Protein-Protein Interaction Extraction: A Supervised Learning Approach

Juan Xiao[1,2], Jian Su[1], GuoDong Zhou[1] and ChewLim Tan[2]

[1]Institute for Infocomm Research, Singapore [2]School of Computing, National University of Singapore; {stuxj, sujian, zhougd}@i2r.a-star.edu.sg, tancl@comp.nus.edu.sg

Abstract

In this paper, we propose using Maximum Entropy to extract protein-protein interaction information from the literature, which overcomes the limitation of the state of art co-occurrence based and rule-based approaches. It incorporates corpus statistics of various lexical, syntactic and semantic features. We find 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 of 93.9% recall and 88.0% precision on IEPA corpus provided.

To the best of our knowledge, not only is this the first systematic study of supervised learning and the first attempt of feature-based supervised learning for PPI extraction, but it also provides useful features, such as surrounding words, key words and abbreviations, to extend the supervised learning capability for relation extraction to other domains such as news.

1. Introduction

Protein-protein interaction is becoming critical in the field of molecular biology due to demands for automatic discovery of molecular pathways and interactions in the literature. The goal of PPI extraction is to recognize various interactions, such as transcription, translation, post translational modification, complexing and dissociation between proteins, drugs, or other molecules from the biomedical literature. Due to the availability of the large MedLine abstract collection publicly available, most of the current work has been done on MedLine abstracts.

Existing PPI works can be roughly divided into two categories: co-occurrence based approaches (Stapley and Benoit, 2000 and Shatkay et al., 2000) and rule-based approaches (Ono et al., 2001; Koike et al., 2003; Thomas et al., 2000; Friedman et al., 2001; Daraselia et al.). Co-occurrence based approaches simply use co-occurrence statistics of two proteins to predict their relation. In this way, they can only extract well-known PPIs but may not be able to find new emerging PPIs. On the other hand, rule-based approaches utilize pre-defined phrase pattern rules. As a result, they are unable to discover new phrase patterns without the known keywords. Once the rule set reaches a certain size, it is very difficult to insert additional rules for further performance improvement. Moreover, rule-based approaches may require redefining of the whole pattern rules when they are applied to a new domain.

The protein-protein interaction extraction is a relation extraction task. In the relation extraction with news domain, some work has also been reported. Zelenko et al., 2003 utilize a kernel-based classification approach to extract relations by computing kernel functions between parse trees. Culotta and Sorensen, 2004 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, 2004 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 (ACE[1], 2004), an evaluation conducted by NIST to measure information extraction technologies. It shows a better performance than Culotta and Sorensen, 2004 on ACE corpus. Inspired by Nanda's work, we propose in this paper to use Maximum Entropy models to combine diverse lexical, syntactic and semantic features for PPI extraction. We would like to see how good it will be for PPI extraction, and what kinds of features are needed here and the corresponding contribution to the overall performance. On implementation, our system has shown very encouraging performance with 93.9% recall and 88.0% precision on the IEPA corpus. We also find that some features, including surrounding words feature, keyword feature and mention pairs which were not used in Nanda's work, are very useful for PPI extraction.

[1] http://www.nist.gov/speech/tests/ace

Comparing with co-occurrence based approaches, our approach has the ability to extract newly discovered protein-protein interactions. In contrast to rule-based approaches, our approach can discover new phrase patterns which are not captured in the known trigger word list. It is also able to incorporate the corpus statistics of various features to achieve good performance. Furthermore it can be easily adapted to extract other relations among biomedical entities given in the training corpus instead of re-writing phrase pattern rules.

Another issue worth noticing is that different systems define different scopes on PPI information. For example, consider the sentence "We studied the interaction of protein A and protein B". The "Protein A–Protein B" interaction in this sentence is not considered in some systems, because this sentence does not indicate any experimental result. On the other hand, some systems will consider any mention of protein–protein interaction. Here, we adopt a two step approach. The first step is to extract any mention of protein-protein interaction. The second step is to classify the mentions, whether they belong to potential interaction as in the above example, or negative interaction, or positive interaction. In the second step, some attributes of interaction, such as positive, negative, direction and etc, may be extracted also. In this paper we focus on the first step, which is to extract any explicit mentions of protein-protein interaction.

