Annotating Multiple Types of Biomedical Entities: A Single Word Classification Approach

Chih Lee, Wen-Juan Hou and **Hsin-Hsi Chen**
Natural Language Processing Laboratory
Department of Computer Science and Information Engineering
National Taiwan University
1 Roosevelt Road, Section 4, Taipei, Taiwan, 106
{clee, wjhou}@nlg.csie.ntu.edu.tw, hh_chen@csie.ntu.edu.tw

Abstract

Named entity recognition is a fundamental task in biomedical data mining. Multiple-class annotation is more challenging than single-class annotation. In this paper, we took a single word classification approach to dealing with the multiple-class annotation problem using Support Vector Machines (SVMs). Word attributes, results of existing gene/protein name taggers, context, and other information are important features for classification. During training, the size of training data and the distribution of named entities are considered. The preliminary results showed that the approach might be feasible when more training data is used to alleviate the data imbalance problem.

1 Introduction

The volumn of on-line material in the biomedical field has been growing steadily for more than 20 years. Several attempts have been made to mine knowledge from biomedical documents, such as identifying gene/protein names, recognizing protein interactions, and capturing specific relations in databases. Among these, named entity recognition is a fundamental step to mine knowledge from biological articles.

Previous approaches on biological named entity extraction can be classified into two types – rule-based (Fukuda *et al.*, 1998; Olsson *et al.*, 2002; Tanabe and Wilbur, 2002) and corpus-based (Collier *et al.*, 2000; Chang *et al.*, 2004). Yapex (Olsson *et al.*, 2002) implemented some heuristic steps described by Fukuda, *et al.*, and applied filters and knowledge bases to remove false alarms. Syntactic information obtained from the parser was incorporated as well. GAPSCORE (Chang *et al.*, 2004) scored words on the basis of statistical models that quantified their appearance, morphology and context. The models includes Naive Bayes (Manning and Schutze, 1999), Maximum Entropy (Ratnaparkhi, 1998) and Support Vector Machines (Burges, 1998). GAPSCORE also used Brill's tagger (Brill, 1994) to get the POS tag to filter out some words that are clearly not gene or protein names. Efforts have been made (Hou and Chen, 2002, 2003; Tsuruoka and Tsujii, 2003) to improve the performance. The nature of classification makes it possible to integrate existing approaches by extracting good features from them. Several works employing SVM classifier have been done (Kazama *et al.*, 2002; Lee *et al.*, 2003; Takeuchi and Collier, 2003; Yamamoto *et al.*, 2003), and will be discussed further in the rest of this paper.

Collocation denotes two or more words having strong relationships (Manning and Schutze, 1999). Hou and Chen (2003) showed that protein/gene collocates are capable of assisting existing protein/gene taggers. In this paper, we addressed this task as a multi-class classification problem with SVMs and extended the idea of collocation to generate features at word and pattern level in our method. Existing protein/gene recognizers were used to perform feature extraction as well.

The rest of this paper is organized as follows. The methods used in this study are introduced in Section 2. The experimental results are shown and discussed in Section 3. Finally, Section 4 concludes the remarks and lists some future works.

2 Methods

Most of the works in the past on recognizing named entities in the biomedical domain focused on identifying a single type of entities like protein and/or gene names. It is obviously more challenging to annotate multiple types of named entities simultaneously. Intuitively, one can develop a specific recognizer for each type of named entities, run the recognizers one by one to annotate all types of named entities, and merge the results. The problem results from the boundary decision and the annotation conflicts. Instead of constructing five individual recognizers, we regarded the multiple-class annotation as a classification problem, and tried to learn a

classifier capable of identifying all the five types of named entities.

Before classification, we have to decide the unit of classification. Since it is difficult to correctly mark the boundary of a name to be identified, the simplest way is to consider an individual word as an instance and assign a type to it. After the type assignment, continuous words of the same type will be marked as a complete named entity of that type. The feature extraction process will be described in the following subsections.

2.1 Feature Extraction

The first step in classification is to extract informative and useful features to represent an instance to be classified. In our work, one word is represented by the attributes carried *per se*, the attributes contributed by two surrounding words, and other contextual information. The details are as follows.

