-Label Text Categorization", *In Proceedings of the 21th* Intenational Conference on Machine Learning (ICML'04), 2004.

[10] Joachims T.,, "Text categorization with support vector machines: learning with many relevant features". *In Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137-142, 1998

[11]Koller D. and S. Tong. "Active learning for parameter estimation in Bayesian networks". *In Neural Information Processing Systems*, 2001.

[12] Hasenager M.,. "Active Data Selection in Supervised and Unsupervised Learning". PhD thesis, Technische Fakultat der Universitat Bielefeld, 2000.

[13]Hatzivassiloglou, V., P.A. Duboue, and A.Rzhetsky. "Disambiguating Proteins, Genes and RNA in Text: a Machine Learning Approach". *Bioinformatics* Vol.17, pages S97-106, 2001.

[14] Dagan, I. And A.Itai, "Word Sense Disambiguation using a second language monolingual corpus", Computational Linguistics, 20(4), December 1994.

[15] Tateishi, Y., T. Ohta, J. Tsujii, "Building an Annotated Corpus in the Molecular-Biology Domain", *In Proceedings of COLING 2000 Workshop on Semantic Annotation and Intelligent Content*, pages 28-34, 2000.

[16] Lifeng Chen, Hongfang Liu and Caroal Friedman, "Gene Name Ambiguity of Eukaryotic Nomenclatures", Bioinformatics, Vol21, No.2, pages 248-256,. Jan 15, 2005 .