Further more, although supervised learning had been reported by Huang Minlie, et al 2004 for PPI extraction, only preliminary pattern induction has been implemented, which is basically corpus statistics on POS patterns without any pattern generation to cover new similar patterns which is not available in corpus. Craven, M., and Kumlien, J., 1999 used sentence classification approach for subcellular-location relations. It's not suitable for PPI extraction, since there may be more than one PPI and judgement needed when there're more than two proteins existing in a sentence. On the other hand, Marcotte EM, et al 2001's supervised learning text classification can only decide PPI information which is only mentioned in the text without the extraction function. Palakal M, et al, 2002 only use HMM to decide the direction of PPI provided, which is much simpler task than PPI extraction itself. In summary, our approach is the first systematic study of supervised learning and the first attempt of feature-based supervised learning for PPI extraction.

The rest of this paper is organized as follows. We introduce our system work flow in section 2, and a Maximum Entropy Classifier in section 3. We further report our experiment in section 4, followed by a discussion in section 5 and a conclusion in section 6.

2. Our System Work Flow

Our system consists of 6 steps to extract interaction information from the input sentences. The system work flow is shown in Figure 1. After the Tokenization and Morphological Analysis step, all the word root forms have been derived. Following POS Tagging, Named Entity Recognition spots all the protein names in the input sentences. The Sentence Analysis step further comes up with base text chunks, such as base NP, base VP, and a parse tree with the input sentence shown in figure 2. The Co-reference Resolution is to link different mentions of same proteins together. Different mentions include protein names, their abbreviations, synonyms, and other nominal mentions, such as "this protein", "it", "they", etc. With all these information of an input sentence, a Maximum Entropy Classifier is further trained on the training corpus to make a judgment as to whether the current protein pair has interaction relationship. In other words, we model the extraction as a binary classification problem. We will introduce the maximum entropy model and features used in the classifier in section 4.

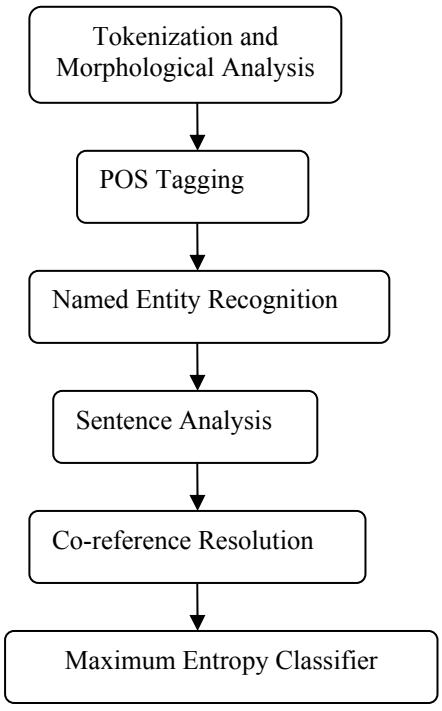

Figure. 1. Flow chart of the main steps of our system.

3. Maximum Entropy Classifier

Maximum entropy model is a probability distribution estimation technique widely used in recent years for natural language processing tasks, such as part-of-speech tagging (Ratnaparkhi et al., 1996), text classification (Nigam et al., 1999) and named entity recognition (Chieu and Ng., 2002). Nanda, 2004 first introduced Maximum Entropy Model for relation extraction on ACE corpus. Inspired by Nanda's work, we propose in this paper to use Maximum Entropy models to combine diverse lexical, syntactic and semantic features for PPI extraction.

3.1 Maximum Entropy Model

The principle of the maximum entropy model in estimating probabilities is to include as much information as is known from the data while making no additional assumptions. The probability distribution that satisfies the above property is the one with the highest entropy. The maximum entropy model is defined as:

$$P(o,h) = \frac{1}{Z(h)} \prod_{j=1}^{k} \alpha_j^{f_j(h,o)}$$

where o is the outcome, h is the history or context (feature vector in our task). $Z(h)$ is a normalization function. $\{f_1, f_2, ..., f_k\}$ are feature functions and $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ are the model parameters. Each parameter corresponds to exactly one feature and can be viewed as a "weight" for that feature. All features used in the maximum entropy model are binary, which is defined as:

$$f_j(h,o) = \begin{cases} 1, & \text{if } o = true, keyword = inhibit; \\ 0, & otherwise. \end{cases}$$

In our PPI task, o is either true or false on whether the current protein pair has interaction relationship, h is the feature vector, f_i is a feature function. We have used the open NLP maximum entropy package[2] in our system.

