PROTEIN-PROTEIN INTERACTION: A SUPERVISED LEARNING APPROACH

XIAO JUAN

NATIONAL UNIVERSITY OF SINGAPORE

2005

PROTEIN-PROTEIN INTERACTION: A SUPERVISED LEARNING APPROACH

XIAO JUAN
(*B.SC. (Hons), NUS*)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

November, 2005

Name: Xiao Juan
Degree: M.Sc.
Dept: Computer Science, School of Computing
Thesis Titile: Protein-Protein Interaction: A Supervised Learning Approach

ABSTRACT

In this thesis, we try to explore an effective solution for Protein-Protein Interaction (PPI) extraction, a specific relation extraction (RE) task in bio-literature, through a systematic study using Maximum Entropy model. We explore a rich set of features, including lexical, syntactic and semantic features. Finally, we propose a method with all features integrated via a Maximum Entropy model for PPI. Evaluation on IEPA corpus shows our system achieves 93.9% recall and 88.0% precision. Noting the unique problems in PPI extraction in contrast to existing RE tasks and the lack of current in depth studies in this area, our work finds new insights into PPI extraction. For instance, we explore some features (keyword, protein pairs and protein abbreviations features) hitherto not attempted in other PPI research. Our study also gives us further insight to RE in general, which is still a research area far from mature. For instance, we find the abbreviation feature, which has not been attempted in other feature-based approaches in news domain. Furthermore, comparing to other RE findings, we find that protein pairs, surrounding words and chunk features contribute a large portion of performance improvement.

Keywords: Protein-Protein Interaction, Maximum Entropy Model, Feature-based supervised Learning

ACKNOWLEDGEMENT

I would like to express my great gratitude to Dr. Su Jian, my supervisor Dr. Zhou Guodong and my supervisor A/P Tan Chew Lim, for their advice, guidance and support throughout the duration of my postgraduate study. They have been always accessible and holding discussion and meetings periodically. Their insightful opinions are very important to this thesis.

I also thank the Department of Computer Science, School of Computing, NUS and Institute for Infocomm Research for providing me the opportunity and financial support to study in NUS.

I would like to thank my parents and Mr Qin MingHui for their concern, help and support. Without them, I would never be able to fulfill my study. I also thank my lab-mates Zhang Jie, Yang XiaoFeng, Niu ZhengYu, Chen JinXiu and other friends for their discussion and help.

TABLE OF CONTENTS

SUMMARY

Extracting Protein-Protein Interaction (PPI) from biomedical literature is a difficult but important task for information management and knowledge discovery in biomedical domain. Although more researchers have begun to attempt relation extraction in newswire domain during the last few years, relation extraction in biomedical domain, is ad-hoc and lacks systematic study. Two types of approaches have dominated in this field: co-occurrence based approaches and rule-based approaches. Co-occurrence based approaches depend on co-occurrence information between proteins and can only predict frequently occurring interactions, while rule-based approaches are unable to find PPI embedded in new phrase patterns which are not defined in existing rules. Moreover, it is not easy to use rules to capture the linguistic knowledge optimally. Another problem is that the rules have to be re-written for a subtask/domain of PPI extraction and new relation extraction, which is very time consuming. This thesis will systematically study a particular relation extraction: protein-protein interaction in the biomedical documents. Various lexical, syntactic and semantic knowledge are incorporated and studied systematically by using the feature-based Maximum Entropy Model.

Our work finds new insights into PPI extraction. For instance, we explore some features (keyword, protein pairs and protein abbreviations features) hitherto not attempted in other PPI research. As a result, our system achieves a very promising result of 93.9% in recall and 88.0% in precision on IEPA corpus provided by **Ding et al.**, [2002].

Our study also gives us further insight to relation extraction, which is still a research area far from mature and with quite low performance in ACE task with only 55.5 % F-measure being reported. For instance, we find the abbreviation feature, which has not been attempted in other feature-based approaches in news domain. Furthermore, comparing to other relation extraction findings, we find that protein pairs, surrounding words and chunk features contribute a large portion of performance improvement. We also find that parse tree and dependency tree features, which are useful for other RE , are not useful for PPI extraction.

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Introduction

The living cell is a complex machine that depends on proper functioning of its numerous parts, including proteins. Understanding protein functions and how they interact with each other is the next difficult but important challenge for life science researchers. 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. In this thesis, an interaction between two terms is defined as a direct or indirect influence of one on the quantity or activity of the other. Such interactions include:

1) A increased B.
2) A activated C, and C activated B.
3) A-induced increase in B is mediated through C.
4) Inhibition of C by A can be blocked by an inhibitor of B.

PPI extraction is critical due to the following reasons:

Firstly, PPI indicates how proteins interact with each other. Such information can help understand biological processes, such as DNA replication. In the beginning of this process, protein DnaB and protein DnaC interact with each other and form a DnaB-DnaC protein complex to unwind DNA strands. PPI information is also useful in understanding transcription processes, metabolic pathways, signaling pathways and cell cycle control.

Secondly, interaction between molecules is also important in developing new medicines and treatments to peculiar diseases. PPI extraction is a first step in building protein-protein interaction networks. A comprehensive human protein interaction network will facilitate identification of proteins that can be targeted for therapeutic and diagnostic applications. Understanding biological pathways for normal and disease states will revolutionize medicine in many ways: 1) Creating opportunities for novel therapies for the treatment and prevention of diseases. 2) Providing tailored therapies for individual patients, and 3) Accelerating the drug discovery process.

Finally, PPI extraction has many other important applications, e.g., pathway construction. However, PPI information is still scattered throughout numerous publications. Bringing the relevant information together becomes a bottleneck in the research and discovery process. The volume of such information grows exponentially

with more than 500,000 new articles available online in each year. Although many efforts have been made to create databases that store this information in computer readable form, populating these sources requires manual processing of interpreting and extracting interaction relationships from the biological research literature. Therefore, automatically extracting protein-protein interaction from unstructured text efficiently and accurately would greatly improve the content of these databases and provide a method for managing the continued growth of new literatures.

Although there are huge needs in intelligent information extraction methods to process these large data efficiently and effectively, there are few tools to extract PPI from free text literatures. It is time to begin the adventure now.

1.2 Organization of This Thesis

This thesis is organized as follows. In the next chapter, we review current approaches in PPI extraction and analyze the weaknesses of using relation extraction in newswire domain for PPI extraction. In Chapter 3, we introduce the Maximum Entropy Model and detail various features explored in our system. Finally we report our experiments in Chapter 4, followed by a discussion in Chapter 5 and a conclusion in Chapter 6.

Chapter 2

RELATED WORK

2.1 PPI Systems

The input to a PPI extraction system is a set of texts in the biomedical domain. Because of the availability of the large MEDLINE database, the input usually is a set of MEDLINE abstracts. The output is a set of filled templates. Each template contains a pair of slots which are filled by protein names. The relationship between these two proteins may be presented in the template.

Another issue worth noticing is that different systems define different scopes on PPI extraction. For example, consider the sentence "We studied the interaction of protein A and protein B". The ProteinA–proteinB interaction in this sentence is not considered in some systems, because this sentence does not indicate an experimental result. We adopt a two step approach. The first step extracts all protein pairs. The second step classifies the protein pairs, to indicate whether they interact or not. We

will leave extracting more interaction information, such as interaction types and direction to the future work.

In this chapter, we first discuss the differences between PPI and other relation extractions. Second, we summarize the approaches used in current PPI extraction systems, which fall into two major categories: co-occurrence based approaches and rule-based approaches. Third, because PPI can be viewed as a relation extraction task from biomedical documents, we give a brief summary of current relation extraction works in the newswire domain. Finally, we provide justification on the choice of method: feature-based machine learning method.

2.2 The Differences between PPI and Other Relation Extraction in General

Although PPI extraction can be viewed as a relation extraction task in the biomedical domain, PPI extraction has its own characteristics. PPI extraction is a more challenging problem than traditional relation extraction due to the following reasons:

1. Complicated Sentence Structure in Biomedical Literature

Firstly, compared to traditional information extraction in newspaper articles, the sentence structure in biomedical papers is more complicated. Sentences in biomedical papers (e.g. MEDLINE abstracts) are longer and more complicated than sentences in newswire domain. As an example, we randomly chose 1000 sentences from ACE corpus in the newswire domain and 1000 sentences from IEPA corpus in the biomedical domain. We find that the average sentence length in ACE is 25.7 words while the average sentence length of IEPA is 35.0 words.

