RELATION EXTRACTION FROM BIOMEDICAL TEXT

by

Zhongmin Shi

B.Eng., Northwestern Polytechnical University, 1993

M.Sc., Dalhousie University, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Fall 2007

APPROVAL

Name: Zhongmin Shi

Degree: Doctor of Philosophy

Title of thesis: Relation Extraction from Biomedical Text

Examining Committee: Dr. James Delgrande
Professor
Chair

Dr. Fred Popowich
Professor
Senior Supervisor

Dr. Anoop Sarkar
Assistant Professor
Supervisor

Dr. Martin Ester
Associate Professor
SFU Examiner

Dr. Raymond Ng
Professor, Computer Science, UBC
External Examiner

Date Approved: Dec. 3, 2007

Declaration of Partial Copyright Licence

Abstract

In this thesis, we study the extraction of biomedical relations specifically, the extraction of bacterial protein subcellular localizations (BPLs), from abstracts of biomedical scientific articles. A BPL indicates where the protein is located in the bacterium. The extraction of BPLs provides a valuable clue to the biological function of the protein and helps to identify suitable drug, vaccine and diagnostic targets. The work is motivated by our collaboration with researchers in molecular biology, with the goal of automatically extracting BPLs from text to expand their BPL database.

We first introduce a Biomedical Information Retrieval (IR) system, which expands synonyms from a set of biomedical ontology sources and applies a boosting algorithm that captures natural language sub-structures embedded in the text to re-rank retrieved documents. Experiments show that the boosting algorithm works well in cases where the conventional IR system performs poorly.

Our research on the BPL extraction focuses on two learning perspectives: generative and discriminative learning. We propose a three-tier system that integrates a generative model, a discriminative model and a graph-based model to extract BPLs from MEDLINE abstracts. The generative model integrates syntactic features and domain-specific semantic features on the parse tree for a sentence. The model is capable of identifying biomedical named-entities and relations simultaneously from a large set of noisy data and exhibits a significant improvement on the overall performance against a supervised alternative.

We also introduce a discriminative model that applies rich syntactic features from parse trees to extract relations from single sentences. A hybrid pipelined system that integrates generative and discriminative models shows a further improvement against the generative model alone.

Finally we implement a graph model to identify global and hidden relations from multiple sentences and to detect inconsistent predictions.

The study is new to the biomedical natural language processing community in terms of the specific molecular biology task and the capture of the ternary relation among bacterium, protein and location. Our key contributions also lie in learning from noisy data, integrating syntactic and semantic features to extract named-entities and relations simultaneously and establishing an annotated BPL corpus that will benefit relation extraction research.

To my entire family, who always supported me in my academic undertakings.

"Colorless green ideas sleep furiously."

— Avram Noam Chomsky, *1957*

Acknowledgments

This thesis was submitted in December 2007 to Simon Fraser University, Canada, for the degree of PhD in the School of Computing Science.

Many people have helped me in the course of my research and any merit in this thesis is in large measure due to them. First and foremost, I deeply acknowledge my debt to my senior supervisor, Dr. Fred Popowich, who continuously provided invaluable advices and facilities for my research as well as my daily life, and who always generously supported me financially, mentally and spiritually, over these years.

I am greatly thankful for my co-supervisor, Dr. Anoop Sarkar of Simon Fraser University, who led me into the miracle of natural language processing, taught me all I know of computational linguistics and statistical machine learning and pulled me through all of the hard times during the research.

I also express my sincere gratitude to Gabor Melli, Dr. Martin Ester, Dr. Fiona Brinkman and colleagues in the Brinkman Laboratory, Department of Molecular Biology and Biochemistry of Simon Fraser University, for introducing me to the Prokaryotic Protein Localization Project and providing me priceless collaborations on the research. Without them, I would not have completed my thesis in the area of biomedical information extraction.

My thanks are due to Yudong Liu, Baohua Gu, Yang Wendy Wang, Javier Thaine and other fellow colleagues at the Natural Language Laboratory of Simon Fraser University, who collaborated and provided helpful discussions and constructive comments for my research and who made the pleasant and comfortable research ambiance in the laboratory.

My thesis has been funded by the School of Computing Science of Simon Fraser University and the National Sciences and engineering Research Council of Canada.

Their support is thankfully acknowledged.

I am highly grateful to all my friends and their families in my daily life, for their continuous support and care to my family and my studies, and for all the happiness they gave to me.

A special acknowledgment should be given to my parents and parents-in-law, for always being there when I needed them most and for their unselfish support that has accompanied me to come to this point. I also thank my elder sister and brother for their support of my studies over years.

The last and the most heartfelt acknowledgment must go to my wife Yingzi and my son Roy, for their constant understanding, inspiration, encouragement and companionship throughout the period of my studies, which are sources of my happiness, strength, endurance and motivation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction to Biomedical Relation Extraction

With the rapid growth in biological and medical research in the last decade, the amount of biomedical data has dramatically increased and is becoming one of the largest data sources available on the Web for research and public uses. For instance, the Genome database at National Library of Medicine[1] provides genome sequence data of over 1,200 organisms, all of which are either completely sequenced or in the process of sequencing. In addition, MEDLINE at National Library of Medicine provides approximately 13 million references to biomedical articles from 4,800 journals published in more than 70 countries from the year 1950. The volume of MEDLINE grows rapidly, with over 2,000 new articles being added every day[2].

The challenge of finding useful information from the rapidly growing collection of biomedical data lies in developing techniques to aid the understanding of the data. Data Mining is the identification of previously unknown patterns from normally large amount of data and establishment of relationships among data. Data mining techniques are widely applied to many application scenarios, such as customer relationship management and web mining. Data mining is one of the major approaches to identifying gene sequences and determining gene expression levels. Well defined tasks

[1] http://www.ncbi.nlm.nih.gov/sites/entrez?db=Genome

[2] MEDLINE is a bibliographic database of biomedical scientific articles at National Library of Medcine (NLM, http://www.nlm.nih.gov/).

include aligning multiple DNA sequences based on how nucleotides and known genes pair up with one another as well as identifying gene expressions from microarray data and interactions among protein and gene subsequences.

Another direction for information extraction from biomedical data involves natural language processing in two ways. First, language parsing techniques are being applied to gene sequence data to exploit the sequential structure of genes, with the motivation that genes are believed to be the language of life and could thus be understood by using techniques of language processing. Second, natural language processing techniques are being used to understand biomedical scientific articles, for instance, from MEDLINE, and to extract knowledge from them. An example of a MEDLINE record is shown in Figure 1.1. Our research focuses on information extraction from MEDLINE records.

It is extremely difficult for biomedical researchers to build up their own knowledge base from existing publications and update it daily. Since simple indexing and keyword searching cannot satisfy complex searching requirements, automatic methods that can understand human languages and identify interesting information are becoming essential in biomedical information management. The task includes not only identifying individual terms of biomedical substances, such as diseases, drugs, genes and gene productions, but also extracting relevant information of what is expressed or predicted about specific terms, including relations and inferentials [3] among biomedical substances and other hidden information.

Biomedical term identification is a **Named Entity Recognition** (NER) task in the specific domain of biomedical literature. Table 1.1 lists some biomedical terms in the MEDLINE record in Figure 1.1. The task received great attention for years in the research areas of molecular biology, natural language and machine learning. Techniques involved include tagging [90], learning from contextual information [94, 8], statistical thesaurus generation [19], rule-based [27], **Hidden Markov Models** (HMMs) [74], **Support Vector Machines** (SVMs) [48] and **Naive Bayes Classification** [114] for dictionary-based approaches, in which one or more public lexicons or terminology databases are used; decision tree [79], HMMs [18, 51, 100] and SVMs [110, 50]

[3] Inferential: result of reasoning involving inferences from general principles.

Title: *Functional Characterization of the HasA Hemophore and Its Truncated and Chimeric Variants: Determination of a Region Involved in Binding to the Hemophore Receptor*
Abstract: *Hemophores are secreted by several gram-negative bacteria (Serratia marcescens, Pseudomonas aeruginosa, Pseudomonas fluorescens, and Yersinia pestis) and form a family of homologous proteins. Unlike the S. marcescens hemophore (HasA), the P. fluorescens hemophore HasA has an additional region of 12 residues located immediately upstream from the C-terminal secretion signal. We show that HasA undergoes a C-terminal cleavage which removes the last 21 residues when secreted from P. fluorescens and that only the processed form is able to deliver heme to the S. marcescens outer membrane hemophore-specific receptor, HasR. Functional analysis of variants including those with an internal deletion of the extra C-terminal domain show that the secretion signal does not inhibit the biological activity, whereas the 12-amino-acid region located upstream does. This extra domain may inhibit the interaction of the hemophore with HasR. To localize the hemophore regions involved in binding to HasR, chimeric HasA-HasA proteins were tested for biological activity. We show that residues 153 to 180 of HasA are necessary for its interaction with the receptor.*

Figure 1.1: Title and abstract of MEDLINE record PMID: 10913071.

for non-dictionary-based approaches. These techniques will be described in more detail in Chapter 2.

Type of Biomedical Terms	**Mentions**
Organism	S. marcescens, Pesudomonas aeruginosa, P. fluorescens
Protein	HasAPF, HasRSM, HasR
Protein Domain or Region	C-terminal secretion signal
DNA Domain or Region	12-amino-acid region, hemophore regions
Molecular Function	binding, interaction
Location	secreted

Table 1.1: Biomedical terms in the MEDLINE record in Figure 1.1.