2.1.1 Word Attributes

The word "attribute" is sometimes used interchangeably with "feature", but in this article they denote two different concepts. Features are those used to represent a classification instance, and the information enclosed in the features is not necessarily contributed by the word itself. Attributes are defined to be the information that can be derived from the word alone in this paper.

The attributes assigned to each word are whether it is part of a gene/protein name, whether it is part of a species name, whether it is part of a tissue name, whether it is a stop word, whether it is a number, whether it is punctuation, and the part of speech of this word. Instead of using a lexicon for gene/protein name annotation, we employed two gene/protein name taggers, Yapex and GAPSCORE, to do this job. As for part of speech tagging, Brill's part of speech tagger was adopted.

2.1.2 Context Information Preparation

Contextual information has been shown helpful in annotating gene/protein names, and therefore two strategies for extracting contextual information at different levels are used. One is the usual practice at a word level, and the other is at a pattern level. Since the training data released in the beginning does not define the abstract boundary, we have to assume that sentences are independent of each other, and the contextual information extraction was thus limited to be within a sentence.

For contextual information extraction at word level (Hou and Chen, 2003), collocates along with 4 statistics including frequency, the average and standard error of distance between word and entity and t-test score, were extracted. The frequency and t-test score were normalized to [0, 1]. Five lists of collocates were obtained for cell-line, cell-type, DNA, RNA, and protein, respectively.

As for contextual information extraction at pattern level, we first gathered a list of words constituting a specific type of named entities. Then a hierarchical clustering with cutoff threshold was performed on the words. Edit distance was adopted as the measure of dissimilarity (see Figure 1). Afterwards, common substrings were obtained to form the list of patterns. With a list of patterns at hand, we estimated the pattern distribution, the occurrence frequencies at and around the current position, given the type of word at the current position. Figure 2 showed an example of the estimated distribution. The average KL-Divergence between any two distributions was computed to discriminate the power of each pattern. The formula is as follows:

$$\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} D(p_i \| p_j)\,,$$ where p_i and p_j are the distributions of a pattern given the word at position 0 being *type i* and *j*, respectively.

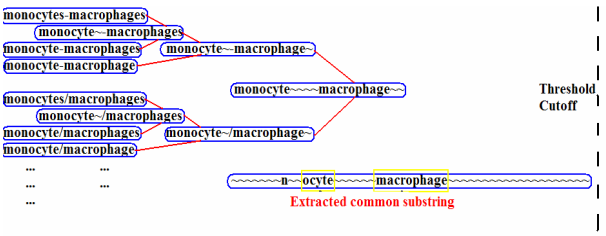

Figure 1: Example of common substring extraction

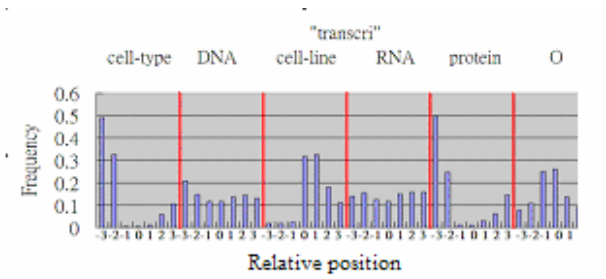

Figure 2: Pattern distributions given the type of word at position 0

2.2 Constructing Training Data

For each word in a sentence, the attributes of the word and the two adjacent words are put into the feature vector. Then, the left five and the right five words are searched for previously extracted collocates. The 15 variables thus added are shown below.

$$\sum_{i=-5,i\neq 0}^{5} Freq(w_i \mid type)$$

$$\sum_{i=-5,i\neq 0}^{5} t-test\_score(w_i \mid type)$$

$\sum_{i=-5, i\neq 0}^{5} f(i \mid \hat{m}_{w_i,type}, \hat{s}_{w_i,type})$, where f is the pdf of normal distribution, *type* is one of the five types, w_i denotes the surrounding words, $\hat{m}_{w_i type}$ and $\hat{s}_{w_i type}$ are the maximum likelihood estimates of mean and standard deviation for w_i given the type. Next, the left three and right three words along with the current word are searched for patterns, adding 6 variables to the feature vector.

