SVM-based Protein-Protein Interaction Extraction from Medline Abstracts

Baojin Cui [#1], Hongfei Lin [#2], Zhihao Yang [#3]

[#] *Department of Computer Science and Engineering, Dalian University of Technology, Dalian, 116024, China*

[1] cuibaojin@gmail.com

[2] hflin@dlut.edu.cn

[3] yangzh@dlut.edu.cn

***Abstract*—Nowadays, protein-protein interaction (PPI) extraction has become a research focus. Many methods have been applied to this domain, such as supervised learning approaches. This paper applied Support Vector Machine (SVM) to extract PPI, which bases on several lexical features and one syntactic feature achieved through link grammar parser. Due to syntax's complexity different sentence structure can not have the same parse tree, which leads to the data sparseness problem here. In order to solve the sparseness problem, syntactic feature is used in a simple way. This special syntactic feature helps to improve F-score nearly five percentage points in supervised learning approach.**

I. INTRODUCTION

Protein-protein interaction has become a research focus in the field of molecular biology because people intensively demands for automatic discovery of interactions in the literature. The goal of PPI extraction is to recognize various interactions between proteins, or other molecules from the biomedical literature. Characterizing protein interaction partners is crucial to understanding not only the functional role of individual proteins but also the organization of entire biological processes. Owing to the availability of the large Medline abstract collection publicly available, most of the current work has been done on Medline abstracts.

The protein-protein interaction extraction belongs to relation extraction task. Some work has been reported. Zelenko et al. [12] utilize a kernel-based classification approach to extract relations by computing kernel functions between parse trees. Culotta and Sorensen [1] use a similar approach as Zelenko's method and further extend it to estimate kernel functions between augmented dependency trees. Due to the computation complexity, speed is still a serious problem for kernel approaches to be used in practical applications. Nanda [7] has proposed using Maximum Entropy Model to integrate lexical, syntactic and semantic features for relation detection and characterization (RDC) task containing 24 relation types on news articles with Automatic Content Extraction, an evaluation conducted by NIST to measure information extraction technologies. It shows a better performance than Culotta and Sorensen on ACE corpus.

PPI extraction aims to recognize various interactions between proteins, or other molecules from the biomedical literature. Inspired by Nanda's work, Juan Xiao et al [4] have also proposed using Maximum Entropy models to combine diverse lexical, syntactic and semantic features for PPI extraction. It shows that the use of shallow lexical features contributes a large portion of performance improvements in contrast to the use of parsing or partial parsing information. Yet such lexical features have never been used before in other PPI extraction systems. As a result, such a new approach achieves a very encouraging result on the Interaction Extraction Performance Assessment (IEPA) corpus provided. IEPA corpus [3] is mined from Medline. Ding, J et al. [4] extracts biochemical interactions using a Link Grammar Parser in IEPA corpus. The experiment indicates that although the parser was originally developed for conversational English and made many mistakes in parsing sentences from the biochemical domain, it nevertheless achieved better overall performance than a co-occurrence-only method.

Supervised learning had been reported by Huang Minlie, et al for PPI extraction, but only preliminary templates induction has been implemented. Craven,M. and Kumlien,J. [2] used sentence classification approach for subcellular-location relations. It's not suitable for PPI extraction, since there may be more than one PPI and judgments needed when there're more than two proteins existing in a sentence. On the other hand, Marcotte EM, et al's [6] supervised learning text classification can only decide PPI information which is only mentioned in the text without the extraction function.

Inspired by the work of Juan Xiao and Ding, J, we aim to test lexical features' contribution to the overall performance, and how much improvement syntactic feature will work for PPI extraction. Our experiment shows that simple lexical features play a great role in PPI on the IEPA corpus and the link path feature helps to improve F-score nearly five percentage points.

The remaining part of this paper is organized as follows: Section 2 describes our methods. Section 3 presents the experiment and discussion. Section 4 makes conclusion. Finally, Section 5 lists some relative reference literatures.