Words provide additional information when they are found relevant to each other. Once the biomedical terms are identified, the next step is to determine certain relations in order to answer the crucial questions, such as "how a gene is related to some disease or drug", "how two proteins interact with each other", "what the pathway of a gene product is" or "how to represent unknown relations of biomedical substances in building an ontology". Many biomedical discoveries originate from the identification and characterization of relations among macromolecules. Many interesting interactions are reported in unstructured free text, and thus, unfortunately, are unavailable

for high-throughput analysis. Because of the vast number of molecules and relations, identifying them manually is daunting. Therefore, researchers are investigating the suitability of text processing algorithms to extracting interactions [108].

The **Natural Language Processing** (NLP) community has particular interest in identifying relations from text. The series of Message Understanding Conferences (MUCs) sponsored by DARPA defined various **Information Extraction** (IE) tasks, including the relations among different types of entities and involved the uniform evaluation metrics applied to the IE tasks[4]. The task for MUC-6 involved the filling of a template with extracted information for specific class of events. Multilingual IE tasks were defined in the MUC in 1997.

Automatic Content Extraction (ACE) at the National Institute of Standard and Technology (NIST) is another IE program for the newswire. Since 2003, the program has been providing relation-detection and relation-recognition tasks. These tasks require the detection of certain types of binary relations in the source language data, and the selection of information about these relations. This information is then merged into a unified representation for each detected relation [1].

The shared tasks of the Conference on Computational Linguistic Learning (CoNLL) in 2004 and 2005 focused on **Semantic Role Labeling** (SRL): analyzing propositions expressed by some target verbs in a sentence. In particular, for each target verb, all the constituents in the sentence which fill a semantic role of that verb have to be recognized [13]. Typical semantic arguments include Agent, Patient, Instrument, etc. and also adjuncts such as Locative, Temporal, Manner, Cause, etc. The target verbs exhibit semantic relations among semantic arguments.

In contrast to the news domain, terms in a biomedical text are more difficult to recognize and the relation identification relies largely on domain knowledge. There have also been open evaluations of biological IE tasks on relation identification. The Knowledge Discovery and Data Mining (KDD) Challenge Cup is an open evaluation of data-mining algorithms. The 2002 contest specifically focused on biological text mining. It posed two tasks. The first task dealt with identifying papers containing

[4]http://www-nlpir.nist.gov/related_projects/muc/

experimental evidence for gene expression and identifying the relevant gene products [121]. The second focused on predicting how a gene is related to cellular activities [22].

Another community-based evaluation on biomedical IE is the Critical Assessment of Information Extraction Systems in Biology (BioCreAtIve)[5]. One of its 2004 tasks involved the annotation of human proteins with Gene Ontology (GO) classes, the identifiers for an ontology of gene functions, and documents retrieval that would provide the annotations.

The Text Retrieval Conference (TREC) sponsored by NIST and the U.S department of Defense provides the **Information Retrieval** (IR) community with an infrastructure necessary for large-scale evaluation of IR methodologies [113]. Since 2003, TREC included an information extraction task in its Genomics track. The first year's track consisted of two tasks. The first was an ad hoc document retrieval task given a document collection, topics and relevance judgments. The second dealt with annotations of Gene Reference into Function (GeneRIF). GeneRIF resource was used as both a source of relevance judgments for ad hoc document retrieval and as a target text for information extraction. The track in 2004 also featured two tasks. The first was also an ad hoc retrieval task using topics obtained from real research scientists and a large subset of the MEDLINE database. The second focused on categorizing full-text documents and simulating the task of curators, thus providing structured annotations of gene functions. The Genomics track attracted the most participants in all of TREC 2004.

The biomedical relations identified by most systems can be categorized as follows:

- *unnamed relation*, which provides the associated biomedical terms but does not specify the actual relation.
- *relation class*, which does not specify the relation either but indicates which predefined classes the relation may fall in.

[5]Critical Assessment of Information Extraction Systems in Biology: http://www.pdg.cnb.uam.es/BioLINK/BioCreative.eval.html. 2004.

- *named-relation*, the actual relation among terms.
- *pathway*, the course of the cellular or metabolic process that the biomedical substances may affect.

1.2 Introduction to Bacterial Subcellular Localization

Subcellular Localization (SCL) is one typical biomedical relation. Bacterial SCL states where proteins locate in bacteria. For example:

Example 1.1: E. Coli produces [$_{LOCATION}$ membrane-bound] [$_{PROTEIN}$ lytic transglycosylase] and localizes it in [$_{BACTERIUM}$ murein sacculus].

indicates a *membrane-bound* localization relation between the said bacterium and protein. The bacterial SCL is a key functional characteristic of proteins, since a protein has to be translocated to the correct intra- or extra-cellular compartments or attach to a membrane in order to function properly. This characteristic is essential to the understanding of the functions of different proteins and the discovery of suitable drugs, vaccines and diagnostic targets. Locations of bacterial proteins are listed in Figure 1.2 for Gram+ and Gram- bacteria.

Different from protein-protein, gene-gene and protein-disease relations that are associations of protein molecules from the perspective of biochemistry and signal transduction, the SCLs indicates functions of single proteins and therefore are more fundamental to the study of proteins.

However, experimental determining SCLs is a laborious and time consuming task. Research in computer sciences has been carried out to automatically predict SCLs from protein sequences and the biomedical scientific text. Some of these SCL prediction methods, for instance, Support Vector Machines (SVMs), now exceed the accuracy of some high-throughput laboratory methods for the identification of protein subcellular localization [91]. In later chapters, we will introduce our proposed models that can further improve the performance of SVMs on the SCL prediction from text.

From the natural language processing point of view, finding relations from the biomedical text is more difficult than from widely used domains, for instance, newswire.

The SCL extraction highly relies on protein name identification, which has been recognized as a much harder task than the identification of person, location, organization names and the like. In addition, domain knowledge is required to understand the biomedical text.

This thesis introduces our approach for identifying bacterial SCLs from MEDLINE articles. Specifically, our task is to extract from biomedical articles a relation among: a LOCATION, e.g., *membrane-bound*, a particular BACTERIUM, e.g. *murein sacculus*, and a PROTEIN name, e.g. *lytic transglycosylase*. Therefore, the task is to identify a **BACTERIUM-PROTEIN-LOCATION** (**BPL**) function, a relation among bacterium, protein and location.

Examples of expected system output are shown in Table 1.2. In many circumstances, such relations are not explicitly given and thus this task may require some level of induction from the context. For instance, "binding of [$_{PROTEIN}$Trpl] to the site I" infers a *cytoplasmic* localization from the fact that Site I is the site to which a DNA molecule binds and that the DNA locates at the cytoplasmic layer.

This work is motivated by our collaboration with molecular biologists, who have built an BPL database for bacterial proteins. These BPLs are either curated manually by the biologists or predicted by an automatic BPL prediction model from the most recent NCBI Taxonomy dataset[6] of completely sequenced genomes. Our task is however to extract BPLs from MEDLINE articles.

The task is new to BioNLP in terms of the specific biomedical relation being sought. Therefore, we have to build an annotated corpus from scratch and we are unable to use existing BioNLP shared task resources in our experiments.

The BPLs extracted by our proposed approach will be ultimately examined by human experts, populated into the SCL database and then used by the biologists to improve the accuracy of the SCL prediction model. We have worked closely with them to ensure that the output produced by our system is directly useful in expanding their protein localization database.

[6]http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Taxonomy

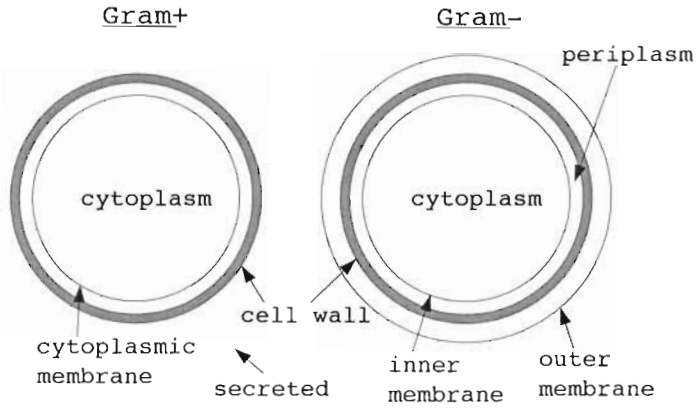

Figure 1.2: Illustration of locations of proteins with respect to the bacterial cell structure.

Organism	Localization	Protein	Relevant Sentence	Pubmed ID
Metha. fervidus	CW	slgA	The genes (slgA)...	1712296
Halo. salinarium	C	Hp71	The samples were...	9396829

Table 1.2: Examples of the BPL output

1.3 Thesis Organization

In this thesis we examine research related to biomedical relation extraction and focus ourselves on two learning directions: generative and discriminative. We propose a generative model that integrates syntactic features and domain-dependent semantic features of a sentence in the parse tree for a sentence, and is capable of identifying biomedical named-entities and relations simultaneously. We also introduce a discriminative model that applies rich syntactic features from parse trees to extract relations from single sentences. In addition, we implement a graph model that finds global and hidden relations from multiple sentences and documents. The overall system is a 3-tier approach that integrates the generative, discriminative and graph models to extract BPL relations from MEDLINE articles.

Research related to biomedical relation extraction is categorized and described in Chapter 2. A biomedical IR system that combines the synonym-based query expansion and boosting-based re-ranking is described in Chapter 3. Chapter 4 introduces details of our relation extraction task, including the process of creating curated data set, details of evaluation metrics and two baseline systems. From Chapter 5 to 7, we describe details of the proposed system, including a generative model, discriminative model, hybrid models and a graph model to address the relation extraction task. Models are evaluated based on our standard test set and compared with baseline systems in each chapter. Finally, in Chapter 8 we summarize our research contributions and draw some conclusions on the techniques being applied and the evaluations being carried out in this research. Appendices contain descriptions of some widely used biomedical data sources and pseudo code of the proposed algorithms.