It is found that most of relations in the newswire domain are local. Table 1 shows that about 70% of relations exist where two mentions are embedded within each other or separated by at most one word, as shown in the ACE corpus (**Zhou et al.**, [2005]).

# of words	0	1	2	3	4	5	>=6	Over all
# of relations	4163	2693	569	559	463	265	1118	**9830**

Table 1: Distribution of relations over the number of the words in between on a part of ACE corpus

However it is not the case in biomedical domain. The distance between two interacting protein names in IEPA corpus varies widely. We found that about 70% of relations exist where the two mentions are separated by more than five words in

the IPEA corpus. Table 2 shows the distance distribution between two interacting protein pairs in IEPA.

# of words	0	1	2	3	4	5	6	7	8	9	>=10	overall
# of relations	0	62	52	41	31	34	41	36	39	28	265	**629**

Table 2: Distribution of relations over the number of interacting words on the training set

2. Lack of adapted specified NLP tools in biomedical domian

Although many natural language processing tools have been used to solve PPI problem, most of them (e.g. POS tagger, chunking and full parser) are trained on the general/newswire domain. This badly affects PPI extraction much more than relation extraction from the newswire domain. For example, a good full parser will enable the PPI extraction system to find useful information much more reliably, e.g., the dependency information between all the phrases in the sentence and the authors' view represented by embedding. This is due to the dramatic performance drop of a full parser when it is applied to a new domain/task.

3. Lack of benchmark corpus

Unlike newswire domain, there is no benchmark corpus for PPI extraction task. With a benchmark PPI corpus, different approaches can be compared and the advantages/disadvantages of different approaches can be studied well. In this way,

better approaches can be proposed. Many IE researches have benefited much from benchmark corpora MUC and ACE for relation extraction on the newswire domain. Due to the lack of a benchmark PPI corpus, current PPI extraction systems are tested on corpora prepared by individual researcher groups. This makes it difficult to compare different PPI extraction systems directly.

2.3 Current Protein-Protein Interaction Extraction Approaches

Most of current PPI work can be divided into two categories: Co-occurrence-based approach and rule-based approach. There are also some PPI related works that incorporate machine-learning methods. We will introduce co-occurrence based approaches in section 2.3.1 and rule-based approaches in section 2.3.2 respectively while other approaches are summarized in section 2.3.3.

2.3.1 Co-occurrence-based Approaches

The co-occurrence-based approach depends on extraction of co-occurrences between protein names from MEDLINE documents to predict their interaction. For example, **Stapley et al.** [2000] proposed a system to extract gene interaction information by

using co-occurrence statistics between two genes. The premise of their work is that, if two genes have a related biological function, the two gene names or their aliases within the biomedical literature may co-occur.

One problem with this approach is that it can only extract frequently occurring PPIs but may not be able to find new emerging and/or less frequently occurring PPIs. Another problem is that it fails to determine protein interaction types.

2.3.2 Rule-based Approaches

Most of current PPI extraction systems use rule-based approaches. The rule-based approach uses templates that match specific linguistic structures to recognize and extract protein interaction information from MEDLINE documents. To generate meaningful templates, the text unit of rule-based approaches is often a sentence or a phrase. Following are some of representative systems:

Sekimizu et al. [1998] firstly collected frequently occurring verbs from MEDLINE sentences which contain PPI pairs. Then they used partial parsing techniques to extract noun phrases from sentences. Finally they developed rules to find the subject and object of the high-frequency verbs. They tested their system on some abstracts extracted from MEDLINE using keyword “protein binding” and certain protein

names. They claimed that their method could achieve precision at 73% with recall missing.

Thomas et al. [2000] used a statistical parser and manually generated rules to fill templates with information on proteins and their interactions. They concentrated on three verb phrases (interact with, associate with, bind to) for which they developed templates. They calculated recall and precision in four different manners for three sets of abstracts with recall ranging from 24% to 63% and precision from 60% to 81%.

PIES [Wong, 2001] required users to submit key terms, such as "calyculin," and searched Medline for abstracts containing these terms. From the matching abstracts, "inhibit" and "activate" interactions were considered. They used BioNLP to extract the relevant information from the sentences, and the Graphviz software package to visually display the results. In their system users could save and update the retrieved information. There was no evaluation data provided.

Ono et al. [2001] proposed a manually written rule-based approach to extract protein-protein interaction in a single sentence. In their work, they identified protein names in the literature using a protein name dictionary, which was constructed manually. A sentence that contains at least two proteins was parsed with simple part of speech rules. Then the sentence was parsed using a simple pattern-matching rule to recognize the PPI. They tested their system on sentences that contained at least two protein names and one of the keywords such as: 'interact', 'associate', 'bind', and 'complex'. About 1500 sentences were tested with the overall recall at 85% and the overall

precision at 92%.

Leroy and Chen [2002] extracted PPIs from individual sentences using preposition-based rules with a set of pre-defined prepositions such as "of" and "by". They claimed that building templates around prepositions was able to capture more information than only looking for particular genes. 50 new abstracts containing keyword "E2F1" (a gene name) were tested. The average precision of all types of templates was 70% and the average recall was 47%.

MedScan [Daraselia et al., 2004] utilized manually written rules to extract human protein interactions from MEDLINE based on full-sentence parsing. It was examined on 1.2 million MEDLINE abstracts which contained at least one notation of human protein and were successfully parsed by full-sentence parser. They manually reviewed 361 randomly extracted protein interactions and concluded that 91% of them were correct. Then they estimated recovery rate by the manual analysis of 91 randomly selected sentences from 43 abstracts containing PPI, and the recall was found to be 21%.

The above rule based approaches using different heuristics, being tested on the corpus developed by the individual groups themselves, quite different performances are reported. It is not clear how well the problem has been solved and what kinds of linguistic knowledge are useful for the solution

Besides there are some limitations with rule-based approaches:

Firstly, it's unable to find PPI embedded in new phrase patterns which are not defined in the existing rules. Secondly, it's not easy to use rule to capture the linguistic knowledge in an optimal way. Thirdly, there are many subtasks of PPI extraction, such as, transcription factors interaction extraction and human protein interaction extraction. And there're other relations between bio-molecules in the bio-literature. The rules have to be rewritten for a new subtask/domain of PPI extraction and new relation extraction, which is very time-consuming.

2.3.3 Other Approaches

Although some supervised learning approaches have been reported in the bio-literature (**Huang et al** [2004], **Craven and Kumlien** [1999], **Marcotte et al.** [2001] and **Palakal et al**. [2002]), none of them systematically study the protein-protein interaction extraction task.

Huang et al. [2004] generated some POS patterns using some corpus statistics and evaluated their system on a set of sentences which contain certain interaction verbs (e.g. inhibit).

Craven and Kumlien [1999] used a sentence classification approach for subcellular-location relations in a sentence. **Marcotte et al.** [2001] utilized a Bayesian approach to decide whether or not a given biomedical paper discusses protein-protein interactions. Such a text / sentence classification based approach is not suitable for

PPI extraction. as there could be more than two proteins exits in a sentence, which two proteins holding interaction relation still need to be determined.

Palakal et al. [2002] used HMM to decide the direction of a given PPI.

In summary, little work has been done for supervised PPI extraction. In the next section, I will review supervised relation extraction in news, a general domain.

2.4 Relation Extraction from Free Text in the General Domain

Protein-protein interaction describes an interaction relation between a pair of proteins. In this regard, PPI extraction is a special relation extraction task. In this section, we look into related work on relation extraction. Other background information on information extraction is provided in Appendix I.

2.4.1 Introduction

The task of relation extraction was introduced as a part of the Template Element task in MUC6 formulated as the Template Relation task in MUC7 (MUC. 1987-1998) and

extended in the Automatic Content Extraction[1] (ACE) since 1999. During the last few years, relation extraction in the general domain has begun to attract more and more researchers.

Most work at MUC used rule-based approaches with the exception of **Miller et al.,** [2000]. They augmented syntactic full parse trees with semantic information corresponding to entities and relations, and built generative models for the augmented trees. However, complicated relation extraction tasks may impose a big challenge to the complex modeling approach used by **Miller et al.,** [2000] which integrates various tasks such as part-of-speech tagging, named entity recognition, template element extraction and relation extraction, in a single model.