II. METHODS

A. Link Grammar Parse

Link grammar was first introduced by Sleator and Temperley to simplify English grammar with a context free grammar [8]. The basic idea of link grammar is to connect pairs of words in a sentence with various links. Each word is

viewed as a block with connectors coming out. There are various types of connectors, and connectors may point to the right or to the left. A link consists of a left-pointing connector connected with a right-pointing connector of the same type on another word. A valid sentence is one in which all the words are connected in some way. Based on link grammar a parser is developed. The parsers' dictionary can also be easily enhanced to produce better parses for biomedical text [9]. Owing to the dictionary, parser can recognize most words in biomedical domain. By the way, link grammar parser can recognize most words' parts of speech.

B. Support Vector Machine

The support vector machine (SVM), introduced by Vapnik [11], is a learning algorithm for solving two-class pattern recognition problems and is known for its good performance. SVM is used for many types of natural language processing.

Each training data sample is labeled either a positive or negative example.

$$(x_1, y_1),...,(x_k, y_k) \quad x_i \in R^n, y_i \in \{+1,-1\}.$$

$\boldsymbol{x_i}$ is the feature vector of the i-th sample and has a dimension of $\boldsymbol{n}$. $\boldsymbol{k}$ is the total number of vectors. $\boldsymbol{y_i}$ is the class of the vector; it is either a positive example (+1) or negative example (-1). SVM separates the two types of examples with a hyper plane. The hyper plane is given by

$$(w \cdot x) + b = 0 \qquad w \in R^n, b \in R.$$

The two dashed lines are the boundaries between the positive and negative examples. They are parallel to the hyper plane, and the distance between them is called the margin. The two dashed lines and margin d are given by

$$(w \cdot x) + b = \pm 1 \quad d = 2/\|w\|.$$

An SVM finds the values for the variables $\boldsymbol{w}$ and $\boldsymbol{b}$ which maximizes the margin for the training data. These variable values minimize $\|\boldsymbol{w}\|$ under the constraint

$$y_i[(w \cdot x) + b] \geq 1.$$

In general, while the training data cannot be linearly separated, the non-linear boundary can be made linear using a kernel function $\boldsymbol{K}$ ($\boldsymbol{x_i}$, $\boldsymbol{x_j}$), which moves the training data to a high-dimensional space. Of the various types of kernels available, a d-th polynomial kernel is used:

$$K(x_i, y_j) = (x_i \cdot y_j + 1)^d.$$

SVM^{light} [10] developed by Joachims, T. is used as our classifier. SVM^{light} is an implementation of Vapnik's Support Vector Machine for the problem of pattern recognition, for the problem of regression, and for the problem of learning a ranking function.

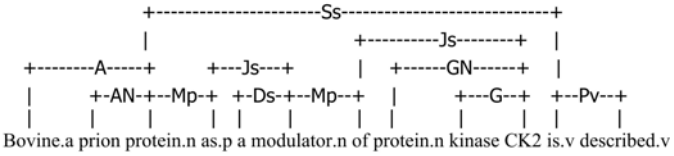
With the help of link grammar parser which includes a 60000 ordinary dictionary [9] and a 200000 biomedical dictionary [10], complex syntactic information of sentences is extracted. Then after analyzing all kinds of information including syntactic and lexical information, SVM^{light} is used to extract the PPI relationship.

C. Features

In order to see the performance clearly, only four basic features and one link path feature are chosen:

- Words in two protein names. These features include all words that appear in two protein names.
- Words between two protein names. These features include all words that are located between two protein names. If no word appears between two protein names, "NULL" is the value to be set for this feature.
- Words surrounding two protein names. These features include left n words of the first protein name and right n words of the second protein name. n is the number of surrounding words considered which is set to be seven in our experiment. Similar to words between two proteins, if there is no word surrounding two protein names; "NULL" will be used instead.
- Pair of anchors of two protein names. The anchor of each protein is extracted first. Then two anchor words are combined to form a single word. Features in feature-based classification methods are treated as independent of each other; therefore, two protein names are combined to evaluate them together.
- Link path. Using the Link Grammar parser, every sentence is parsed to find whether there exists a link path from the first protein to the second protein. The path consists of words passed and corresponding link types, but itself is not considered as the feature. If there does exist a path, the value of the feature is set to be "LinkIsTrue", and if not, the value is "LinkIsFalse". Parsing sentence "Bovine prion protein as a modulator of protein kinase CK2" using link grammar parser, we search for whether there exists a link path from bovine prion protein to kinase CK2. In this sentence, there does exist a link path like "bovine prion protein >Mp >as.p >Jp >modulator >Mp >of >Js >kinase CK2". So the value of the feature is set to be "LinkIsTrue". An example is shown in Figure 1.

(a)



(b)

Bovine.a prion protein.n >Mp>as.p>Jp>modulator.n>Mp>of>Js> kinase CK2

Fig. 1. The results after parsing

III. Experiment and Discussion

A. Process Flow

As shown in Figure 2, the experiment is composed of five steps. Name Entity Recognition spots every protein name in every sentence including the same entity in different position. Simple preprocessing on one hand erases unnecessary characters outside protein names, such as '(' and ')', on the other hand append "is described" for a fragmental title [4].

Then a link grammar parser is used to find the link path from one protein to another protein. According to two proteins' location, Sentence Analysis divides sentence into three parts: left from first protein, right from second protein and words between two protein names. At last with this information, SVM^{light} is used as our classifier to judge whether the protein pair has interaction relationship. Actually the extraction is modeled as a binary classification problem.

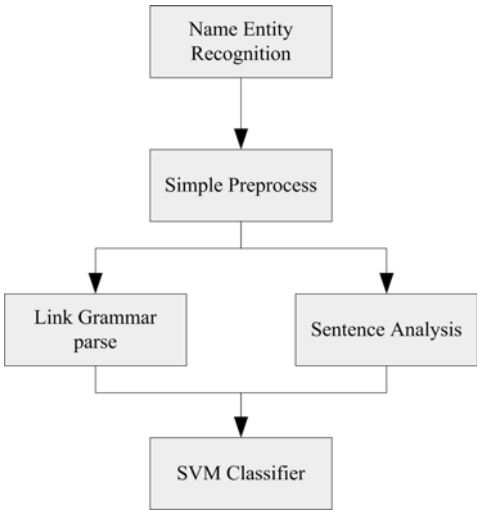


Fig. 2. Process flow

B. Results

The corpus used is the Interaction Extraction Performance Assessment (IEPA) corpus which is provided by Iowa State University [3]. It consists of 303 abstracts including 487 sentences retrieved from Medline. All protein names are tagged correctly, so that our approach can focus on the interaction extraction. Each sentence contains at least one pair of proteins. There are 644 co-occurrences in 487 sentences. Among these co-occurrences there are 336 positive instances and 308 negative instances.

TABLE I
THE PERFORMANCE OF DIFFERENT FEATURES AND THEIR COMBINATIONS

Words in two names	*	*	*	*	*
Words between two names		*	*	*	*
Words surrounding two names			*	*	*
Pair of proteins				*	
Link path					*
Precision (%)	59.4	67.5	69.2	69.1	72.8
Recall (%)	61.9	79.2	81.5	77.4	87.5
F-score (%)	60.6	72.9	74.9	73.0	79.5

The positive and the negative are divided into two equal parts respectively as train set and test set, and measure the performance using precision/recall/F-score. The performance of different features and their combinations are shown in Table 1.

C. Discussion

The corpus used in our experiment is IEPA, the same as Juan Xiao's., but the total of positive instances and negative instances we find is not consistent with his. According to Ding J. [4], there are 644 PPI pairs in all, including 336 positive pairs. But Juan Xiao's work [5] finds 633 positive instances and 1080 negative instances. The number of PPI pairs we used in the experiment is in accordance with Ding J's.