Chapter 2

Related Work

2.1 Introduction

The Biomedical Natural Language Processing (Bio-NLP) research community has built numerous information extraction systems to identify relations from biomedical articles. Various types of relations have been addressed using NLP, machine learning and other state-of-the-art techniques. We will briefly introduce these relation types below and will describe details of the techniques applied to extract these relations in the rest of this chapter.

The extraction of protein-protein interactions has been widely studied. For instance, [68, 81, 29] were interested in interactions among yeast proteins; [26, 10] focused themselves on interactions between human proteins; systems proposed in [6, 8] experimented with cell cycle control; and [82] attempted to extract interactions among signal compounds, such as Cytokines. The fact is that all proteins in a given cell are connected through an extensive network and the interactions among them affect all processes in a cell.

"The success of genetic epidemiology in identifying polymorphisms associated with common, complex multifactorial diseases largely depends on the detection and characterization of gene-gene and gene-environment interactions" [73]. The gene-gene relation identification tasks have been addressed in [62, 60, 88, 98, 33]. Other relations involved in biological processes of gene regulation include the gene-protein interaction [97, 92] and gene-drug-disease interaction [94, 23, 111, 109, 44, 85, 34, 108].

Other specific relations between molecular entities include cellular localization or structure of proteins that allow proteins to function [23, 46], molecular binding relations - a core phenomenon in molecular biology that provide a strong indication of macromolecular functions [93], molecular pathways involving a series of enzymatic reactions that converts one biological substance to another [33] and relations in general [32, 67].

Various techniques being used for the task of relation extraction from biomedical text. NLP techniques that are used to analyze syntax and semantics of biomedical text include shallow parsing, full parsing, negation detection and co-reference resolution. Relations can also be extracted by identifying co-occurrence of biomedical names. Identification of predicates and their arguments in a sentence also helps finding verbal relations. Rules that formularize patterns of relations can be built manually, semi-automatically and fully-automatically. Statistical machine learning techniques have also been widely applied to the task. These techniques include Naive Bayes classification, Bayesian rules, Bayesian Networks, Support Vector Machines (SVMs), association rules, clustering and kernel methods. Graph-based approaches are capable of visualizing the relations and identifying unapparent relations and pathways. Details of these techniques are described in Section 2.2. Section 2.3 introduces techniques applied to the subcellular localization extraction, the main focus of in this work. Finally, results of some systems being introduced are listed and discussed in Section 2.4.

2.2 Techniques

Since biomedical relations happen among molecular substances, they may be identified from text following certain rules, which would reveal the existence of the relations, e.g., term co-occurrence. Statistical and other machine learning methods that can infer such rules have thus been widely applied to find biomedical relations. Rules can be built from features of the surface string patterns, e.g., words and their frequencies, and/or the syntactic and semantic information by NLP techniques. In this section we categorize the techniques used in biomedical relation identification.

2.2.1 Syntactic and semantic analysis

Parsing techniques are used to analyze the sentence and determine its structure. Such syntactic information has also been widely applied to identification of biomedical terms and relations. However, building the complete parse tree requires polynomial time and gives relatively low efficiency and accuracy. In contrast to the full parser, a shallow parser provides partial parsing information. It is usually used to determine Part-Of-Speech (POS) tags and to find phrases and relations between phrases. For instance, the Charniak's parser is applied to the GENIA corpus with 96.4% POS tagging accuracy and a lower full parsing accuracy (82.9%) [57].

Shallow parsing

Shallow parsing is a natural language processing technique to identify constituents, e.g., nouns, verbs, adjectives, adverbs and the like, of a sentence. It may also provide some understanding of the structure of the sentence, without specifying the full structure in a parsed tree form.

Sekimizu et al. adopt a shallow parser, EngCG [117], in place of the traditional full parser [97]. EngCG is a constraint grammar parser that assigns morphological, syntactic and boundary tags for each word in the corpus, based on syntactic rules, syntactic and heuristic constraints. Figure 2.1 shows a result of shallow parsing by EngCG on a sentence that contains a protein-protein interaction. The strong point about EngCG is its ability to "guess" some morphological or even syntactic tags of unknown words, which is especially useful for tackling biomedical domain texts [97]. Biomedical literature normally contains more unknown words than, for instance, newswire. Lease et al. reported that unknown word rates in Wall Street Journal and GENIA are 2.7% and 25.5% respectively [57].

EDGAR [94] matches **Noun Phrases** (NPs) with concepts in a controlled vocabulary. It attempts to determine the relations among biomedical terms with respect to the interaction of gene expression and drug sensitivity in particular cell types. It begins with assigning a partial syntactic parse to each sentence in the given abstracts with a stochastic tagger [25] that resolves part-of-speech ambiguities in support of a shallow parser [4]. Each NP is examined in the parse for each sentence. It is then

```
"<*i*l-4>"      "*i*l-4" <*> <?> N NOM SG @SUBJ
"<was>"         "be" <SV> <SVC/N> <SVC/A> V PAST SG1,3 VFIN  @+FMAINV
"<the>"         "the" <Def> DET CENTRAL ART SG/PL @DN>
"<only>"        "only" A ABS  @AN>
"<cytokine>"    "cytokine" N NOM SG  @PCOMPL-S
"<that>"        "that" <NonMod> <**CLB> <Rel> PRON SG/PL  @SUBJ
"<binds>"       "bind" <SVO> <SV> <P/with> V PRES SG3 VFIN @+FMAINV
"<to>"          "to" PREP  @ADVL
"<a>"           "a" <Indef> DET CENTRAL ART SG @DN>
"<hemopoietin>" "hemopoietin" <?> N NOM SG @NN>
"<receptor>"    "receptor" N NOM SG  @<P
"<and>"         "and" CC @CC
"<that>"        "that" <NonMod> <**CLB> <Rel> PRON SG/PL  @SUBJ
                "that" PRON DEM SG  @SUBJ
"<did>"         "do" <SVO> <SVOO> <SV> V PAST VFIN  @+FAUXV
"<not>"         "not" NEG-PART @NEG
"<activate>"    "activate" <SVO> <DER:ate> V INF  @-FMAINV
"<p21ras>"      "p21ras" <?> <NoBaseformNormalisation> N NOM SG/PL @OBJ
```

Figure 2.1: An example of shallow parsing result by EngCG [97].

determined whether the NP matches a UMLS Metathesaurus concept[1].

Genescene [59, 61, 62, 60] identifies structures of relations with the help of a shallow parser. It introduces the Arizona relation parser [69] that automatically extracts precise and semantically rich relations between pairs of NPs in MEDLINE abstracts. The relation can contain five elements. A **Left-hand side NP** is connected to a **Right-hand side NP** by a **Connector**, which is usually a verb. A **Modifier**, usually anf adverb, can modify the connector. A **Negation** can modify the entire relation. For example:

Example 2.1: Thus [$_{LHS}$ Hsp90] does [$_{negation}$ not] [$_{connector}$ inhibit] [$_{RHS}$ receptor function] solely by steric interference.

To extract structure information from text, the relation parser distinguishes between two sets of words: open- and closed-class words. Open-class words are the category of

[1]The UMLS Metathesaurus is a biomedical and health-related vocabulary database in the National Library of Medicine: http://www.nlm.nih.gov/research/umls/.

words that continuously grows because new words are added. These are nouns, NPs, and verbs that provide the semantic content of text. Closed-class words are those categories whose membership does not change, such as prepositions, conjunctions, and negations. The relation parser uses closed class words to establish the structure for a sentence. These structures become templates for relations [60]. Templates will be introduced in Section 2.2.4.

Full parsing

In contrast to the shallow parsing, the full parsing is a natural language processing technique to specify grammatical structure of a sentence, in a form of the parsed tree that contains all syntactic dependence information among constituents of the sentence.

Yakushiji et al. used a full parser with a large-scale, general-purpose grammar to extract predicate-argument structures from text [119]. The parser converts a variety of sentences in the text, which describe the same event, into an argument structure. In the argument structure, the verb is represented by the event and the subject and object are represented by arguments. Figure 2.2 shows an example of the argument structure, which basically consists of **Relation** and **Argument** elements. The event templates are then mapped by the predicate-argument structures as described in Section 2.2.4. Two preprocessors are used to reduce ambiguity in the syntactic parsing stage. One preprocessor identifies and semantically classifies NPs that are technical terms; these are treated as atomic units in the parsing stage. The second preprocessor uses local constraints instead of part-of-speech tagging to reduce lexical ambiguity [119].

In contrast to [119], GENIES extracts and constructs information about cellular pathways from full articles based on manually-built grammar rules [33]. It attempts to obtain a complete parse according to grammar rules, which consist of semantic patterns interleaved with syntactic and semantic constraints to identify relevant relations. The grammar was defined manually by observing typical syntactic and semantic co-occurrence patterns in the sample text. If full parsing fails, it uses alternative strategies, such as segmenting and shallow parsing. GENIES categorizes verbs into semantic classes as listed in Figure 2.3. The following is the parse tree produced by

<PROTEIN_10> incorporation into cells was also observed when the cells were incubated with <PROTEIN_2> or with <PROTEIN_3>.

$$\begin{bmatrix} \text{REL} & : \textit{"observe"} \\ \text{ARGS} & : \begin{bmatrix} \text{Comps} : \textit{"<PROTEIN\_10> incorporation into cells"} \end{bmatrix} \\ \text{ADJ} & : \textit{"also"}, \textit{"when the cells were incubated"}, \\ & \quad \textit{"with <PROTEIN\_2> or with <PROTEIN\_3>"} \end{bmatrix}$$

$$\begin{bmatrix} \text{REL} & : \textit{"incubate"} \\ \text{ARGS} & : \begin{bmatrix} \text{Comps} : \textit{"the cells"} \end{bmatrix} \end{bmatrix}$$

Figure 2.2: An example of argument structure of a sentence [119].

their full parser on the sentence, *phosphrylated Cbl coprecipitated with CrkL, which was constitutively associated with the C3G*:

$[action, attach, [protein, Cbl, [state, phosphorylated]],$

$[protein, CrkL, [action, attach, [protien, CrkL], [protein, C3G]]]]$

where Cbl, CrkL and C3G are protein names, *coprecipitate* and *associate* belong to the *attach* semantic class, as listed in Figure 2.3. GENIES was integrated into the GeneWays System [52, 95].