From then on, various classification-based learning approaches have been explored for relation extraction and achieved good performance. Current classification-based machine learning approaches can be divided into two categories: feature-based classification approaches (**Kambhatla**, [2004]) and kernel-based classification approaches (**Zelenko et al.**, [2003]; **Culotta and Sorensen**, [2004]).

2.4.2 Kernel-based Classification Approaches

In the kernel-based classification approaches, an example is not represented by a feature vector. The kernel-based approaches define a kernel function to compute the

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

similarity between examples. A kernel function is a similarity function satisfying certain properties. More precisely, a kernel function K over the object space X is a binary function K: $X \times X$ -> [0, ∞] which maps a pair of objects x, y to their similarity score $K(x,y)$.

Zelenko et al. [2003] proposed a kernel-based classification approach to extract "person-affiliation" and "organization-location" relations. The first step of their approach is a shallow parser which parses examples into shallow parse trees. The shallow parser also identifies names, noun phrases, and a restricted set of parts of speech in text. The parse tree nodes contain a type and a head (text field). To represent a relation, the nodes get a 'role' field, for example, to capture a person-affiliation relation, one node (the person) gets role = "member" and one node (the organization) gets role = "affiliation". As their kernel, they used a measure of similarity between two trees. Basically, two trees are considered similar if their roots have the same type and role, and each has a subsequence of children (not necessarily consecutive) with the same types and roles. The value of the similarity depends on how many such subsequences exist, and how spread out they are. All the training examples are converted into such shallow parse trees with role labels, and used to train the system. They obtain an F-measure of 87 for person-affiliation relation classification and 83 for organization-location relation classification.

Culotta and Sorensen [2004] used a similar approach as **Zelenko et al.** [2003]'s method and further extended it to estimate kernel functions between augmented

dependency trees. Their method was evaluated on the same corpus as **Kambhatla** [2004]'s method. They compared performance of different kernel functions. The best performance reported in their paper was 45.8 F-measure based on 4 ACE super types which was worse than **Kambhatla** [2004]'s performance.

Kernel-based classification approaches have been successfully applied in many applications such as text categorization and natural language parsing. A unique property of the kernel methods is that we do not need to generate features explicitly. More precisely, an object is no longer a feature vector as it is common in a machine learning algorithm. Instead, objects retain their original representations and are used within learning algorithms only via computing a kernel (similarity) function between them. Therefore, kernel-based approaches are able to explore the implicit feature space without much feature engineering. Yet further research work is still expected to make it effective with complicated relation extraction tasks such as the one defined in ACE.

Another disadvantage of a kernel-based classification approach is that it is computationally slow for practical applications. Moreover, a kernel function is required for each kind of relation for a multi-slot information extraction. For example, a unique kernel function is needed for each relation such as *person-organization*, *organization-location*, and etc. Finally, how to find an optimal kernel function is still an unsolved problem.

2.4.3 Feature-based Approaches

The feature-based approaches rely on feature-based representation of objects. That is, an object is transformed into a collection of features $f_{1,}$…, *fn*, thereby producing an *N*-dimensional vector for each object.

Kambhatla [2004] employed Maximum Entropy models with features including word, entity type, mention level, overlap, dependency tree, parse tree information for ACE relation detection and characterization (RDC) task, which contains 24 relation types in the newswire domain. It achieved a good performance of 52.8 F-measure, which is much better than **Culotta and Sorensen** [2004]'s work.

2.4.4 Our Approach

In this thesis, we try to explore an effective solution for PPI extraction, a specific relation extraction task in bio-literature, through a systematic study using Maximum Entropy model. Most of the previous work on PPI extraction uses rule based with different heuristics. Being tested on the corpus developed by the individual groups themselves, quite different performances are reported. It is not clear how well the problem has been solved and what kinds of linguistic knowledge are useful for the solution. On the other hand, relation extraction being an independent task was first

identified during MUC 7. So far rule-based approaches, statistical modeling, kernel based approaches had been explored for relation extraction from news articles. However, these approaches have their own disadvantages.

Firstly, although most of the state-of-the-art PPI approaches are rule-based, there are some limitations with this approach: 1) It's unable to find PPI embedded in new phrase patterns which are not defined in the existing rules. 2) It is not easy to use rules to capture the linguistic knowledge in an optimal way. 3) There are many subtasks of PPI extraction, such as, transcription factors interaction extraction and human protein interaction extraction. And there're other relations between bio-molecules in the bio-literature. The rules have to be rewritten for a new subtask/domain of PPI extraction and new relation extraction, which is very time-consuming.

Secondly, complex modeling approach integrates various tasks such as part-of-speech tagging, named entity recognition, template. Therefore, complicated relation extraction tasks may impose a big challenge to this approach.

Thirdly, compared to feature-based approach, kernel-based classification approach is much slower. Furthermore, feature-based classification approach is more flexible, e.g., feature weights can be learned. Literature shows that feature-based classification approaches can achieve better performance than kernel-based approaches, especially on complicated relation extraction tasks such as ACE RDC task.

Due to the shortcomings of the above approaches, and inspiring by the relative success of **Kambhatla** [2004], we want to use maximum entropy based approach to do a systematic study on PPI extraction. We would like to find out how such a model can work on PPI extraction task. And through the systematic feature engineering, we try to answer what kind of linguist knowledge in what way is useful for PPI extraction task. Such a study would also give us further insight to relation extraction in general, which is still a research area far from mature and with quite low performance in ACE task with only 55.5 % F score being reported.

Chapter 3

MAXIMUM ENTROPY BASED PPI EXTRACTION

In this chapter, firstly we introduce the maximum entropy model used in our system. Then we introduce various features explored in thesis respectively.

3.1 Maximum Entropy

Maximum Entropy is a probability distribution estimation technique which was first introduced to NLP by **Berger et al.,** [1996] and **Della Pietra et al.** [1997]. Since then, Maximum Entropy technique has been widely used in recent years for various natural language processing tasks, such as part-of-speech tagging (**Ratnaparkhi** [1996]), text classification (**Nigam et al.** [1999]) and named entity recognition (**Chieu and Ng.** [2002]). **Kambhatla** [2004] first introduced Maximum Entropy Model for relation extraction on ACE corpus.

To use Maximum Entropy Model, the task must be re-formulated as a classification problem, in which the task is to observe some linguistic observation or history $h \in H$ and predict the correct outcome class $o \in O$. This involves constructing a classifier $\phi : H \to O$,which in turn can be implemented with a conditional probability distribution *p*, such that $p(o \mid h)$ is the probability of outcome class *o* given some history *h*. Maximum Entropy Model is estimated according to maximum entropy principle which tries 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.

To fulfill this maximum entropy principle, maximum entropy model tries to maximize the entropy of a probability distribution to certain known constrains/information. Here, the entropy of the probability distribution *p* is defined as :

$$H(p) = -\sum_{h \in H, o \in O} p(o \mid h) \log p(o \mid h) \qquad (1)$$

The maximum entropy model is defined over $H \times O$, where H is the set of all possible features or "history", and O is the set of possible outcomes. The probability $p(o \mid h)$ is estimated as follows:

$$p(o \mid h) = \frac{1}{Z(h)} \prod_{j=1}^{k} \alpha_j^{f_j(h,o)} \qquad (2)$$

where $Z(h)$ is a normalization function. $\{f_1, f_2, ..., f_k\}$ are feature functions and $\{\alpha_1, \alpha_2, ..., \alpha_k\}$ are the model parameters. Each parameter corresponds to exactly one feature function and can be viewed as a "weight" for that feature. All feature functions used in the maximum entropy model are binary (0 or 1):

$$f_j(h,o) = \begin{cases} 1, & if o = o_i, h = h_k \\ 0, & otherwise \end{cases} \quad (3)$$

Therefore, the joint probability of a history *h* and outcome class *o* is determined by those parameters whose corresponding features are active or presented, i.e., those α_j such that $f_j(h,o)$=1. The training process of Maximum Entropy model is to find values of all model parameters, while the predicting process is to compare the joint probabilities of history and different outcome classes. The model parameters for the distribution *p* are obtained via *Generalized Iterative Scalings* [**Darroch** and **Ratcliff**, 1972] .