3. 2 Features

To achieve high performance, we explore various features to capture lexical, syntactic and semantic information and examine the effect of utilizing these features. The set of features evaluated in our system are listed as follows:

- **Words**

There are three sets of word features used in our system. We use a different feature label for each set of word features.

1. Words from two protein names
   These features include all words that appear in two protein names.
2. 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.
3. 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 three in our experiment. Similar to words between two proteins, if there is no word surrounding two protein names; "NULL" will be used instead.
   All words are treated as bag-of-word. That is, the order of these words is not considered.

- **Overlap**

Number of the other protein names that appear between two protein names.

- **Keyword**

If there is a keyword existing between two protein names or among the surrounding words of two protein names, the keyword and its position are added into the keyword feature. There are three kinds of positions: (1) between two protein names; (2) within n words left of the first protein name; (3) within n words right of the second protein name. n is set to be three in our experiment. The keyword list with Joshua M. Temkin and Mark R. Gilder, 2003[3] is used for this feature.

- **Chunks**

Each sentence is parsed by a partial parser to capture phrase level information of training examples. Chunk features used in our system include three feature sets. We use a different feature label for each set of chunk features.

1. All heads of base phrases appearing between two protein names
   Similar to word features, these phrase heads are treated as bag-of-word, which means the order of these word is not considered.
2. All chunk heads surrounding the protein name pair

[2] http://maxent.sourceforge.net

[3] The keyword list from Joshua M. Temkin and Mark R. Gilder, 2003 combines keywords from Friedman et al., 2001 and the NIH relevant term list for oncogene expression (NIH, 1999).

These features include n_1 chunk heads to the left of the first protein name and n_2 chunk heads to the right of the second protein name. n_1 is set to be two and n_2 is set to be one as default in our system.

3. All phrase types appear between two protein names.

- **Parse tree**
  Each sentence is parsed by a full-sentence syntactic parser. The path connecting two protein names in the syntactic parse tree is used as a parse tree feature. For example, the parse path between **bovine_prion_protein** and **protein_kinase** in Figure 2 is NP(B)_S_VP_PP_NP_PP.

- **Dependent tree**
  Each internal node of the syntactic parse tree contains a head word. Therefore, the dependent tree is built from the corresponding parse tree of the sentence according to the head words. An example is shown in figure 3 on the same sentence as figure 2. The feature used is as follows.

1. Flag indicates whether one protein name is dependent on the other in the dependent tree.
2. Root information of the sub-dependent-tree
   The root information of the sub-dependent-tree includes the word and POS tag of the root node of the minimum sub-dependent-tree which contains two proteins. For example, "interacts" is the root node of **bovine prion protein** and **protein kinase** in Figure 3.