MedScan also extracts relations based on semantic structures built on the top of syntactic trees. Unlike [119], MedScan applies a more compact and efficient representation of rules and produces more formal semantic structures for sentences. It has been used to extract relations between human proteins by efficiently processing sentences from MEDLINE abstracts and producing a set of semantic structures representing the meaning of each sentence [80, 26]. Sentences are processed by a syntactic parser based on active chart parser algorithm [3] in combination with a bottom-up parsing approach. The parser constructs a set of alternative syntactic structures using a set of grammar rules in a form of augmented transition networks (ATNs), which are formally equivalent but are a significantly more compact and efficient representation of rewrite rules, where common parts of rules are packed into a single network path and are traversed only once during parsing [80]. Once the syntactic structures of an input sentence are produced, a semantic processor transforms each of them into a normalized semantic tree, representing logical relations between words in a sentence.

Class	Actions and Processes
activate	hasten, incite, up-regulate
attach	bind, form complex, add
breakbond	sever, cleave, dephosphorylate
cause	based on, due to, result in
contain	contain, container
createbond	methylate, phosphorylate
generate	express, produce, overexpress
inactivate	repress, suppress, down-regulate
modify	mutate, modify
process	myogenesis, apoptosis, cell cycle
react	interact, react
release	disassemble, discharge
signal	regulate
substitute	replace, substitute

Figure 2.3: Semantic classes associated with actions, processes and other relations [33].

```
inhibition
{
     agent: {11768=TGF-beta 2}
     patient: activity
                { patient: {11892=tumor necrosis factor} }
     attribute: [type: ''in''; value: resting lymphocytes]
     attribute: [type: ''after''; value: Treatment
                                         {agent:
                                          subject:
                                          substance: nitric oxide
                                         }]
```

Figure 2.4: An example of semantic structure by [80].

Semantics in MedScan represent the meaning of a sentence as a tree of categorized predicate-argument relations between lexemes. A semantic tree is built from elementary semantic nodes, each representing a particular sentence lexeme. Semantic nodes reference other argument semantic nodes through two types of slots. **Role Slots** represent the most important lexeme relations, such as subjects or objects of actions. **Attribute Slots** represent auxiliary lexeme relations such as time, place or mode of action [80]. Figure 2.4 shows an example of the semantic structure built on the top of a syntactic tree, which contains role slots and attribute slots identified from the sentence.

Coreference resolution

Coreference is essential to establish the connections and identify biomedical relations among sentences and is regarded as one of the most difficult problems in NLP. The coreferential terms normally include pronouns[2], definites[3] and indefinites[4].

Sekimizu et al. described their preliminary coreference resolution module that deals with impersonal pronouns[5] only in [97]. The algorithm is similar to that in [56].

[2]Pronoun is a function word or expression that replaces a noun or an NP.

[3]Definite is the NP distinguishing between entities which are specific and identifiable in a given context and indefinites, for example, *the cat*.

[4]Indefinite is the word that replaces a noun without specifying which noun it replaces, such as *one*, *another*, *each*.

[5]In contrast to personal pronoun, an impersonal pronoun does not refer to a particular person.

It takes morphological and syntactic features of all terms from the previous stage and resolves each impersonal pronoun as follows:

- Collect the potential antecedent term(s) in the same sentence.
- Filter each pronoun-antecedent pair using number, sortal and modifier consistency constraints.
- Order pairs by dynamic syntactic preference, such as recency and salience.

Consider Example 2.2.

Example 2.2: ... when cellular sterol levels are low, the SREBPs are released from the endoplasmic reticulum membrane, allowing them to translocate to the nucleus and activate SREBP target genes.

The sentence contains an *activate* relation between *them* and *SREBP target genes*. Using the coreference resolution technique introduced above, they were able to connect *SREBPs* and *them* and predict such a relation between *SREBPs* and *SREBP target genes*. A similar technique was used in Medstract [89, 90] to resolve biologically relevant sortal terms (i.e., proteins, genes, and bio-processes) and pronominal anaphora including third person pronouns and reflexive pronouns. A reflexive pronoun is a pronoun with a reflexive relationship with its self-identical antecedent, such as myself, yourselves.

In addition to patterns and preference rules applied in [97], Thomas et al. used statistical methods to recognize phrases referring to entities and events of interest [112]. Hahn and Romacker tracked coreference relations by center lists in the MedSynDiKaTe system [38]. Similar to the method of [97] that orders a list of pronoun-antecedent pairs, the center list provides a list of antecedents of an anaphoric expression in the subsequent utterance, with the decreasing order of the preference for establishing referential links based on the centering model [37, 107]. An utterance is the basic unit of text. It could be a sentence, a clause or a phrase. The centering model describes relations among local coherence, the use of referring expressions and the track of focus attention within a discourse segment.

Negation Detection

Although negative expressions do not occur frequently in biomedical articles, negation detection is necessary since a detection failure may result in an opposite prediction. For instance, an *inhibit* relation between *Hsp90* and *receptor function* would be mistakenly predicted if the negation is ignored in the following sentence:

Example 2.3: Thus [$_{LHS}$ Hsp90] does [$_{negation}$ not] [$_{connector}$ inhibit] [$_{RHS}$ receptor function] solely by steric interference.

Blaschke and Valencia employed a rule-based method in the relation identification (in Section 2.2.4) and included negative rules to reduce the number of false positive identifications. Negations are given an associated score of zero to prevent them from contributing to the establishment of associations between the corresponding names [8].

Summary

Full parsing provides more complete information about a sentence than shallow parsing, however, it requires more computation. Shallow parsing could be flexible enough for specific subtasks, such as document retrieval, information extraction and question answering, and it could be more reliable in handling ill-formed sentences, such as conversation. Coreference resolution is essential to associating terms and identifying relations among terms. Negative expressions were ignored in most of the systems, since they do not frequently occur in sentences.

Now that we have examined some fundamental syntactic processing techniques, let us move on to some semantics-related and statistics-intensive approaches.

2.2.2 Term co-occurrence

The assumption of the co-occurrence approach is that, if two genes have a related biological function, it is likely that these two gene names (or aliases of those genes) co-occur within the biomedical literature [104]. The relations between co-occurring terms can be identified statistically or by rules.

BIOBIBLOMETRICS [104] retrieves biomedical information using the Saccharomyces cerevisiae Genome Database (SGD) and a set of MEDLINE abstracts published between 1997 and 1998 containing the term 'Saccharomyces cerevisiae', with two or more gene names co-occurring in each abstract. From this co-occurrence data a matrix that contains dissimilarity measurements of every pair of genes is constructed, based on their joint and individual occurrence statistics as below.

$$b_{ij} = \frac{|S_i| + |S_j|}{|S_i \bigcup S_j|} \tag{2.1}$$

where S_i and S_j are sets of all documents containing gene i and j respectively. b_{ij} is the dissimilarity metric and used to identify syntactic relations.

An analysis and knowledge discovery method, introduced in [106], aims to identify related genes as well as their shared functionality (if any) based on a collection of retrieved relevant MEDLINE abstracts in a vector space model. The weight of the k^{th} term in the abstract a_i is calculated as:

$$W_i[k] = T_i[k] * log(N/n[k]) \tag{2.2}$$

where $T_i[k]$ is the frequency of the k^{th} gene term in the abstract a_i, N is the total number of abstracts in the collection, and $n[k]$ is the number of abstracts containing the kth gene term [106]. The relation of two genes is measured as the sum of the product of gene weights over all abstracts. To find out what the relation is, the following method is applied: if a word in a sentence that contains co-occurrences of genes matches a relation in the thesaurus, the word is given a score of 1. The highest score over all sentences for a given relation is then taken to be the score of the relation between two genes or proteins [106].

Concept Space, implemented in GeneScene system [60], is a bottom-up technique that captures relations between pairs of noun phrases from MEDLINE abstracts. Three external knowledge sources are used to tag NPs: GO[6], HUGO nomenclature[7]

[6]Gene Ontology (GO, http://www.geneontology.org/) provides a common language to describe aspects of a gene products biology.

[7]HUGO nomenclature provides names to human genes: http://www.genenames.org/.

and UMLS. Each identified NPs is assigned a weight by an equation similar to 2.2. Concept Space finally produces a list of relations that consist of two, ordered, relevant, medical NPs. Each relation consisting of two NPs, T_k and T_j, is given a weight $Weight(T_k, T_j)$ indicating its importance [60] by the following asymmetric cluster function [16]:

$$Weight(T_k, T_j) = \frac{\sum_{i=1}^{n} d_{ij} \times d_{ik}}{\sum_{i=1}^{n} d_{ij}} \tag{2.3}$$

where i denotes the ith of n documents and d_{ij} indicates the number of occurrences of T_j in the ith document. The function is asymmetric due to the fact that $Weight(T_k, T_j) \neq Weight(T_j, T_k)$.