As basic lexical features, words in two protein names and words between two names improve the result significantly as shown in Table 1. Especially the second feature contributes more than 12 percentage points to F-score. The two simple features play a great role.

The F-score increases from 72.9 to 74.9, two percentage points, after surrounding words features are added into feature set. The number of surrounding words in the experiment is set to be seven when the best result is achieved.

The introduction of feature of pair of protein names is not as useful as we expect. They do not improve the result, but lower the recall and consequently lead to a little reduction of F-score. But in Juan Xiao's work, the feature contributes much. Due to the different positive and negative sum, or different method, same feature probably plays different roles.

The last feature, link path, helps to improve F-score nearly five percentage points. In Juan Xiao's work, out of expectation, the use of parse tree features and dependent tree features deteriorate the performance because of their various values. But in our experiment the component of link path itself, consisted of link type and link word, is not set to be the feature value because syntax is so complicated in language that different sentence structure can not have the same parse tree. So in order to solve the sparseness problem syntactic feature is used in a simple way to solve the sparseness problem. The link path's existence is considered to be the feature value. It shows that Link path's existence is an effective way to solve the puzzle in supervised learning approach.

IV. CONCLUSION

This paper applies a supervised learning approach for protein-protein interaction extraction using Support Vector Machine model. Several lexical features and one syntactic feature are incorporated. It shows that these shallow lexical features such as words in two protein names, words between two protein names do contribute a large portion of performance and syntactic feature in link path's existence description also contributes significantly. For further study, we will introduce more features to improve the performance of PPI extraction and try the experiment on other corpora to verify the link path's existence feature's effect furthermore.

ACKNOWLEDGMENT

This work is supported by grant from the Natural Science Foundation of China (Nos.60373095 and 60673039) and the National High Tech Research and Development Plan of China (2006AA01Z151). We are grateful to Professor Berleant from Iowa State University of Science and Technology for providing the IEPA corpus.

REFERENCES

[1] A. Culotta, and J. Sorensen, "Dependency tree kernels for relation extraction", *in Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, pp. 423–429, Jul. 2004.

[2] M. Craven, and J. Kumlien, "Constructing biological knowledge bases by extracting information from text sources", in *Proc. of the 7th International Conference on the Intelligent System for molecular Biology, Heidelburg*, Germany, pp. 77–86, Aug. 1999.

[3] J. Ding, D. Berleant, D. Nettleton, and E. Wurtele, "Mining MEDLINE: abstracts, sentences, or phrases", in *Proc. Pacific Symposium on Biocomputing, Hawaii*, USA, pp. 326–337, Jan. 2002.

[4] J. Ding, D. Berleant, J. Xu, and A. W. Fulmer, "Extracting biochemical interactions from MEDLINE using a link grammar parser", in *Proc. the 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento*, CA, pp. 467, Nov. 2003.

[5] J. Xiao, J. Su, G. D. Zhou and C. L. Tan, "Protein-protein interaction extraction: a supervised learning approach" , *in Proc. of First International Symposium on Semantic Mining in Biomedicine,* Cambridgeshire, UK, Apr. 2005.

[6] E. M. Marcotte, and I. Xenarios, "Mining literature for protein-protein interactions", *Journal of Machine Learning Research*, vol.17, pp. 359–363, 2001.

[7] K. Nanda, "Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations", in *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, pp. 178–181, Jul. 2004.

[8] D. Sleator and D. Temperley, "Parsing English with a link grammar", in *Third International Workshop on Parsing Technologies*, Oct. 1991

[9] P. Szolovits, "Adding a medical lexicon to an English parser", in *Proc. AMIA 2003 Annual Symposium, Bethesda*, MD, pp. 639–643.

[10] T. Joachims, *Making large-Scale SVM Learning, Practical Advances in Kernel Methods - Support Vector Learning*, Cambridge: MIT-Press, 1999.

[11] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[12] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction", *Journal of Machine Learning Research*, vol.3, pp.1083–1106, 2003.