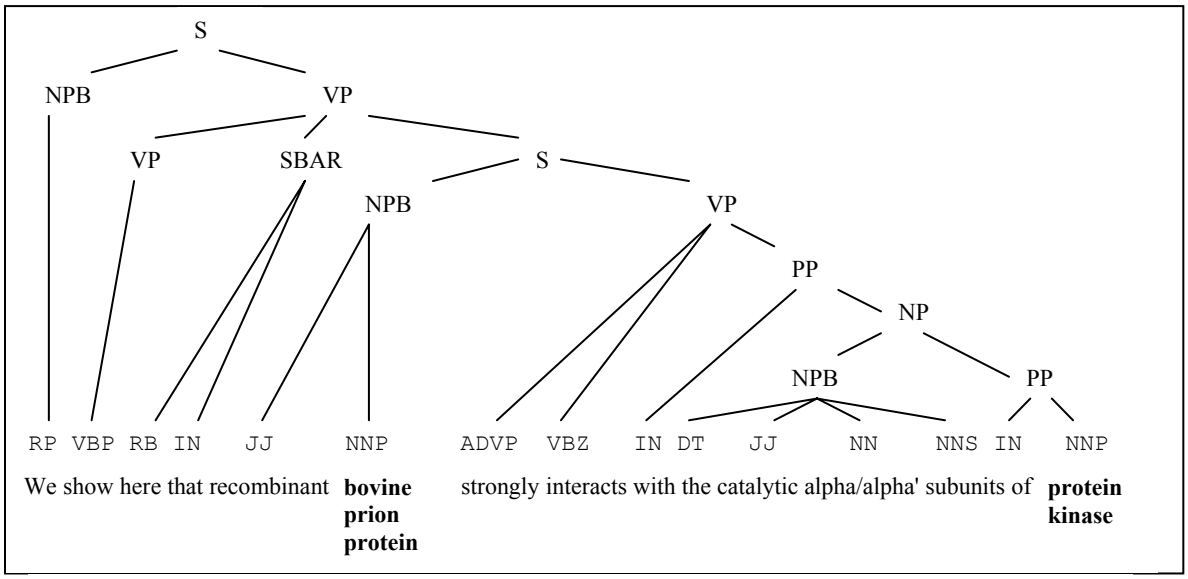

Figure 2. Parse tree for sentence "We show here that recombinant **bovine prion protein** strongly interacts with the catalytic alpha/alpha' subunits of **protein kinase**"

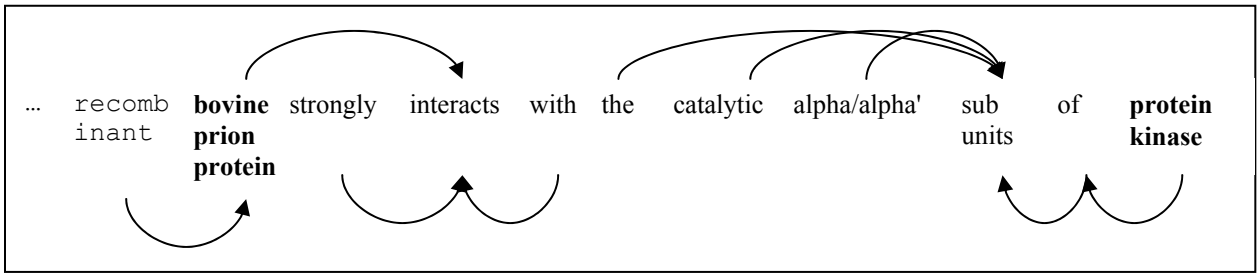

Figure 3. Dependent tree for sentence "We show here that recombinant **bovine prion protein** strongly interacts with the catalytic alpha/alpha' subunits of **protein kinase**"

- **Pair of heads of two protein names**
  The head of each protein is extracted first. Then two head words are combined to form a single word. Features in feature-based classification methods are treated as independent of each other; therefore, we combine two protein names to evaluate them together.

- **Pair of abbreviations of two proteins**
  In order to reduce the data sparseness problem, co-reference resolution module to link different mentions of the same protein could be used. Currently in our experiment, we only try out on the abbreviations. The protein names will be mapped to unique abbreviations correspondingly. Abbreviations of the two protein names are combined as a single string feature. In case where no abbreviation is available, the original name is used.

Here we use the sentence shown in Figure2 and Figure 3 to show the feature vector generated, which is shown as Table 1.

Feature names	**Feature values**
First protein name	p1_bovine, p1_prion, p1_protein
Second protein name	p2_protein, p2_kinase
Words between two protein names	b_strongly, b_interact, b_with, b_the, ...
Left words	l_here, l_that, l_recombine
Right words	r_.
Overlap	ProteinNameInBetween=0
Keyword	Keyword=interacts_between
Chunk heads in between	chunk_head_strongly, chunk_head_interacts, chunk_head_with, chunk_head_alpha/alpha', chunk_head_subunit, chunk_head_of
Surrounding chunk heads	leftChunkHead=here_that, rightChunkHead=interacts
Chunk types in between	ChunkType=ADVP_VP_PP_NP_NP_PP
Parser tree path	PaserPath=NPB_S_VP_PP_NP_PP
Dependent	Dependent=false
Dependent root	DependentRoot=interacts, DependentRootPos=VBZ
Pair of two protein heads	PairOfProteinHead=prion_kinase
Pair of abbreviations	AbbreviationPair=bprp_protein_kinase

Table 1. The feature vector for sentence "We show here that recombinant **bovine prion protein** strongly interacts with the catalytic alpha/alpha' subunits of **protein kinase**."