Ding et al. investigate in [28] the task of mining relations among biochemical terms, based on term co-occurrence at different levels: abstracts, adjacent sentence pairs, sentences and phrases, using the standard information retrieval performance measures of recall, precision and effectiveness. The corpus consists of MEDLINE abstracts retrieved from PUBMED using ten queries, each of which is the AND of two biochemical nouns. The text between two successive periods is defined to be a sentence. The text between any two successive punctuation marks is defined as a phrase. Their experiment results show that term co-occurrence at sentence pairs performs poorly in precision and more sophisticated text processing techniques than statistical term co-occurrence (e.g., rule-based extraction or some machine learning approaches) can increase precision.

XplorMed [84, 86] is intended to extract dependency relations between the words of the abstracts in MEDLINE. The system starts with a query in MEDLINE and calculates relations between words present in the same abstract described using probabilistic binary relations. The degree of **relatedness** between the words is defined as the reciprocal of dissimilarity measurement 2.1. The degree of **inclusion** of word i into word j is defined as:

$$Inclusion(i, j) = \frac{|S_{ij}|}{|S_j|} \tag{2.4}$$

where S_i is set of all documents containing the word i and S_{ij} is the set of all documents

containing both i and j. Important words can be identified by their high association score, which is the sum of the inclusion degree of all other words into that word:

$$Association\_Score(i) = \sum_k Inclusion(k, i) \tag{2.5}$$

The system selects words with high association score and predicts relations between them [84].

G2D is a database of candidate genes for mapped inherited human diseases [85]. A data mining algorithm extracts associations of genes and diseases by searching among MEDLINE, RefSeq as well as GO and matching RefSeq sequences of genes with chromosomal mapping information of diseases. The algorithm consists of three major steps:

1. The associations between pathological conditions (MeSH C) and chemical terms (MeSH D) are computed by their co-occurrences in MEDLINE abstracts, while the relations between chemical terms and terms describing protein function are calculated using the RefSeq database, which contains more than $10,000$ genes whose function is annotated with terms from GO;

2. The algorithm combines the associations of functional terms to chemical terms with the previously established associations of pathological conditions to chemical terms, to derive the aforementioned relations between pathological conditions and protein-function terms;

3. The gene candidates for a given mapped disease are then sorted by carrying out a sequence comparison between the respective chromosomal region and the set of scored RefSeq sequences.

The gene relations can be translated from content-based relations among MEDLINE abstracts, as introduced in [99, 98]. According to their method, each gene is mapped to a single document, roughly discussing the gene's biological function. The document is the representative of the gene and is called the kernel of the gene. The literature database is then searched for documents similar to the gene's document.

Thus, for each gene, a set of similar documents related to its functional role is produced. Since each set corresponds to a gene, the similar document sets can be mapped back to their corresponding genes, and functional relations among these genes can be established.

ArrowSmith, BITOLA and G2D suggest the linkage between a given disease and certain genes by tracing chains of terms. However, since they accept only one or two intermediate terms, the variance of meaning between terms could be too large to enable a precise relation of a disease to genes. Takahata and Kouchi attempted to minimize such variance by proposing a hypothesis on the relation between a disease and a gene, with intermediate terms falling into physiology, biological phenomena or biochemical material category [109]. The co-occurrence relation between any terms in MEDLINE abstracts is calculated by the reciprocal of Equation 2.1.

Summary

The co-occurrence of terms can be found within a sentence, among sentences and even within articles. Most term co-occurrence systems only find unnamed relations as described above. Many systems prefer statistical approaches by applying some dissimilarity measurement between two biomedical terms that form a relation.

2.2.3 Predicate-argument structure

In the scope of natural languages, **Arguments** are logical subject and object and a **Predicate** can refer to a verb, noun or pronoun that "connects" subject and object . Therefore, predicate-argument structure indicates how subconstituents are semantically related to their predicates. The idea behind extracting relations by predicate-argument structure is that sentences usually contain a significant number of terms connected by verbs that indicate the type of relations between them.

Early work towards biomedical relation identification was conducted by [97] by Sekimizu et al [89, 7]. They describe an automatic way of extracting the relations between the proteins and gene products expressed by a set of frequently-seen verbs from MEDLINE abstracts. Their system first determines all NPs and introduces a heuristic algorithm [97] to find the arguments (subject and object) of frequently-seen verbs. A protein name identification system [35] is then used to determine whether

the arguments are protein names.

Instead of using frequently seen verbs used in [97], the SUISEKI project [6] manually chooses a set of pre-specified words indicating relations related to protein interactions. Then a series of simple rules that interpret the construction "protein - relation - protein" is applied to recognize one relation in each sentence.

Proux et al. [88] use a combination of existing linguistic and knowledge processing tools to extract gene interactions from Flybase, the database on Drosophila Melanogaster. These sentences contain two gene names and have been checked by experts to determine whether they contain gene interactions. A shallow parser is used to extract basic syntactic relations such as subject-verb or verb-object. The relation is then built into a conceptual graph, which will be introduced in Section 2.2.6.

ClearResearch [32] implements a generic template, VerbalRelation, which is a set of rule-based patterns consisting of shallow syntactic information (i.e., POS tags, phrases) and named-entity information (i.e., genes, proteins, diseases, etc.) at a full sentence or phrase level. The template is used to match relations by extracting two NPs connected by a verb. The extracted NPs are then classified according to the pre-defined categories (e.g., genes and diseases) to which their terms belong.

Methods that extract relations using predicate-argument structure as introduced above are actually creating rules, which indicate verbal relation and biomedical substances associated with the relation. In fact, rule-based approaches are closely related to the syntactic processing and term co-occurrence that we described in previous sections. Next, we will discuss various methods from the rule-based point of view.

2.2.4 Rule-based extraction

The predicate-argument structure actually acts as a typical pattern to match the phrases indicating relations. There could be other identifiable patterns to which the text conforms. This can be used to define a **Template** or **Frame** - a table with slots that can be instantiated with the bits of information extracted from a given article. The aim of an information extraction system is then to compose a set of pattern matching rules for assigning entities and events in the slots of such templates. This approach is called **Rule-Based**, **Template-Based** or **Frame-Based** extraction.

Rules can be generated manually, automatically learned from hand-made rules or generated fully automatically.

Manually-built rules

Blaschke et al. in [8] show an extension of their work on predicate-argument structure as described in [6] and introduce an object-oriented rule-based approach. The templates are defined manually by filtering large amounts of text to find the most frequent constructions that implicate two protein names and express a direct or indirect relation. Typical descriptions of protein relations include "*protein A is a*" and "*protein B is a new member of a family*". Each template is assigned a probability score depending on its reliability and also accounts for negation and distance between protein names and relation term (the larger the distance, the lower the score). The score of a relation is the sum of the scores of all templates it matches.

A system for the extraction of protein-protein interactions from MEDLINE abstracts is given in [81]. It used only surface forms of word patterns that were presented by the word positions. Several keywords that are frequently encountered and are related to protein interaction are collected from the abstracts. The system thus searches for particular patterns including the keywords. The patterns also represent positions between the keywords, protein names, and other characteristic words, such as prepositions in the sentences.

Highlight is a general purpose IE system and is applied to the relation extraction from MEDLINE abstracts [112]. Instead of using surface pattern of words as described in [81], It applies POS taggers and partial parsers for certain syntactic structures, such as NPs, and performs discourse analysis to identify co-referring NPs. The domain specific patterns are written to map relevant information to templates that contain slots for specific information. The system captures only the subset of protein interactions associated with the verb phrase, such as *interact with*, *associate with*, and *bind to*. A given template is ranked according to a measure of confidence that it is filled correctly, depending on factors such as certainty that each NPs is a protein, number of times the relation occurs and modality associated with the relation.

EDGAR [94] extract information about drugs and genes relevant to cancers from

MEDLINE citations, as well as abstracts. They argue that many abstracts show the characteristic that relations are usually described in a single sentence containing a gene name, a drug name and a cell name. A set of simple hand-made rules corresponding to this characteristic is thus made to assert the interactions of drugs, genes and cells. However, the relevant relations are not always expressed this straightforwardly and some phenomena - such as coordination, anaphora and underspecified reference - can complicate the task. For instance, to process the sentence:

Example 2.4: The overexpression of [$_{gene}$ catalase] or [$_{gene}$ Cu,Zn-superoxide dismutase] ([$_{gene}$ Cu,Zn-SOD]) did not affect the sensitivity of [$_{cell}$ HeLa] cells to [$_{drug}$ cis-platinum].

their system would identify (*catalase*, *HeLa*, *cis-platinum*) and (*Cu,Zn-superoxide dismutase*, *HeLa*, *cis-platinum*), but would not be capable of handling the *or* construct.

ARBITER is a Prolog program that extracts assertions about macromolecular binding relations asserted in MEDLINE abstracts [93]. Only those binding terms that are asserted in the text as participating in a particular binding predication are extracted, and a set of pre-defined morphological and semantic rules is specified on NPs to determine binding arguments. A partial analysis of negation and coordination is undertaken by ARBITER, but anaphora resolution and a syntactic treatment of relativization are not attempted [93]. So it can handle the above example with the *or* logic, but would fail to identify the specific referents of *these drugs* in the sentence below.

Example 2.5: By contrast, activated H-ras, which acts downstream of src, failed to induce resistance to either of these drugs.

BioNLP is a component of the system PIES [118] and is focusing on the discovery of specific protein-protein relations from MEDLINE abstracts and the automatic construction of underlying pathway maps[8] [78]. BioNLP maintains a set of function words for each supported relation type. For example, function words of the *inhibit-activate* relation are:

[8] A pathway map is a directed graph in which biomedical substances and events are interconnected with each other based on interactions among them.

inhibitor: {inhibit, suppress, negatively regulate}
activator: {activate, transactivate, induce, upregulate, positively regulate}

Noun phrases in classical dictionaries are excluded from the set of function words, while those in protein dictionaries are included. BioNLP then seeks out sentences containing any of the function words for specific relation type and then searches for any protein names, which are then associated with the function words using hand-made pattern matching rules.