$\sum_{i=-3}^{3} \sum_{p \in P_{w_i}} \text{Prob}_p(i \mid type)$, where *type* is one of the six types including 'O', P_{w_i} is the set of patterns matching w_i, Prob_p denotes the pmf for pattern *p*. Finally, the type of the previous word is added to the feature vector, mimicking the concept of a stochastic model.

2.3 Classification

Support Vector Machines classification with radial basis kernel was adopted in this task, and the package LIBSVM – A Library for Support Vector Machines (Hsu *et al.*, 2003) was used for training and prediction. The penalty coefficient *C* in optimization and *gamma* in kernel function were tuned using a script provided in this package.

The constructed training data contains 492,551 instances, which is too large for training. Also, the training data is extremely unbalanced (see Table 1) and this is a known problem in SVMs classification. Therefore, we performed stratified sampling to form a smaller and balanced data set for training.

Type	# of instances (words)
cell-type	15,466
DNA	25,307
cell-line	11,217
RNA	2,481
protein	55,117
O	382,963

Table 1: Number of instances for each type

3 Results and Discussion

Since there is a huge amount of training instances and we do not have enough time to tune the parameters and train a model with all the training instances available, we first randomly selected one tenth and one fourth of the complete training data. The results, as we expected, showed that model trained with more instances performed better (see Table 2). However, we noticed that the performances vary among the 6 types and one of the possible causes is the imbalance of training data among classes (see Table 1). Therefore we decided to balance the training data.

First, the training data was constructed to comprise equal number of instances from each class. However, it didn't perform well and lots of type 'O' words were misclassified, indicating that using only less than 1% of type 'O' training instances is not sufficient to train a good model. Thus two more models were trained to see if the performance can be enhanced. One model has slightly more type 'O' instances than the equally balanced one, and the other model has the ratio among classes being 4:8:4:1:8:16. The results showed increase in recall but drop in precision.

Kazama *et al.* (2002) addressed the data imbalance problem and sped up the training process by splitting the type 'O' instances into sub-classes using part-of-speech information. However, we missed their work while we were doing this task, and hence didn't have the chance to use and extend this idea.

After carefully examining the classification results, we found that many of the 'DNA' instances were classified as 'protein' and many of the 'protein' instances were classified as 'DNA'. For example, 904 out of 2,845 'DNA' instances were categorized as 'protein' under 'model 1/4'. The reason may be that Yapex and GAPSCORE do not distinguish gene name from protein names. Even humans don't do very well at this (Krauthammer *et al.*, 2002).

We originally planned to verify the contribution of each type of features. For example, how much noise was introduced by using existing taggers instead of lexicons. This would have helped gain more insights into the proposed features.

4 Conclusion and Future work

This paper presented the preliminary results of our study. We introduced the use of existing taggers and presented a way to collect common substrings shared by entities. Due to lack of time, the models were not well tuned against the two parameters – C and gamma, influencing the capabilities of the models. Further, not all of the training instances provided were used to train the model, and it will be interesting and worthwhile to investigate. How to deal with data imbalance is another important issue. By solving this problem, further evaluation of feature effectiveness would be facilitated. We believe there is much left for our approach to improve and it may perform better if more time is given.

	Model 1/10			Model 1/4					
	Recall	Prec.	F-score	Recall	Prec.	F-score	Recall	Prec.	F-score
Full (Object)	0.4756	0.4399	0.4571	0.5080	0.4759	0.4914			
Full (protein)	0.5846	0.4392	0.5016	0.6213	0.4614	0.5296			
Full (cell-line)	0.2420	0.2909	0.2642	0.2820	0.3341	0.3059			
Full (DNA)	0.2784	0.3249	0.2998	0.2888	0.4479	0.3512			
Full (cell-type)	0.3863	0.5752	0.4622	0.4196	0.6115	0.4977			
Full (RNA)	0.0085	0.1000	0.0156	0.0000	0.0000	0.0000			
	Model balanced equally			**Model slightly more 'O'**			**Model 4:8:4:1:8:16**		
Full (Object)	0.1480	0.0990	0.1186	0.1512	0.1002	0.1206	0.5036	0.3936	0.4419
Full (protein)	0.1451	0.1533	0.1491	0.1458	0.1527	0.1492	0.5629	0.4280	0.4863
Full (cell-line)	0.1580	0.0651	0.0922	0.2280	0.0319	0.0560	0.4060	0.2261	0.2904
Full (DNA)	0.1326	0.0466	0.0690	0.1591	0.0582	0.0852	0.3759	0.2457	0.2972
Full (cell-type)	0.1650	0.1375	0.1500	0.1494	0.1908	0.1676	0.4701	0.4900	0.4798
Full (RNA)	0.0932	0.0067	0.0126	0.0169	0.0075	0.0104	0.0593	0.1148	0.0782