Words in two names	*	*	*	*	*	*	*	*	*	*
Words between two names	*	*	*	*	*	*	*	*	*	*
Surrounding words		*	*	*	*	*	*	*	*	*
Overlap			*							
Keyword feature				*	*	*	*	*	*	*
Chunk features					*	*	*	*	*	*
Parse tree						*	*	*	*	
Dependent tree							*	*	*	
Pair of proteins								*	*	*
Abbreviation pair									*	*
Recall (%)	80.5	86.1	85.9	86.6	87.2	87.1	87.2	90.1	93.6	93.9
Precision (%)	75.0	81.2	81.1	81.7	83.1	83.0	82.8	85.3	88.0	88.0
F-measure	77.5	83.6	83.3	84.1	85.1	85.0	84.9	87.7	90.7	90.9

Table 2. The performance of different features and their combinations, the last column shows the most effective feature sets and the best performance achieved on the IEPA corpus.

4. Experiments

The data set used in our system is the Interaction Extraction Performance Assessment (IEPA) corpus which is provided by Iowa State University. The corpus is annotated for purpose of corpus property study, there's no system performance reported on it for PPI. It consists of 303 abstracts retrieved from MedLine using ten queries (each query was an AND expression of two biochemical nouns) through PUBMED interface (Ding et al., 2002). Among these abstracts there are 633 positive instances (the protein pairs having interaction relation) and 1080 negative instances (the protein pairs without interaction relation). All protein names are tagged correctly in the IEPA corpus, so that our approach can focus on the relation extraction.

POS tagger used in our experiment is trained on the GENIA corpus with the MedLine abstracts containing POS information using an HMM model (Shen Dan, et al, 2003). Collin's Parser[4] is used to parse the input sentence with POS and protein
names tagged in the corpus. Each dependent tree is generated from the corresponding syntactic sparse tree which is the output of Collin's parser. The abbreviation information is derived from the tagged protein name and bracketed abbreviation behind the full name in the IEPA corpus.

We evaluated our system on the IEPA corpus using 10-fold cross validation and measured the performance using precision/recall/F-measure. The best performance achieved so far is 93.9% recall, 88.0% precision and 90.9 F-score. The results of different features and their combination are shown in Table 2.

5. Discussion

5.1 Effectiveness of different features

- **Surrounding words**
  Nanda, 2004 used only information between two mentions. After analyzing the training data, we find that some information does not occur between two protein names but surrounding two protein names. For example, in the following sentence:

  ```
  Interactions between leptin and NPY
  affecting...
  ```

  If we only consider words between these two protein names, there is only one word "and" occurring in between. It is hard to conclude that **leptin** and **NPY** are interacting with each other. But if we take the surrounding words into account, the word "Interactions" indicates the interaction relation evidently. Therefore, we added surrounding word features and surrounding chunk features of the two protein names into the feature set. In our experiments, the F-measure increased from 77.5 to 83.6 after surrounding words features were added into feature set, which shows the importance of such features.

[4] http://www.ai.mit.edu/people/mcollins/code.html

- **Overlap feature**
  From experiment results, we find that the number of other protein names in between cannot contribute to the performance much. The use of overlap feature decreases recall by 0.2 and precision by 0.1. Therefore, we do not integrate the overlap feature in later experiments.

- **Keyword feature**
  The keyword feature is not as useful as we expected. The use of keyword feature increases F-measure by 0.5 only. The reason could be that the information has already been covered by word features in many cases.
  Yet our approach could find PPI information holding the indicative words *assemble* and *phosphorylation,* which are not in the key word list.

- **Chunk features**
  The chunk features are somewhat useful. Introducing the chunk features to our system increases recall by 0.6%, precision by 1.4% and improves F-measure by 1. Headwords of chunks emphasize significant words that are surrounding or between the two protein names.
  Currently Collin's parser is used with input of POS tagging and protein names. Collin's parser is trained on Penn Tree Bank with Wall Street Journal articles. So we expect that necessary adaptation to MedLine abstracts could make this feature fully effective. Some further experiments on the number of chunks involved may further improve the performance as well.

- **Parse tree and dependent tree features**
  Out of our expectation, the use of parse tree features and dependent tree features deteriorate the performance by 0.1 in F-measure each. One reason could be due to the adaptation problem mentioned earlier for chunk features. Furthermore, Collins' parser does not deal with PP attachment well even on news articles. It will be another effort to further improve the parsing performance. The other reason could be that the IEPA corpus is still not big enough, which leads to the data sparseness problem here.