BioRAT extracts biomedical information from the full papers of MEDLINE [19, 20]. It defines the relations by template, which consists of a hand-made object-oriented-based set of rules. It also includes a template design tool with a graphical user interface, which allows non-expert users to develop templates without having to learn a complex new language [20]. BioRAT is one of the Bio-IE systems that use the full papers instead of just abstracts. Their experiments confirm that the density of 'interesting' facts found in the abstract is much higher than the corresponding density in the full text. They also suggest the study of information density, which focuses on finding the location of each fact extracted from the set of full-length papers.

Semi-Automatically-built rules

The rule-based systems introduced above match sentences by hand-made patterns on some pre-defined set of syntactic structures representing certain types of relation. However, a relation can be represented in various forms in natural language text and the workload of preparing patterns manually would be significantly expensive if the text involves a wide scope of events. For example, Yakushiji et al. attempt to minimize manual pattern construction, by converting the surface form of sentences to predicate-argument structures, using a general-purpose, domain-independent parser [119]. The predicate-argument structures are then converted to template representations by domain-specific mapping rules. Although the mapping rules are still hand-made, the variation is decreased significantly by converting the surface form of sentences to argument structures and thus requires much less human effort.

Plake et al. proposed a method for automated extraction of protein-protein interactions from scientific text [87]. The system matches sentences against syntactic

patterns typically describing protein interactions. They define a set of 22 patterns, each a regular expression consisting of anchor positions and parameterizable constraints. This small set is then refined and optimized using a genetic algorithm on a training set. No heuristic definitions are necessary, and the final pattern set can be generated completely without manual curation.

Automatically-built rules

GeneScene [59, 61, 62] extracts relations by filling preposition-based templates, specifically the action, theme and agent slots. Their system first identifies basic templates, based on English closed word classes, such as prepositions and conjunctions. This can be accomplished by retrieving the main verb close to the preposition to fill the action slots, and searching for NPs to the left and right of the verb and preposition to fill the theme and agent slots. NP detection is based on a variant of stop word phrasing: punctuation, auxiliaries, verbs, and closed-class words that are used as indicators of the start and end of phrases [59]. Basic templates are then combined and rewritten by rewrite rules into more complex patterns that reflect the underlying sentence logic, which is necessary to correctly represent the information [59].

Three methods of automatically generating rules are introduced in [10] to extract protein interactions from MEDLINE abstracts. They all start with sentences containing interacting proteins and repeatedly generalize these sentences to form rules, which allows adjunctions of words and counts the distances between words. The first method finds **Longest Common Subsequences** (LCS) between original rules and words in LCS composing the new rules. The second method uses **Edit Distance** (ED) and creates more specific rules that contain disjunctive words. The common words between the rules are preserved, the disjunctions of words are made when one is replaced by another in the edit sequence, and words that are added or deleted in the edit sequence are dropped. The third method finds all common sequences between the two rules and considers their conjunction as the generalization. An algorithm similar to beam search is then applied using one of the above generalization methods. It considers only the best rules for generalization at any time.

Summary

Both the term co-occurrence approach and predicate-argument structure can be regarded as special forms of rule-based extraction. Finding co-occurring terms is relatively intuitive for identifying relations between terms, and the predicate-argument structure seems a more precise method by assuming that many relations are represented as co-occurring terms connected by verbs. Therefore, predicate-argument structure systems can identify named relations that associate with verbs. The rule-based extraction systems introduced in this section generate even more complex rules in addition to co-occurrence and NP-verb-NP.

Rules can be created manually, semi-automatically and fully automatically. Generally speaking, manual rules are expensive, especially when the text involves a wide scope of events. Semi-automatic rule-based approaches usually start from a set of manual rules and create new rules by converting, optimizing or learning from existing rules. In contrast to approaches with manual rules, fully automatic rule-based methods directly learn rules from text, thus are highly adaptable and scalable to various information extraction tasks. Next, we will examine systems that apply machine learning techniques to extract relations.

2.2.5 Statistical machine learning approaches

Statistical machine learning has been widely applied to biomedical relation identification tasks. These methods include Naive Bayes classification, Bayesian rules, Bayesian Networks, Support Vector Machines (SVMs), association rules, clustering and kernel methods.

Craven et al. classify sentences into those with relation and without relation [23]. They define a few types of relations, including *subcellular-localization*, *cell-localization*, *tissue-localization*, *associated-diseases* and *drug-interactions*. They attempted to extract a pair of words that could possibly express some type of the relations, when both words occur in the same sentence and when the sentence is classified as a positive instance by the statistical model. The model is learned by a Naive Bayes classifier with a bag-of-words representation. Given a sentence e of n words $(w_1, w_2, ..., w_n)$, Naive Bayes estimates that the sentence belongs to each possible class $c_j \in C$ corresponding

to a relation from the training data as following.

$$Pr(c_j|e) = \frac{Pr(c_j)Pr(e|c_j)}{Pr(e)} \approx \frac{Pr(c_j)\prod_{i=1}^{n} Pr(w_i|c_i)}{Pr(e)} \quad (2.6)$$

Marcotte et al. show that the frequencies of discriminating words in MEDLINE abstracts scored by a Bayesian approach can be used to determine whether or not a given paper discusses protein-protein interactions [68]. Discriminating words that appear at unexpectedly high or low frequencies in abstracts discussing the interactions are identified from the training set of abstracts. They indicate words that would be useful for discriminating the training abstracts from other abstracts. Using a Bayesian approach, each of the many MEDLINE abstracts can then be scored for its probability of discussing the interactions according to the frequencies of the discriminating words observed in the abstract.

Hristovski et al. use association rules between pairs of medical concepts as a method to determine which concepts are related to a given starting concept [45, 44]. They first calculate all the associations between the major MeSH terms, and then limit the amount of associations, by only taking the associations between major MeSH headings and with high support and confidence measures [45]. This method, however, produces a large number of candidate concept relations that have to be evaluated. For example, their system proposed 15,617 potential discoveries from 2582 documents in which *Multiple sclerosis* occurs. To decrease the number of candidate relations and to make the system more suitable for disease candidate gene discovery, they included background knowledge about the chromosomal location of the starting genetic disease as well as the chromosomal location of the candidate genes when such knowledge is available [44]. The background knowledge consists of genes and chromosomal location information extracted LocusLink[9], OMIM[10] and HUGO. As a result, they are able to limit the candidate genes to those that fall into the same chromosomal location as

[9]LocusLink (http://www.ncbi.nlm.nih.gov/projects/LocusLink/) provides a single query interface to curated sequence and descriptive information about genetic loci. It is superseded by Entrez Gene (http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene).

[10]Online Mendelian Inheritance in Man (OMIM, http://www.ncbi.nlm.nih.gov/sites/entrez?db=OMIM, is a database of human genes and genetic disorders at National Library of Medicine.

the starting disease.

PreBIND is a component of a literature mining system designed to find protein-protein interaction information from MEDLINE abstracts and to present this to curators or public users for review and submission to the BIND database[11] [29]. It uses Support Vector Machines (SVMs) to quickly train a machine learning algorithm to recognize interaction-like articles and bypasses the laborious process of building a domain-specific semantic grammar. PreBIND first maps positive and negative examples to a multi-dimensional vector space and then employs SVMs to discover a boundary that best separates positive from negative examples. So text samples can be classified by this boundary.

VCGS (Vocabulary Cluster Generating System) is designed to automatically extract and determine associations among cancers from MEDLINE abstracts [34]. Firstly, each document is mapped to a vector by Latent Semantic Analysis (LSA) [55]. Each element in a vector is a $tf \cdot idf$ weight of the corresponding term, where $tf \cdot idf$ is the product of term frequency and inverse document frequency. Next, term vectors are generated by considering all the weights corresponding to a term as represented in all the documents. Terms are then associated with each other by the clustering method based on their distance in the vector space. A similar method is employed in [42]. They utilize LSA to identify conceptually related genes based on titles and abstracts in MEDLINE. Related genes are identified by rank order or by hierarchical clustering using the gene distance matrices.

Chang uses the co-occurrence approach to identify the related genes and drugs from MEDLINE abstracts and classifies them into five pre-defined categories based upon Pharmacogenomics Knowledge Base (PharmGKB[12]), which collect information about related genes and drugs using a community-based online submission tool [108].

[11]The Biomolecular Interaction Network Database (BIND, http://bind.ca), is a database of curated and achieved biomolecular interaction and pathway data.

[12]PharmGKB, the Paramacogenetics and Pharmacogenomics Knowledge Base, http://www.pharmgkb.org/

These categories are *Clinical Outcome*, *Pharmacodynamics and Drug Response*, *Pharmacokinetics*, *Molecular and Cellular Functional Assays* and *Genotype*. In the experiments of co-occurrence, he counts the number of abstracts and sentences where both the gene and drug occurred and finds that the absence of a thesaurus of gene and drug synonyms results in nearly a quarter of relations being missed. Each document is then mapped to a vector of words. Finally, a Maximum Entropy classifier is applied to assign each gene-drug pair one of the five categories.