In our PPI task, *o* is either true or false indicating whether the current protein pair has interaction relationship, h is an element of observation vector, $f_j(h,o)$ is a binary feature function given the element of observation vector *h* and outcome class *o*. Following is an example of a binary feature function given an observation "keyword=inhibit" with the outcome class "true". That is, "keyword=inhibit" indicates the existence of the interaction relation in the training data.

$$f_j(h,o) = \begin{cases} 1, & if \quad o = true, keyword = inhibit; \\ 0, & otherwise. \end{cases} \tag{4}$$

The detailed description of Maximum Entropy Model can be found in **Berger et al.** [1996][2]. We have used the open NLP maximum entropy package[3] in our system.

3.2 Features

In this thesis, we explore various features to capture lexical, syntactic and semantic information and examine the effect of these features.

- **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 in two protein names

     These features include all words that appear in two protein names. For example, if the name of protein is "**bovine prion protein**", words appear in this protein name are "**bovine**", "**prion**" and "**protein**".

  2. Words between two protein names

[2] http://www.ai.mit.edu/courses/6.891-nlp/J1996-1002.pdf
[3] http://maxent.sourceforge.net

These features include all words that are located between two protein names. If no word appears between two protein names, "NULL" is returned.

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" is returned.

All words are treated as bag-of-word. That is, the order of these words in a category is not considered.

- **Overlap**

This category of features includes:

1. Number of protein names in between;

This feature counts all protein names that are located between two protein names.

2. Number of words in between.

This feature counts all words that are located 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* left words of the first protein name;

3. within *n* right words of the second protein name. (*n* is set to be three in our experiment.)

In our experiment, the keyword list with **Temkin and Gilder** [2003][4] is used for this feature.

- **Chunks**

It is well known that chunking plays a critical role in the Template Relation task of the 7th Message Understanding Conference (MUC-7 1998). The related work mentioned in Section 2 extended to explore the information embedded in the full

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

parse trees. In this thesis, we separate the features of base phrase chunking from those of full parsing. In this way, we can separately evaluate the contributions of base phrase chunking and full parsing. Here, each sentence is parsed by a partial parser to capture text chunking information of training examples. Our system differentiates three sets of chunk features.

1. All head words of base phrases between two protein names

   Similar to word features, these phrase heads are treated as bag-of-word.

2. All chunk heads surrounding the protein name pair

   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. According to results of our experiments, n_1 is set to be two and n_2 is set to be one.

3. All phrase types appear between two protein names.

- **Parse tree**

A parse tree represents the syntactic structure of a string according to some formal grammar. In our system, each sentence is parsed by a full-sentence syntactic parser. This category of information concerns about features inherent only in the full parse tree.

The path of phrase labels (removing the duplicates) connecting the two protein names in the parse tree. For example, the path of phrase labels between **bovine_prion_protein** and **protein_kinase** in Figure 1 is NP_S_VP_PP_NP_PP.

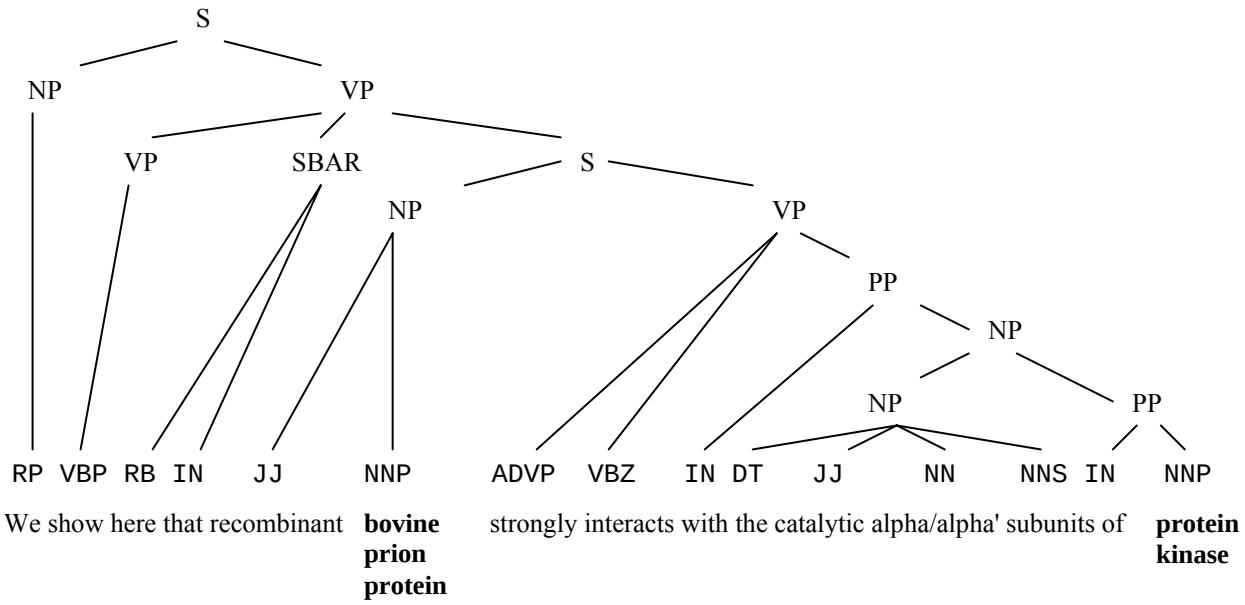

Figure 1. Parse tree for sentence "We show here that recombinant **bovine prion protein** strongly interacts with the catalytic alpha/alpha' subunits of **protein kinase**"

- **Dependency tree**

Each internal node of the syntactic parse tree contains a head word. Therefore, the dependency tree is built from the corresponding parse tree of the sentence according to the head words. A parse tree with head words on the sentence "We show here that recombinant **bovine prion protein** strongly interacts with the catalytic alpha/alpha'

subunits of **protein kinase**" is shown in Figure 2 and an example of dependency tree derived from Figure 2 is shown in Figure 3. This category of features includes:

1. Flag indicates whether one protein name is dependent on the other in the dependency 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.

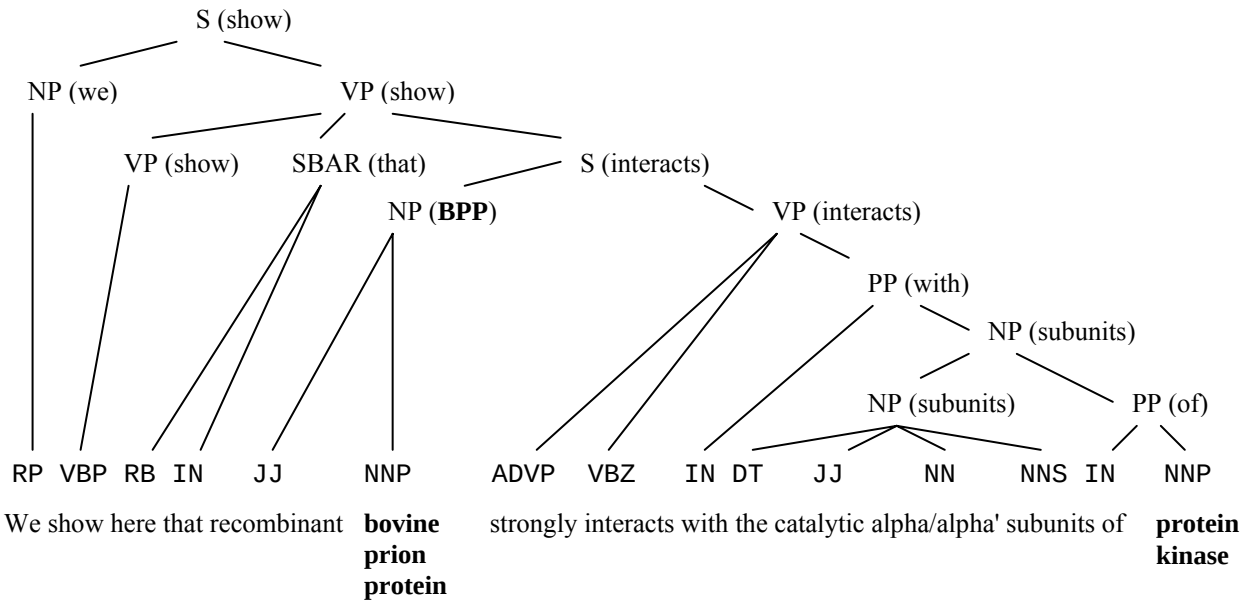

Figure 2. Parse tree with node heads for sentence "We show here that recombinant **bovine prion protein** strongly interacts with the catalytic alpha/alpha' subunits of **protein kinase**"