- **Pair of protein heads**
  The pair of protein heads feature turns out to be very useful in our experiments. It improves F-measure by 2.8. These protein heads may be considered as subtypes of proteins, so that protein names are classified into different categories. These protein categories help to reduce data sparseness problem.

- **Pairs of abbreviations**
  The use of abbreviation pairs gives 3 F-measure improvement. It shows the effectiveness on reducing data sparseness, which encourages us to use more full scale co-reference resolution for PPI extraction in future.

5.2 Error Analysis

Our system achieved an F-measure of 90.9. In order to further evaluate our system and explore possible improvement, we have implemented an error an error analysis. We randomly chose 50 PPIs which are not recognized by our system.

After analyzing, all the 50 errors can be classified into following sources:

1. **Noise in the training corpus** (36%)
   Unlike some relation extraction annotated corpus (e.g. ACE), in which relations are tagged in the texts, the IEPA corpus lists all protein-protein interactions separately from the abstracts. That is, for a protein-protein interaction, we only know two protein names but do not know whether the mentions of the proteins in the sentence are directly related to interaction information. For example in Table 3:

Sentence	Protein 1	Protein 2
However, both EGF and **insulin$_1$** stimulated the accumulation of **phospholipase Cgamma 1** at the actin arc, which was coincident with the EGF receptor in the case of **insulin$_2$** - stimulated cells.	**insulin**	**phospholipase Cgamma 1**

Table 3. A simple examples of IEPA corpus

It is hard to distinguish the protein-protein interaction extracted from protein pair (**insulin$_1$**, **phospholipase Cgamma 1**) or (**insulin$_2$**, **phospholipase Cgamma 1**) unless we re-check it manually. In our experiment, we consider both protein pairs to be correct. Thus, some noisy data is produced and some errors are generated.

2. **Complex sentence structure** (32%)
   Some sentences have very complex structures. In these complex sentences, interacting protein pairs may occur in two sub-sentences and there are many noisy words between them. Therefore, it is difficult to estimate their relation. A better parser could reduce the problem to a certain extent.

3. **Implicit relations** (18%)
   Some protein-protein interactions are not explicitly mentioned in the abstracts. Certain inferences are further needed to get the correct results. For example, in the following sentence,

   NPY in the PVN increases feeding and decreases **uncoupling protein (UCP)** activity in brown fat, whereas **leptin** decreases NPY biosynthesis in the Arc, which presumably decreases PVN NPY.

   There is no direct relation between **uncoupling protein (UCP)** and **leptin**. Certain inferences could be done in the future to find such implicit mention in the future.

4. **Data sparseness** (14%)
   There are 7 out of 50 errors that may be caused by data sparseness.

6. Conclusion

In this paper, we have proposed a supervised learning approach for protein-protein interaction extraction using Maximum Entropy model which achieves promising performance of a 90.9 F-score. We have incorporated various lexical, syntactic and semantic features. We have found that some shallow lexical features, such as words, head of protein names, which are not used before in other existing PPI systems, contribute a large portion of performance improvement. Our approach also has the ability of discovering new phrase patterns / key words, such as *assemble* and *phosphorylation*, which are not in our key word list.

Our approach overcomes the shortcoming of prior work with co-occurrence which can only extract well known interactions reliably. Furthermore, our approach does not suffer from the rule-based approach's inability in incorporating new phrase patterns / key words and yet at the same time it provides better adaptability. Our approach is also able to incorporate the corpus statistics of various features to achieve good performance, without the difficulty in rule-based approaches in inserting additional rules for further performance improvement once the rule set reaches a certain size.

To the best of our knowledge, not only is this the first systematic study of supervised learning, the first attempt of feature-based supervised learning for PPI extraction, but it also provides some useful features, such as surrounding words, key words, abbreviations, so as to extend the supervised learning capability for relation extraction to newswire and other domains.

Acknowledgement

We are grateful to Professor Berleant from Iowa State University of Science and Technology for providing the IEPA corpus.

REFERENCES

[1] Chieu, H.L., and Ng, H.T.(2002). Named Entity Recognition: A Maximum Entropy Approach Using Global Information. *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)* 1:190-203.

[2] Chieu, H.L., and Ng, H.T. (2003). Named Entity Recognition with a Maximum Entropy Approach, *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, 160-163

[3] Craven, M., and Kumlien, J., (1999). Constructing Biological Knowledge Bases by Extracting Information from Text Sources, *Proceeding of the 7th International Conference on the Intelligent System for molecular Biology (ISMB-99),*: 77-86.