Hasegawa et al. discover relations among terms based upon clustering on context vectors of term pairs [39]. The term pair is defined as two terms co-occurring within the same sentence and being separated by at most N intervening words. A context vector consists of the bag of words formed from all intervening words from each term pair. Each word of a context vector is weighted by $tf \cdot idf$. All context vectors are then clustered based on their cosine similarities. The cosine similarity of two vectors A and B is defined as:

$$Cosine\_Similarity(A, B) = arccos \frac{A \cdot B}{||A|| \cdot ||B||} \tag{2.7}$$

Each cluster represents one relation and is characterized by common words as cluster labels in its context vectors. However, it is difficult to choose the total number of clusters or to set a threshold on cosine similarity when clustering. Besides, less common (frequent) words that contain important information would likely be ignored. Although only tested in The New York Times (1995) in [39], this method may be applied to the identification of other kinds of relations.

Applying the kernel methods to relation classification based on the dependency tree is discussed in [24]. Context vectors are also used in this system, but augmented by the syntactic and semantic features of words. Each context vector is represented by a **Dependency Tree** and is classified by an SVM based on the similarity with other vectors, which is obtained from the kernel method applied to the dependency tree. A dependency tree is a representation that denotes grammatical relations between words in a sentence. **Kernel** methods are non-parametric density estimation techniques that compute a **Kernel Function** between data instances, where the kernel function can

be thought of as a similarity measure. Two kernel functions are defined over features of the dependency tree: the matching function and the similarity function that count for the similarity among vectors and among features respectively. Their system was tested on news text and would be applicable to the biomedical relation identification task.

2.2.6 Graph-based extraction

Graph mining, based on compiling relations with semantic and syntactic information and searching for common structural patterns in sub-graphs is implemented and named **Lexical Networks** in BioTeKS [67]. Lexical networks apply data-mining techniques to graphs that are derived from syntactic parse trees, where the nodes in a graph represent proteins, and the links represent relatively strong co-occurrences between these proteins within a sentence or paragraph. Figure 2.5 shows an example of the lexical network collected from a set of 600 MEDLINE documents. The strong co-occurred proteins are linked with strength. Although the pair-wise term relations can be computed in documents to be compiled into longer sequences that span multiple documents, lexical networks are basically the visualization tool of unnamed relation between proteins and are not used to extract hidden information.

Jessen et al. introduce PubGene, a system that extracts explicit and implicit biomedical knowledge of human genes from MEDLINE abstracts by creating a gene-to-gene co-citation network [47]. The system builds the network by linking two genes if they occurred in the same article and represents each gene in the database by a node in the network. As an indication of strength, each pair of genes is given a weight equal to the number of articles in which the pair was found. The extracted network is validated by three large-scale experiments showing that co-occurrence correctly reflects biologically meaningful relations (60% and 71% for low-weight and high-weight gene pairs respectively).

Acting as an inference network[13], Biobilmetrics is a tool for efficiently exploring biomedical information [104]. As introduced in Section 2.2.2, it constructs a matrix that contains dissimilarity measurements of every pair of genes, based on their

[13]An inference network is basically a Bayesian network to model documents [115].

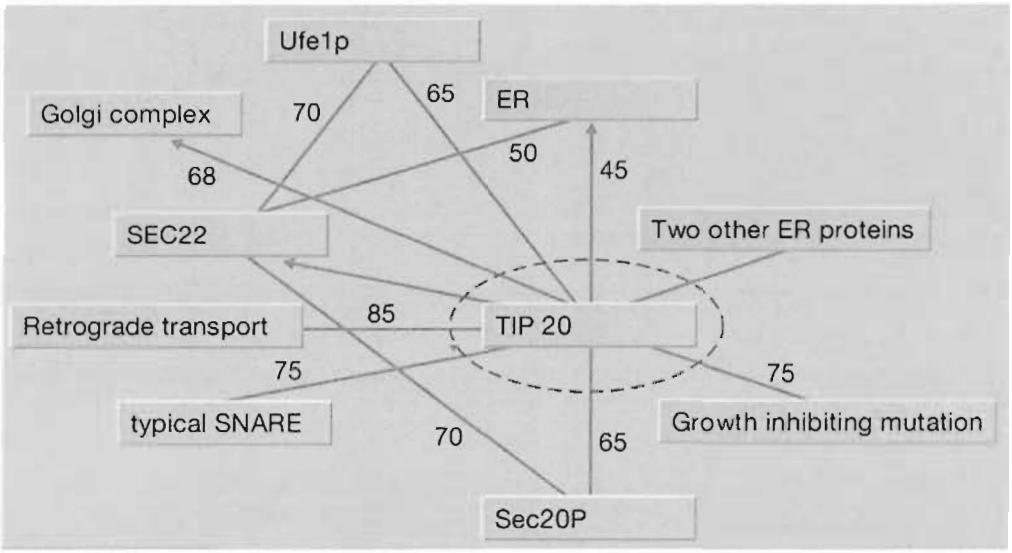

Figure 2.5: A lexical network showing co-occured proteins collected from 600 MEDLINE documents [67].

joint and individual occurrence statistics. A graph is then generated from this matrix. Nodes of the graph represent genes and can be hypertext-linked to sequence databases, while edge lengths are a function of the occurrence of the two genes within the literature and are linked to those MEDLINE documents that generated them. Tested on MEDLINE documents published between 1997 and 1998 and containing the MeSH term *Saccharomyces Cerevisiae*, their system is able to extract knowledge latent with retrieved information. However, since the graph is built on co-occurrences of gene pairs, edges are still representations of un-named relations and may not reflect actual relations.

Leroy and Chen build a co-occurrence-based semantic net representing Concept Space relations in Genescene [60]. Concept Space is a bottom-up technique that captures relations between pairs of NPs from large collections of text. It provides a network of semantically-related concepts that form relations for the entire collection. Each relation is directional and contains two NPs and a weight of co-occurrence analyzed based on the asymmetric cluster function [16] to indicate its strength of

relevance. For example, from the sentence:

Example 2.6: Proliferation and apoptosis were assessed in a panel of NSCLC cell lines that ...

two relations are introduced into the network: *Assess In(apoptosis, panel)* and *Of(panel, NSCLC cell)*. The Concept Space relations are more precise when selected with ontological knowledge (GO, HUGO nomenclature and UMLS).

Proux et al. construct the semantic representation of sentences based on Conceptual Graph from the syntactic dependencies extracted by the parser [88]. The verb is placed at the top of the conceptual graph structure symbolizing the sentence. Nouns appearing in subject or object group, are connected to this verb through links representing their syntactic relation. Figure 2.6 shows an example of a conceptual graph built from the sentence:

Example 2.7: ems directly regulates sc function.

User requests are stored in the system using exactly the same structure. The extraction mechanism then tries to establish a projection between user request graphs and the semantic representation of sentences to detect matching patterns. A projection between two graphs is accepted if and only if one of them contains concepts and relations that are all more abstracted than those of the other graph [88].

Concept Chain Graphs (CCGs) for discovering unknown associations between concepts are introduced in InfoXtract [103, 102]. A chain graph is a probabilistic network model that mixes undirected and directed graphs to give a probabilistic representation that includes Markov random fields and Markov models. It is a hybrid probabilistic IR framework combining a traditional bag-of-words model with higher-level concepts and relations provided by an IE system. The CCG is implemented as a multilevel index where the highest level represents relations, the middle level represents concepts, and the lowest level represents a word index. Figure 2.7 shows concepts (such as protein names, gene names, relations terms, etc.) linked to the gene CDKN1A. These are weighted based on frequency of occurrence. The CCG is applied to discovering

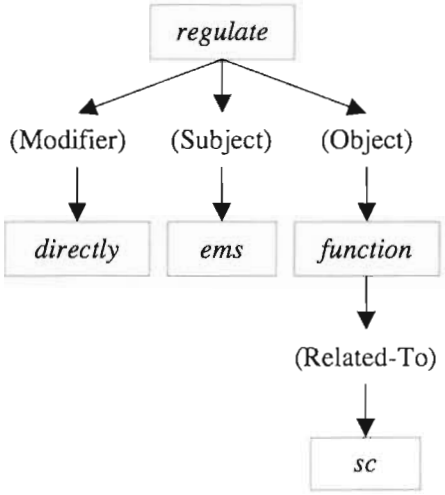

Figure 2.6: An example of conceptual graph from the syntactic dependencies extracted by the parser [88].

unapparent relations and finding paths connecting two and more concepts from a set of relevant documents. It can also be used to generate relevant document summary and narrative description of users.

Summary

Graph-based biomedical relation extraction aims to interconnect biomolecular substances with a graph-like structure based on relationship predictions among them. Most of graph-based approaches are capable of visualizing the relations between terms, due to the natural characteristic of graph representations. In addition, some of them attempt to learn unapparent relations and even pathways from the graph representations, with assumption that nodes interconnected indirectly (with other nodes in between) may be functionally related to each other.

Graphs implemented in these systems can be categorized as directed, undirected and hybrid. Generally speaking, directed graphs (e.g. Bayesian networks) characterize named relations, while undirected graphs (e.g., based on co-occurrence only) represent unnamed relations. Therefore, some unnamed relations may not reflect actual relations.

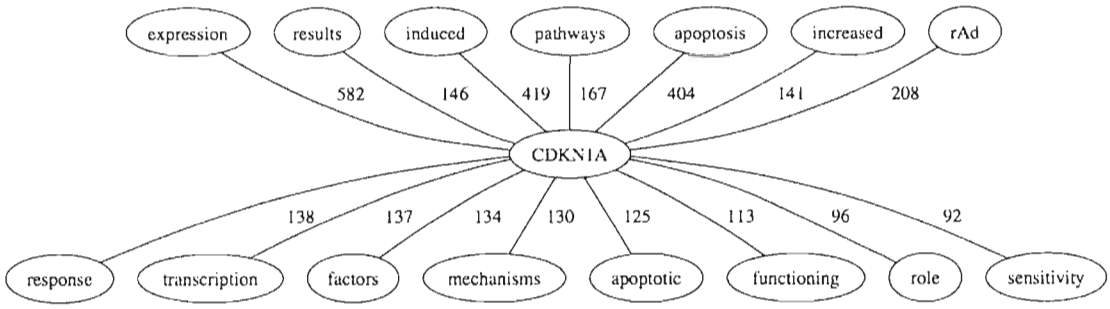

Figure 2.7: Concepts linked to the gene CDKN1A [103].