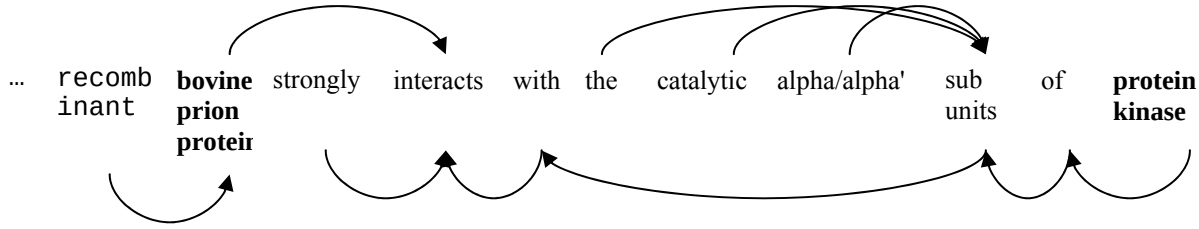

Figure 3. Dependency tree for sentence "We show here that recombinant **bovine prion protein** strongly interacts with the catalytic alpha/alpha' subunits of **protein kinase**"

- **Pair of heads of two protein names**

  Firstly, the head of each protein name is extracted by a set of manually written rules which is based on words and corresponding POS tags in protein name. Then two head words are combined to form a single word. Since features in feature-based methods are treated as independent of each other, we combine two protein names to evaluate the integration effect between them.

- **Pair of abbreviations of two proteins**

  In order to reduce the data sparseness problem, co-reference resolution module is used to link different mentions of the same protein.

  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 feature. In case where no abbreviation is available, the original name is used.

Table 3 shows the feature vector generated for the example sentence "We show here that recombinant **bovine prion protein** strongly interacts with the catalytic alpha/alpha' subunits of **protein kinase**.". Please also refer to Figure 1-3 for more details.

Feature names	Feature values
First protein name	p1_bovine, p1_prion, p1_protein
Second protein name	p2_protein, p2_kinase
Words in between	b_strongly, b_interact, b_with, b_the, ...
Left words	l_here, l_that, l_recombine
Right words	r_.
Overlap	ProteinNameInBetween=0, WordInBetween=8
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 3. The feature vector for sentence "We show here that recombinant **bovine prion protein** strongly interacts with the catalytic alpha/alpha' subunits of **protein kinase**."

Chapter 4

EXPERIMENTATION

4.1 Dataset

Our system uses the Interaction Extraction Performance Assessment (IEPA) corpus provided by Iowa State University. The corpus consists of 303 abstracts retrieved from MedLine using ten queries through PUBMED interface. Each query was an AND expression of two protein names which are discovered to interact with each other (**Ding et al.** [2002]). Then all sentences with at least two protein names are extracted as training set (1136 sentences in total). Among these sentences, 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 pre-tagged correctly in the IEPA corpus, so that our approach can focus on the relation extraction task.

4.2 Experiment Results

POS tagger is trained on the GENIA corpus with the MedLine abstracts containing POS information using an HMM model (**Shen et al.**, [2003]). A HMM-based chunking engine (**Zhou et al.,** [2000]) is used to get partial parsing information. Collin's Parser[5] is used to build parse tree for each input sentence with POS and protein names tagged in the corpus. Each dependency 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 IEPA corpus using 10-fold cross validation and precision/recall/F-measure:

$$precision = \frac{\#TruePositive}{\#TruePositive + \#FalsePositive} \quad (1)$$

$$recall = \frac{\#TruePositive}{\#TruePositive + \#FalseNegative} \quad (2)$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (3)$$

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

The best performance achieved so far is 93.9% recall, 88.8% precision and 90.9 F-score. Table 4 shows the effect of the results of different features and their combinations from simple to complex features. Table 5 also shows the effectness of each feature by excluding one feature only.

Words in two names	*	*	*	*	*	*	*	*	*	*
Words between names	*	*	*	*	*	*	*	*	*	*
Surrounding words		*	*	*	*	*	*	*	*	*
Overlap			*							
Keyword feature				*	*	*	*	*	*	*
Chunk features					*	*	*	*	*	*
Parse tree						*	*	*	*	
Dependency tree							*	*	*	
Pair of protein heads								*	*	*
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 4. The performance of different features that were added into feature set in an incremental way. The last column shows the most effective feature sets and the best performance achieved on IEPA

Features	R	P	F	Features	R	P	F
All features	93.8	87.5	90.5				
All features - Words in two names	92.3	87.9	90.0	All features - Chunk features	92.7	87.8	90.1
All features - Words in between	92.1	87.5	89.7	All features - Parse tree	93.7	88.0	90.7
All features - Surrounding words	91.8	86.8	89.2	All features - Dependency tree	94.0	88.0	90.8
All features - Overlap	93.6	88.0	90.7	All features - Pair of protein heads	92.0	86.1	89.0
All features - Keyword feature	93.5	87.3	90.3	All features - Abbreviation pair	90.1	85.2	87.6

Table 5. Experiment result of experiments in exclusive way.

From the experiment results we find that abbreviation pair feature, surrounding word feature and pair of protein heads feature contribute most. We also find overlap feature, parse tree feature and dependency tree feature decrease the performance. In the next section, we will examine our PPI system in more details.

Chapter 5

DISCUSSION

5.1 Comparisons with Other Systems

Since there is no previous work on the IEPA corpus, direct comparison is impossible. To solve this problem, we have chosen the most similar work as in **Huang et al., [2004]** which achieved the state-of-the-art performance. Because we don't know the exact implementation of their method, we have tried our best to include same information as in **Huang et al., [2004]** (i.e. word features and POS features) and used our feature-based model to extract PPIs on the IEPA corpus. We have tried different feature representations and Table 6 shows the best performance. It shows that our system outperforms Huang's re-implemented system by more than 12 in F-measure. Although such comparison is indirect, it can still provide some indications about the superiority of our system.

Recall	Precision	F-measure
79.5	77.2	78.3

Table 6. Experiment result for re-implemented system as in **Huang et al.** [2004].

5.2 Effectiveness of different features

- **Surrounding words**

Kambhatla [2004] used only information between two mentions. After analyzing the training data, we find that sometimes surrounding words also contain very important infomation. 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 based on this information. However, 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 in an incremental way as in Table 4. Moreover, in an exclusive way as in Table 5, the absence of surround words features decreases the performance by 0.8 F-measure, which shows the importance of such features.

- **Overlap feature**

  From experiment results, we find that the number of other protein names in between does not contribute to the performance much. The use of overlap feature decreases recall by 0.2 and precision by 0.1. In an exclusive way as in Table 5, the absence of overlap feature increases overall performance by 0.2 F-measure. Therefore, we do not integrate the overlap feature in later experiments, although such features do increase the performance in the newswire domain (**Kambhatla** [2004]; **Zhou et al.**, [2005]). This may be due to that most of relations in the newswire domain are local while IEPA corpus is much more complicated. Therefore, the overlap feature in biomedical domain is not as useful as in newswire domain.

- **Keyword feature**

  The keyword feature is not as useful as we expected. Keyword feature only increases F-measure by 0.5. The reason may be that related information has most been covered by word features.

- **Chunk features**

  The chunk features are somewhat useful. They increase recall by 0.6%, precision by 1.4% and F-measure by 1. If chunk features are removed from the whole feature set, the performance drops from 90.5 to 90.1 F-measure.

- **Parse tree and dependency tree features**

  Out of our expectation, the use of parse tree features and dependency tree features deteriorate the F-measure by 0.1 each.

  One reason could be due to the adaptation problem. 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 more effective.

  Furthermore, Collins' parser does not deal with PP attachment well even on news articles. Although a full parser can provide more detailed information than a partial parser, there are some limitations of current full sentence parsers: 1) Full parsers in general tend to be slow because they handle the full possible structure of whole sentences. 2) It is often argued that the results of full parsers have more ambiguity than that of partial parser. 3) Full parsers are more error-prone than partial parsers. 4) The performance of a full parser drops dramatically when it is

applied to a new domain/task. Therefore, much research can be done towards improving a full parser. The use of a good full parser will enable our system to find much more useful information, e.g., the dependency between all the phrases in the sentence and the authors' view represented by embedding, in a more straightforward way.

One more possible reason could be that the IEPA corpus is not big enough, which leads to the data sparseness problem.