Table 2: Performance of each model (only FULL is shown)

References

E. Brill. 1994. Some Advances in Transformation-Based Part of Speech Tagging. *Proceedings of the National Conference on Artificial Intelligence*. AAAI Press; 722-727.

C. Burges. 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2: 121-167.

J.T. Chang, H. Schutze and R.B. Altman. 2004. GAPSCORE: Finding Gene and Protein Names One Word at a Time. *Bioinformatics*, 20(2): 216-225.

N. Collier, C. Nobata and J.I. Tsujii. 2000. Extracting the Names of Genes and Gene Products with a Hidden Markov Model. *Proceedings of 18th International Conference on Computational Linguistics*, 201-207.

K. Fukuda, T. Tsunoda, A. Tamura and T. Takagi. 1998. Toward Information Extraction: Identifying Protein Names from Biological Papers. *Proceedings of Pacific Symposium on Biocomputing*, 707-718.

W.J. Hou and H.H. Chen 2002. Extracting Biological Keywords from Scientific Text. *Proceedings of 13th International Conference on Genome Informatics*; 571-573.

W.J. Hou and H.H. Chen. 2003. Enhancing Performance of Protein Name Recognizers Using Collocation. *Proceedings of the ACL 2003 Workshop on NLP in Biomedicine*, 25-32.

C.W. Hsu, C.C Chang and C.J. Lin. 2003. A Practical Guide to Support Vector Classification. http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html.

J. Kazama, T. Makino, Y. Ohta and J. Tsujii. 2002. Tuning Support Vector Machines for Biomedical Named Entity Recognition. *Proceedings of the ACL 2002 workshop on NLP in the Biomedical Domain*, 1-8.

M. Krauthammer, P. Kra, I. Iossifov, S.M. Gomez, G. Hripcsak, V. Hatzivassiloglou, C. Friedman and A. Rzhetsky. 2002. Of truth and pathways: chasing bits of information through myriads of articles. Bioinformatics, 18(sup.1):S249-S257.

K.J. Lee, Y.S. Hwang and H.C. Rim. 2003. Two-Phase Biomedical NE Recognition based on SVMs. *Proceedings of the ACL 2003 Workshop on NLP in Biomedicine*, 33-40.

C.D. Manning and H. Schutze. 1999. Foundations of Statistical Natural Language Processing. MIT Press.

F. Olsson, G. Eriksson, K. Franzen, L. Asker and P. Liden. 2002. Notions of Correctness when Evaluating Protein Name Taggers. *Proceedings of the 19th International Conference on Computational Linguistics*, 765-771.

A. Ratnaparkrhi. 1998. Maximum Entropy Models for Natural Language Ambiguity Resolution. PhD Thesis, University of Pennsylvania.

K. Takeuchi and N. Collier. 2003. Bio-Medical Entity Extraction using Support Vector Machines. *Proceedings of the ACL 2003 workshop on NLP in Biomedicine*, 57-64.

L. Tanabe and W.J. Wilbur. 2002. Tagging Gene and Protein Names in Biomedical Text. *Bioimformatics*, 18(8): 1124-1132.

Y. Tsuruoka and J. Tsujii. 2003. Boosting Precision and Recall of Dictionary-based Protein Name Recognition. *Proceedings of the ACL 2003 Workshop on NLP in Biomedicine*, 41-48.

K. Yamamoto, T. Kudo, A. Konagaya and Y. Matsumoto. 2003. Protein Name Tagging for Biomedical Annotation in Text. *Proceedings of the ACL 2003 workshop on NLP in Biomedicine*, 65-72.