[4] Culotta, A, and Sorensen, J. (2004). Dependency tree kernels for relation extraction. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004),* 423-429

[5] Daraselia, N., Yuryev, A., Egorov, S., Novichkova, S., Nikitin and A., Mazo, I.. (2004) Extracting human protein interactions from MEDLINE using a full-sentence parser. *Bioinformatics*. 22;20(5):604-11.

[6] Ding, J., Berleant, D., Nettleton, D., Wurtele, E. (2002). Mining MEDLINE: abstracts, sentences, or phrases? *Pac Symp Biocomput*, 326-37.

[7] Friedman, C., Kra, P., Yu, H., Krauthammer, M. and Rzhetsky, A. (2001). GENIES: a natural language processing system for the extraction of molecular pathways from journal articles, *Bioinformatics*, 17:74-82

[8] Huang, M., Zhu, X., Hao, Y., Payan, D.G., Qu, K., Li, M., (2004). Discovering Patterns to Extract Protein-Protein Interactions from Full

Biomedical Texts. *Bioinformatics*: 20(18): 3604-12.

[9] Koike, A., Kobayashi, Y., and Takagi, T. (2003). Kinase Pathway Database: An Integrated Protein-Kinase and NLP-Based Protein-Interaction Resource. *Genome Res.* 13: 1231-1243.

[10] Leroy, G. and Chen, H. (2002). Automated extraction of medical knowledge using underlying logic from medical abstracts, *Pacific Symposium on Biocomputing*, 350-361.

[11] Marcotte, E.M., Xenarios, I., Eisenberg, D. (2001). Mining Literature for Protein-Protein Interactions. *Bioinformatics*: 17(4):359-63.

[12] Nanda, K., (2004) Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004)*

[13] Nigam, H.K., Lafferty, J., and McCallum, A. (1999). Using maximum entropy for text classification. *IJCAI-99 Workshop on Machine Learning for Information Filtering*, 61-67.

[14] Ono, T., Hishigaki, H., Tanigami, A., and Takagi, T. (2001). Automated extraction of information on protein-protein interactions from the biological literature, *Bioinformatics*, 17(2):155-161.

[15] Ratnaparkhi, A. (1996). A Maximum Entropy Model for Part-of-Speech Tagging. *Proceedings of the Conference on Empirical Methods in Natural Language Processing.*, 133-142.

[16] Rinaldi, F., Schneider, G., Kaljurand, K., Dowdall, J., Andronis, C., Persidis, A., and Konstanti, O., (2004). Mining Relations in the GENIA corpus. *Proceedings of the Second European Workshop on Data Mining and Text Mining for Bioinformatics*: 61-68.

[17] Sekimizu, T., Park, H.S., and Tsujii, J. (1998). Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts. *Genome Inform. Ser. Workshop Genome. Inform.* 9:62–71.

[18] Shatkay, H., Edwards, S., Wilbur, W.J., Boguski, M. (2000). Genes, themes, and microarrays: using information retrieval for large-scale gene analysis, *8th Int. Conf. on Intelligent Systems for Mol. Bio. (ISMB-2000)*, 19-23.

[19] Shen, D., Zhang, J., Zhou, G., Su, J., Tan, C.L. (2003). Effective Adaptation of Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain. *Proceedings of the ACL03 Workshop on Natural Language Processing in Biomedicine,* 49-56.

[20] Stapley, B.J. and Benoit, G. (2000). Biobibliometrics: Information Retrieval and Visualization from Co-Occurrence of Gene Names in Medline Abstracts, *Pacific Symposium on Biocomputing*, 5:529-540.

[21] Temkin, J.M., Gilder, R.M.(2003). Extraction of protein interaction information from unstructured text using a context-free grammar, *Bioinformatics*, 19(16): 2046-2053.

[22] Thomas, J., Milward, D., Ouzounis, C., Pulman, S., Carroll, M. (2000). Automatic Extraction of Protein Interactions from Scientific Abstracts, *Pacific Symposium on Biocomputing*, 5:538-549.

[23] Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research,* 3: 1083–1106.

Address for Correspondence:

Xiao Juan

stuxj@i2r.a-star.edu.sg

Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613.