Another reason is that our parse features may not represent parse tree structure well. In the future work, we may explore a more effective scheme to make use of it.

- **Pair of protein heads**

The pair of protein heads feature turns out to be very useful in our experiments. In an incremental way as in Table 4, it improves F-measure by 2.8. In an exclusive way as in Table 5, the absence of overlap feature decreases overall performance by 1.5 F-measure.

- **Pairs of abbreviations**

  Abbreviation pairs improve F-measure by 3. The absence of overlap feature decreases overall performance by 2.9 F-measure. It shows the effectiveness on reducing data sparseness, which encourages us to explore more effective co-reference resolution for PPI extraction in the future.

Through the systematic feature engineering, we find that abbreviation pair feature, surrounding word feature and pair of protein heads feature contribute most. We also find overlap feature, parse tree feature and dependency tree feature decrease the performance.

We find that keyword, protein pairs and protein abbreviations features which are not used by other PPI extractions before are very useful.

Our experiment results give us further insight to relation extraction in general. For instance, we find the abbreviation feature, which has not been attempted in other feature-based approaches in news domain.

Furthermore, comparing to other RE findings, we find that protein pairs, surrounding words and chunk features contribute a large portion of performance improvement. We also find that parse tree and dependency tree features are useful for other RE yet not useful for PPI extraction.

5.3 Error Analysis

Our system achieved the F-measure of 90.9. In order to further evaluate our system and explore possible improvement, we have implemented an error analysis by randomly chosen 50 missing PPIs and classified them into following sources:

1. **Noise in the training corpus** (36%)

   Unlike the annotated corpus (e.g. ACE), in which relations are tagged in the texts, IEPA corpus only 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 which mentions of the proteins in the sentence are directly related to interaction. For example in Table 7:

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 7. A simple example 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 link the interaction manually. In our experiment, we simply regard both protein pairs to be positive. Such simple approach inevitably introduces some noise and errors to our training data.

To avoid this kind of errors, manual evaluation is worthwhile.

2. **Complex sentence structure** (32%)

Some sentences have very complex structures. Therefore, it is difficult to extract contained PPI relation. A better parser could reduce the problem to a certain extent. Following is an example:

> In addition, **glycosyl-phosphatidylinositol-specific phospholipase C** (GPI-PLC), which in isolated rat adipocytes is activated by insulin, was stimulated to up to 5-fold by glucose and 10-fold by glucose plus **insulin** in both yeast spheroplasts and intact cells leading to a concentration-dependent leftward shift of the glucose-response curve for activation of the GPI-PLC

Although there are many shared parsers in general domain, the performance of a full parser drops dramatically when it is applied to a new domain/task. Therefore,

much research can be done towards improving a full parser in a biomedical domain.

3. **Implicit relations** (18%)

Some protein protein interactions are not explicitly mentioned in the abstracts, certain inferences may be 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**.

To reduce this kind of errors, it would be good in the corpus annotation to separate explicit and implicit interactions. Use only explicit mentions for training up and use additional inference model to derive implicit interactions.

4. **Data sparseness** (14%)

To reduce this kind of errors, it is worthwhile to explore computational methods to reduce the data sparseness problem, eg, exploring the knowledge embedded in the large un-annotated corpus.

Chapter 6

CONCLUSION and FUTURE WORK

6.1 Conclusion

In this thesis, we systematically study a particular relation extraction: protein-protein interaction in the biomedical documents. Moreover, we propose 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 explored various lexical, syntactic and semantic features. We have found that some shallow lexical features, such as head of protein names, protein abbreviations and keywords which have not used before in other existing PPI systems, contribute a large portion of performance improvement. We have found the abbreviation feature which has not been attempted in other feature based approaches in news domain. Although parse tree and dependency tree features are reported useful in RE on news domain, they decreased performance in our experiments.

6.2 Our Contributions

Our contribution to the research in protein-protein interaction can be concluded as follow.

Firstly, we systematically study a particular relation extraction: protein-protein interaction in the biomedical documents.

Secondly, we build a relation extraction engine based on Maximum Entropy model which incorporates various lexical, syntactic and semantic features to extract PPI from biomedical literature. To the best of our knowledge this is the first systematic study of feature-based supervised learning for PPI extraction. Our approach overcomes shortcomings of co-occurrence based approaches and rule-based approaches. It achieves a very encouraging result of 93.9% recall and 88.0% precision on IEPA corpus by **Ding et al.**, [2002].

Thirdly, our work finds new insights into PPI extraction. For instance, we explore some features (keyword, protein pairs and protein abbreviations features) hitherto not attempted in other PPI research.

Finally, our study would also give us further insight to relation extraction in general. We find the abbreviation feature, which has not been attempted in other feature-based approaches in news domain. Furthermore, comparing to other RE findings, we find

that protein pairs, surrounding words and chunk features contribute a large portion of performance improvement.

6.3 Future Works

Our future work will focus on reducing errors found in our error analysis.

As shown in error analysis, there are some errors due to noise in the corpus. One possible solution is to build a benchmark PPI corpus. The benchmark corpus should mark exact positions of interacting proteins in the sentence. With a benchmark PPI corpus, different approaches can be compared and the advantages/disadvantages of different approaches can be well studied. In this way, better approaches can be proposed. Many researches have benefit much from benchmark corpus e.g., MUC for IE and Penn Treebank for parsing. A good PPI benchmark corpus will benefit all machine-learning-based PPI extraction systems.

We plan to apply our engine for other relation extraction tasks as well as other sub-domain in biomedical domain. One problem is the availability of a large annotated corpus. One possible solution is unsupervised or semi-supervised methods, e.g. co-training to reduce human effort as much as possible.

A good domain-specific parser will enable our system to find much more reliable information, e.g., the dependency between all the phrases in the sentence and the

authors' view represented by embedding, in a more straightforward way. Therefore, adapting a good general parser to the biomedical domain is worthwhile.

The last aspect of our future work is adopting more methods to extract implicit mentions of PPI. Our current system only extracts explicit mentions of PPI. To extract implicit mentions of PPI effectively, we need to explore other approaches, such as inference resolution.

6.4 Dissemination of Results

This thesis presents a coherent work on the explorations of the protein-protein information extraction. The work on the maximum entropy model, feature set, experiment result and discussion is covered in our paper published in the *First International Symposium on Semantic Mining in Biomedicine (SMBM-2005)* (**Xiao et al.,** [2005]).

References

Agichtein and Gravano [2000] E. Agichtein and L. Gravano. (2000). "Snow-ball: Extracting Relations from Large Plain-text Collections". *Proceedings of the Fifth ACM International Conference on Digital Libraries.* 85-94

Berger et al. [1996] A. Berger, S. Della Pietra, V. Della Pietra. (1996). "A Maximum Entropy Approach to Natural Language Processing", *Computational Linguistics*. 22(1):39-71.

Califf and Mooney [1999] M. Califf and R. Mooney. (1999) "Relational learning of pattern-match rules for information extraction". *Proceedings of the Sixteenth National Conference on Artificial Intelligence(AAAI-99),*328–334.

Chieu and Ng [2002] H.L. Chieu and H.T. Ng. (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.

Chieu and Ng [2002] H.L. Chieu and H.T. Ng, Hwee Tou (2002). "A Maximum Entropy Approach to Information Extraction from Semi-Structured and Free

Text". *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*. 786-791.

Ciravegua [2001] F. Ciravegna. (2001) "Adaptive Information Extraction from Text by Rule Induction and Generalization". *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001).* 1251-1256

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

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

Daraselia et al. [2004] N. Daraselia, A. Yuryev, S. Egorov, S. Novichkova, A. Nikitin and I. Mazo. (2004) "Extracting human protein interactions from MEDLINE using a full-sentence parser". *Bioinformatics*. 22;20(5):604-611.

Ding et al. [2002] J. Ding, D. Berleant, D. Nettleton, E. Wurtele. (2002). "Mining MEDLINE: abstracts, sentences, or phrases?" *Proceedings of Pacific Symposium on Biocomputing*, 326-37.

Freitag and McCallum [1999] D. Freitag and A. McCallum. (1999) "Information extraction with HMMs and shrinkage". *In Proceedings of the sixteenth national Conference on Artificial Intelligence (AAAI-99) Workshop on Machine Learning for Information Extraction,* 31-36

Freitag [1998] D. Freitag. "Information extraction from HTML: application of a general machine learning approach". *In Proceedings of the Fifteenth national Conference on Artificial Intelligence (AAAI-98)*. 517--523

Freitag and Kushmerick [2000] D. Freitag and N. Kushmerick, "Boosted wrapper induction". *Proceedings of the Sixteenth national Conference on Artificial Intelligence (AAAI-2000)*. 577-583

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

Hasegawa et al. [2004] T. Hasegawa, S. Sekine and R. Grishman. (2004). "Discovering Relations among Named Entities from Large Corpora". *Proceedings of the 42nd Annual meeting of the Association for Computational Linguistics (ACL-2004),* 415-422

Hobbs, [2003] J.R. Hobbs. (2002) "Information extraction from biomedical text". *Journal of Biomedical Informatics,* 35-4: 260-264

Huang et al. [2004] M. Huang, X. Zhu, Y. Hao, D.G. Payan, K. Qu and M. Li, (2004). "Discovering Patterns to Extract Protein-Protein Interactions from Full Biomedical Texts". *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA,COLING-2004)*. 22-28.

Kambhatla, [2004] N. Kambhatla (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),* 21-26

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

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

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

Miller et al. [2000] S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, R. Weischedel and Annotation Group (2000) "BBN: Description of the

SIFT System as Used for MUC-7". *In Proceedings of 7th Message Understanding Conference.*

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

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

Palakal et al. [2002] M. Palakal, M. Stephens, S. Mukhopadhyay, R. Raje, and S. Rhodes. (2002). "A Multi-level Text Mining Method to Extract Biological Relationships". *Proceedings of IEEE Computer Society Bioinformatics (CSB) Conference*, 97-108

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

Roth and Yih [2002] D. Roth and W.T. Yih. (2002). "Probabilistic reasoning for entity & relation recognition". *The 19th International Conference on Computational Linguistics(COLING-2002)*, 835-841.

Sekimizu et al. [1998] T. Sekimizu, H.S. Park and J. Tsujii, (1998). "Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts". *Proceedings of the Ninth Workshop on Genome Informatics,* 62-71.

Soderland [1999] S. Soderland, (1999) "Learning Information Extraction Rules for Semi-structured and Free Text". *Machine Learning,* 34: 233-272.

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

Stevenson [2004] M. Stevenson. (2004). "An Unsupervised WordNet-based Algorithm for Relation Extraction". *Proceedings of the Fourth International Conference on Language Resources and Evaluation workshop "Beyond Named Entity: Semantic Labelling for NLP tasks"*.

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

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

Wang, [2001] L. Wong. (2001) "PIES, A Protein Interaction Extraction System". *Proceedings of Pacific Symposium on Biocomputing,* 520-530.

Xiao et al. [2005] J. Xiao, J. Su, G. Zhou and C. Tan. (2005). "Protein-Protein Interaction Extraction: A Supervised Learning Approach" the *First International Symposium on Semantic Mining in Biomedicine (SMBM-2005), 51-59.*

Xiao et al. [2003] J. Xiao, T.-S. Chua and J. Liu. (2003). "A Global Rule Induction Approach to Information Extraction". *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence. (ICTAI-03)*, 530-536.

Xiao et al. [2004] J. Xiao, T.-S. Chua and H. Cui. (2004). "Cascading Use of Soft and Hard Matching Pattern Rules for Weakly Supervised Information Extraction". *In Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004).*

Yang et al. [2003] X. Yang, G. Zhou, J. Su and C. Tan. (2003). "Coreference Resolution Using Competition Learning Approach". *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003),* 176-183.

Zelenko et al. [2003] D. Zelenko and C. Aone. (2003) "Kernel Methods for Relation Extraction". *Journal of Machine Learning Research,* 3:1083-1106

Zhou et al. [2000] G. Zhou, J. Su and T. Tey. (2000). "Hybrid Text Chunking". *Proceedings of the forth Conference on Natural Language Learning (CoNLL'2000*), 11-14.

Zhou et al. [2005] G. Zhou, J. Su, J. Zhang and M. Zhang. (2005) "Exploring Various Knowledge in Relation Extraction"*Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005).*

Appendix I:

Current Works in Information Extraction

Information Extraction (IE) is a process that takes unseen texts as input and produces a fixed-format unambiguous data as output. IE systems have been developed for different writing styles like structured text, semi-structured text and free text.

IE can be roughly divided into single-slot IE and multi-slot IE according to the number of templates (or events) extracted from text. Single-slot IE means that at most one template can be found in each document, such as seminar announcements. Multi-slot IE means that zero or more templates can be found in one document, such as management successions.

The importance of IE has been well recognized and there are many different approaches proposed. Most of the system use rule based approach in MUC time. More and more machine learning approaches are explored in recent years. Most of

supervised learning approaches can be roughly divided into two categories: pattern induction approaches, classification based approaches (feature-based and kernel-based). Table 8 provides a summary for the representative work according to IE approaches and testing data format they used.

Besides, statistical model (**Miller et al**, [2000]), bootstrapping, unsupervised learning are also used in some IE systems. To decrease the requirement of corpus annotation, some researchers turned to weakly supervised learning approaches (**Agichtein and Gravano** [2000]; **Stevenson**, [2004]), which rely on a small set of initial seeds instead of a large annotated corpus. However, there is no systematic way in selecting initial seeds and deciding the "optimal" number of them. Alternatively, **Hasegawa et al.** [2004] proposed an unsupervised learning approach to discover relations from a large raw corpus. They assume that the same NE pairs in different sentences hold the same relation type and use the context words in between the same NE pairs in different sentences to form a word vector. Finally NE pairs are grouped according to the cosine similarity between the word vectors. However, this approach only works well on high-frequent NE pairs due to their naïve assumption and the simple word features (**Hasegawa et al.**, [2004]).

In this section, we provide a brief survey on current information extraction approaches from structured text, semi-structure text and free text.

	Single-slot		**Multi-slot**	
	Pattern Induction	Classification based learning	Pattern Induction	Classification based learning
Semi-Structured text	**SRV** [Freitag, 1998]	**HMM** [Freitag and McCallum, 1999]	**WHISK** [Soderland, 1999]	**ME** [Chieu & Ng, 2002]
	WHISK [Soderland, 1999]	**Snow** [Roth & Yih, 2001]		**Zelenko et al.** [2003]
	Rapier [Califf and Mooney, 1999]	**ME** [Chieu and Ng, 2002]		
	BWI [Freitag and Kushmerick, 2000]			
	$(LP)_2$ [Ciravegua, 2001]			
	GRID [Xiao et al. 2003]			
Free text			**WHISK** [Soderland 1999]	**ME** [Chieu & Ng 2002]
			GRID [Xiao et al. 2003]	**Kambhatla** [2004]
			Xiao et al. [2004]	**Zelenko et al.** [2003]
				Culotta & Sorensen [2004]

Table 8. Summary of current IE systems

1.1 Information Extraction from Structured Text and Semi-Structured Text

In structured text, content is organized in a way that readily enables the user to locate, modify, and retrieve any particular component of the text[6]. Examples of structured text include CNN weather task and various tables in relational databases. Most of approaches in structured IE generate rules to specify the order of relevant information and its context (e.g. HTML tags).

In semi-structured text, considerable information is conveyed by the position, layout and format of text. This applies to a lot of web-based information. Examples include job announcements, seminar announcements, and sales catalogs. Semi-structured text is often not full sentences, but rather short phrases. It is often ungrammatical and telegraphic in style, but does not follow any rigid format. So a natural language parser may not be able to well parse semi-structured text. Simple rules that might work for structured text used for rigidly structured text will not be adequate.

Among all IE approaches in structured and semi-structured text, pattern induction approaches are dominant. It is mainly because information in structured text and semi-structured text is in certain format (strictly or loosely). Given descriptions of positive instances and negative instances, pattern induction approaches try to find a concept

[6] www.tufts.edu/vet/internetvet/glossary.html

covering all positive instances and no negative instances. Examples of current pattern induction approaches include **WHISK** [Soderland, 1999], **$(LP)_2$** [Ciravegna, 2001], **GRID** [Xiao et al, 2003] and **Xiao et al.** [2004]. Although these approaches use variant methods to generate rules, all of them have obtained certain success for information extraction from structured text or semi-structured text.

WHISK [Soderland, 1999] generates rules in a top-down manner. It starts from the most general (empty) rule and repeatedly adds optional feature constraints to eliminate negative instances while retaining positive instances. For structured text, such as CNN weather task and BigBook task, WHISK achieved 100% precision and 100% recall. For Semi-structured text, WHISK achieved 92% recall/95% precision in Rental Ads task, 63% recall/77% precision in seminar announcement task and 52% recall/88% precision in Software Jobs task.

$(LP)_2$ [Ciravegna, 2001] uses a bottom-up approach to generate rules. In contrast to top-down approaches, it starts from a most specific rule (complete description of a single positive instance) and repeatedly eliminates feature constraints to cover more positive instances. Another difference between this approa ch and WHISK is that **$(LP)_2$** examines the individual tags rather than full slots. Firstly, **$(LP)_2$** learns a set of tagging rules for each kind of tags. Then additional rules are induced to correct mistakes in tagging rules. **$(LP)_2$** achieved F-measurement of 86 in seminar announcement task and 84.1 in job announcement task.

GRID [Xiao et al, 2003] emphasizes on utilizing global feature distribution in all of the training instances in order to make better decision on rule induction. Each training instance is represented by a context feature vector (global representation). During training, it incorporates the global information in all positive training examples and selects the most prominent generated features to construct the rule in a bottom-up manner. **GRID** achieved F-measurement of 89.3 in seminar announcements task and 80.8 in job announcement task.

Xiao et al. [2004] proposed a bootstrapping approach in which soft and hard matching rules are combined in a cascading manner. This approach started with a small set of hand-tagged instances. At each iteration, soft pattern rules were generated to tag new training instances. Then a set of hard pattern rules were generated by hard pattern rule induction (**GRID**) on the overall tagged data and these hard pattern rules were used to tag the data again. This weakly supervised method was examined on seminar announcement and approached the performance of a fully supervised information extraction system while using only 20% hand-tagged instances.

1.2 Information Extraction from Free Text

In free text information extraction, most of the information is carried in the text itself, although there may be some positional information (e.g., headlines). The text may hold quite complicated sentence structure. Typical applications are extraction from news and extraction from scientific papers and reports.

Free text information extraction has developed under a series of evaluations: first the Message Understanding Conferences [7] (MUC, 1990's) and more recently the Automatic Content Extraction [8] (ACE) evaluations since 1999. For the MUC evaluations, the event to be extracted was fairly specific, e.g., hiring or firing of an executive by a company, satellite launching and terrorism event. In the ACE evaluations, the focus has been shifted to more general relations and events, such as that a person is at a location and a person has some social relation to another person.

Typical examples of free text IE tasks include "Management Successions" task in MUC-6 and "Terrorism Event" task in MUC-4. PPI extraction can be also viewed as an IE task from free text. Information extraction from free text is much harder than structured text because it involves the interpretation of the information conveyed in text -- information which can be described in many natural and different

[7] http://muc.www.saic.com/
[8] http://www.nist.gov/speech/tests/ace/

ways. Therefore, for free text, an IE system needs several NLP tools, such as, syntactic analysis, semantic tagging, recognition for domain objects and etc

In free text IE, both pattern induction approaches (**Soderland**, [1999]; **Xiao et al**., [2003]; **Xiao et al.**, [2004]) and classification based algorithms (feature-based and kernel-based) (**Chieu and Ng**, [2002]; **Kambhatla**, [2004]; **Zelenko** et al., [2003]; and **Culotta and Sorensen**, [2004]) are widely used.

Pattern Induction Approaches:

WHISK [Soderland, 1999] uses pattern induction method to extract information from Management Succession domain. This system extracts information at the sentence level. Each sentence is segmented into subject, verb, prepositional and other phrases by syntactic analyzer. Domain objects in sentences such as person names, company names, and positions are also identified. Their system achieved 46% recall and 69% precision in Management Succession task.

GRID [Xiao et al, 2003] is also applied on free text. Before learning patterns, both training and testing documents are pre-processed by the same NLP modules including sentence splitting, tokenization, morphological analysis, syntactic analysis, PoS tagging, chunking and named entity recognition. **GRID** system achieved an F-measurement of 49 on Terrorism task.

Xiao et al. [2004] applied same cascaded approach by combining soft pattern rule induction and hard pattern rule induction (**GRID**) on MUC-4 free text corpus. The result showed it used 20% hand-tagged instances to approach the performance of a fully supervised information extraction system.

Systems based on pattern induction approaches have a number of desirable properties. Firstly, a rule is relatively easy to understand. Moreover, a rule has a natural and familiar first order version, named Prolog predicates. Since techniques for learning propositional rules can often be extended to the first-order case, they also can be readily used in pattern induction.

The major problem with current pattern induction approaches is that they often scale relatively poorly with the sample size, particularly on noisy data. Another problem in pattern induction learning systems is that it is difficult to select a good seed instance to start the rule induction process. Much research can be done towards this field.

Another commonly used approach in free text IE is machine-learning-based classification approach. There are two classes of classification approaches: one is feature-based classification approaches (**Chieu and Ng** [2002], **Kambhatla** [2004]), and the other is kernel-based classification approaches (**Zelenko et al.** [2003] and **Culotta and Sorensen** [2004]). The approaches used in **Kambhatla**, **Zelenko et al.**

and **Culotta and Sorensen** are introduced in chapter 2. Therefore, we only introduce **Chieu and Ng**'s work.

Feature-based Classification Approaches

The feature-based classification approaches rely on feature-based representation of objects. That is, an object is transformed into a collection of features $f_1, \ldots, fn$, thereby producing an N-dimensional vector for each object.

Chieu and Ng [2002] proposed a maximum entropy feature-based approach to information extraction from free text. Firstly, this approach generates possible candidates that can fill each slot in template. Then another relation classifier is built to classify binary relationship between each pair of candidates. Features used for relation classifier include words between two candidates, candidate types, etc. The last step of their system is to build templates based on the relation information between entities. This approach was examined on management succession task and achieved an F-measurement of 59.2.

The feature-based classification approaches rely on feature-based representation of objects. The most advantage of feature-based classification approach is that it is relatively easy to apply and much faster than kernel-based classification approach. Furthermore, feature-based classification approach is flexible, e.g., feature weights can be learned.

There are two major problems with feature-based classification approaches. 1) In many cases, data cannot be easily expressed via features. For example, in most NLP problems, feature based representations produce inherently local representations of objects, since it is computationally infeasible to generate features involving long-range dependencies. 2) Domain experts' effort is usually required when the approaches are applied on a new domain.

Kernel-based Classification Approaches

In the kernel-based classification approaches, an example is not represented by a feature vector. The kernel-based approaches define a kernel function to compute the similarity between examples. A kernel function is a similarity function satisfying certain properties. More precisely, a kernel function K over the object space X is a binary function K: $X \times X \rightarrow [0, \infty]$ which maps a pair of objects x, y to their similarity score $K(x,y)$.

Kernel-based classification approaches have been successfully applied in many applications such as text categorization and natural language parsing. A unique property of the kernel methods is that we do not need to generate features explicitly. More precisely, an object is no longer a feature vector as it is common in machine learning algorithms. Instead, objects retain their original representations and are used within learning algorithms only via computing a kernel (similarity) function between

them. Such use of objects allows learning approach to implicitly explore a much larger feature space than the one computationally feasible for processing with feature-based classification approaches. **Zelenko et al** [2001] compared the performance of kernel-based classification methods and feature-based classification methods. The results indicate that kernel methods achieve better performance than feature-based algorithms especially in relation extraction tasks.

The major disadvantage of kernel-based classification approach is that it is computationally slow for practical applications. Moreover, a kernel function is required for each kind of relation for a multi-slot information extraction. For example, a unique kernel function is needed for each relation such as *person-organization*, *organization-location*, and etc. Finally, how to find an optimal kernel function is still an unsolved problem.

The advantages and disadvantages of each kind of approaches are summarized in Table 9.

	Pattern Induction	**Feature-based classification**	**Kernel-based classification**
Pros	A rule is easy to understand	Relatively easy to apply	Do not need to generate features explicitly
	A rule has a natural first order version	Faster than kernel-based approach	Objects retain their original representation
		Flexible, feature weights can be learned	Implicitly explore a much larger feature space
Cons	Scale poorly with the sample size	Data cannot be easily expressed via features	Computational slow & need large memory
	Difficult to select a good seed.		A kernel function is required for each relation
			Hard to find an optimal kernel function

Table 9. Comparison of three machine learning approaches
(pattern induction, feature-based classification & kernel-based classification)