2.3 Subcellular Localization Extraction

In this section we examine research on extraction of a particular biomedical functional relation: **Subcellular Localization** (SCL). The focus on our work, **Bacterial Protein Localization** (BPL), is one type of SCL. As introduced in Chapter 1, BPL is a key functional characteristic of bacterial proteins. The example:

Example 2.8: E. Coli produces [$_{LOCATION}$ membrane-bound] [$_{PROTEIN}$ lytic trans-glycosylase] and localizes it in [$_{BACTERIUM}$ murein sacculus].

indicates a *membrane-bound* localization relation between *lytic trans-glycosylase* and *murein sacculus*. BPLs are essential to the understanding of the function of different proteins and the discovery of suitable drugs, vaccines and diagnostic targets.

Nair and Rost build a subcellular classifier on keywords of functional annotations of proteins in the SWISS-PROT sequence database [76]. They first map each protein annotation onto a keyword-based vector space, i.e., representing the presence of a certain keyword by 1 and the absence by 0. They then use a classifier consisting of vectors of examples with known localization, to classify examples with unknown localization based on keyword information gain. They concentrate on 10 subcellular localizations (see Table 1 of [76]). Their work is elaborated on by Eskin and Agichtein [31], who combine text and sequence analysis by adding subsequences from proteins amino acid sequence as part of terms in the text representation. However, their evaluation results do not suggest an improvement over [76].

Stapley et al. represent yeast proteins as vectors of weighted terms from all the PubMed articles mentioning their respective genes [105]. A support vector machine (SVM) is trained on protein-text vectors to distinguish among subcellular localizations. Their system outperforms a baseline trained on amino acid composition alone, but it is not tested against existing systems and their evaluation results do not demonstrate any improvement over earlier systems. Their experiments also indicate that, by combining amino acid composition with text, the system does not significantly improve performance with respect to the text-based classifier alone.

Another use of molecular function terms to extend sequence-based subcellular localization prediction is proposed by Lu and Hunter [66]. Different from [76] and [31] that explore SWISS-PROT protein annotations, Lu and Hunter extract the relation between GO function annotations and localization information, identifying both highly predictive single terms and terms with large information gain with respect to location, the same method adopted in [76]. Their system exhibits an improvement by the addition of function information over sequence alone. However, the experiment results does not compare nor suggest any improvement over existing systems.

LOCtree [77] is a hierarchical system combining SVMs and other prediction methods to predict SCLs from SWISS-PROT function annotations. They build a hierarchical architecture for each of non-plant, plant and prokaryote SCL tasks as illustrated in Figure 2.8. Each node in the architecture is a binary classifier and is implemented using an SVM. Accuracy of LOCtree (78%) on extraction of five SCLs (i.e., *extra-cellular*, *nuclear*, *cytosolic*, *mitochondria* and *chloroplast*) indicate a significant improvement over SubLoc (57%), PSORT (51%) and NNPSL (52%).

Hoglund et al. predict subcellular localizations from both text and protein sequence data [40]. They first apply SVMs to make predictions from protein sequence data, and then they weight from the text the terms co-occurred with the location name (organism) and assign each protein name a vector based on this terms co-occurred with the protein. Finally an SVM is applied on all protein vectors generated from the sequence data and text. Their evaluation results show a significant improvement over

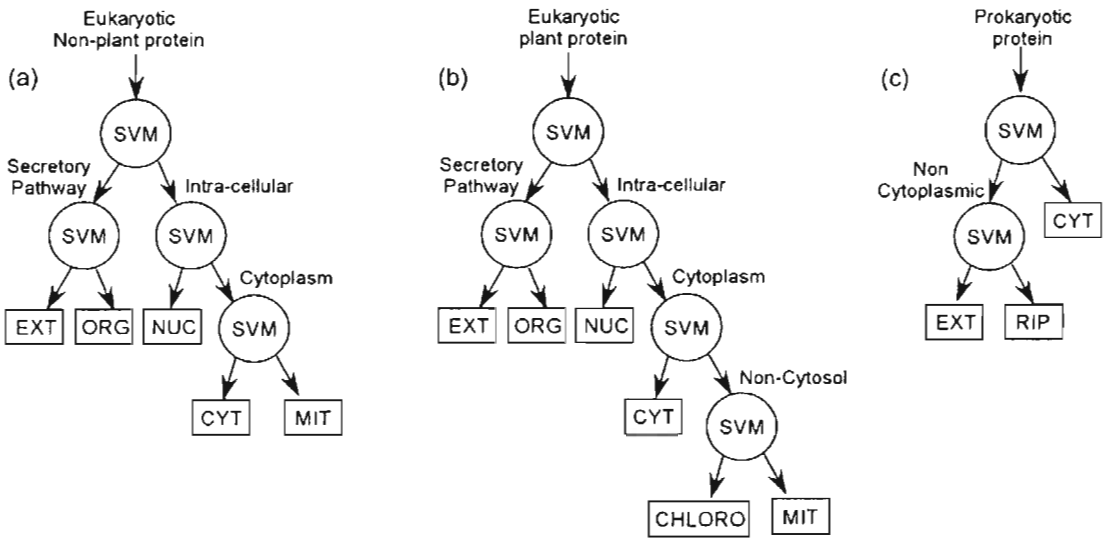

Figure 2.8: Hierarchical architecture of LOCtree [77].

TargetP[14], PLOC[15] and MultiLoc[16].

Summary

In this section, we introduced systems that extract SCLs from biomedical sources. Some systems combine both textual information and sequence data and show improvement over sequence-based extraction alone. However, these approaches either miss the actual location information in their predicted localizations or only focus on a small portion of eukaryotic proteins, while we aim at bacteria-protein-location relations for bacterial proteins.

Moreover, lexical analysis implemented in these systems limits itself at the bag-of-words level. No serious linguistic approaches, such as syntactic or semantic analysis, have been attempted. In the subsequent chapters, we will propose several approaches

[14]TargetP data set contains a total of 3,415 distinct proteins representing four plant (ch, mi, SP, and OT) and three non-plant (mi, SP, and OT) localizations.[30]

[15]POLC consists of proteins extracted from Swiss-Prot release 39.0, covering 12 localizations [83].

[16]The MultiLoc data set contains a total of 5,959 protein sequences, which were extracted from the Swiss-Prot database release 42.0 [41].

that apply and combine syntactic and semantic information to extract SCLs.

2.4 Performance of Related Work

Table 2.1 shows tasks, data sets and experimental results of some biomedical relationship identification systems introduced in this chapter. Systems are grouped by task: protein/gene relation identification, SCL identification and others.

Inside each task group, systems are sorted by data set and then performance. MEDLINE and SWISSPROT are among the most used data sets, while some systems built their own data sets or did not specify what data sets were used in their papers.

The table also shows that systems using the same data set in the same group are comparable to each other in terms of performance. For instance, [29] and [43] extracted protein-protein relations from MEDLINE records and their F-scores are around 92%; [76], [77] and [31] extracted SCLs from SWISS-PROT and also performed closely to each other. As for the rest of systems, we cannot simply compare their performances by experimental results, since the tasks and data sets they used are different, and their evaluation methods may be different too.

System	Task	Data set	Performance (P/R)
preBIND [29]	protein-protein	MEDLINE	0.92/0.92
[43]	protein-protein	MEDLINE	0.91/0.93
[97]	gene-protein	MEDLINE	0.73/−
[88]	gene-gene	Flybase	0.81/0.47
[33]	gene-gene	-	Confidence: $0.85 \sim 1.00$
SUISEKI [8]	protein-protein	-	$0.5 \sim 0.8 / > 0.7$
[82]	protein-protein	-	0.48/0.80
Highlight [112]	protein-protein	-	$0.69 \sim 0.77 / 0.29 \sim 0.58$
GeneScene [62]	gene-gene	-	0.70/0.47
MedScan [80]	protein-protein	-	0.91/0.21
[47]	protein-protein	-	$0.60 \sim 0.72$/−
[68]	protein-protein	-	−/0.71
[40]	SCL	TargetP etc.	0.85/0.86 (for *plant*)
LOCkey [76]	SCL	SWISS-PROT	Accuracy: 0.82
LOCtree [77]	SCL	SWISSPROT	Accuracy: 0.78
[31]	SCL	SWISS-PROT	0.80/0.75
[105]	SCL	MEDLINE	F1: $0.33 \sim 0.82$
Medstract [90]	-	MEDLINE	0.90/0.57
ARBITER [93]	binding	MEDLINE	0.73/0.51
[20]	-	MEDLINE	0.51/0.44
BITOLA [44]	drug-gene	MEDLINE	0.55/0.12
MedSynDiKaTe [38]	build ontology	-	$0.80 \sim 0.93 / 0.81 \sim 0.93$
GENIES [33]	pathway	-	0.96/0.63
PASTA [36]	protein structure	-	0.86/0.67
PubGene [106]	-	-	0.89/0.61
[23]	protein-disease-drug	YPD	0.92/0.21
[119]	-	-	−/0.23

Table 2.1: Performances of some biomedical relationship identification systems

Chapter 3

A Biomedical Information Retrieval System

Information Retrieval (IR) is a process to find documents relevant to a query. The query can be in various forms from one or a few keywords to a complex well-formed question. Compared to information extraction (IE), which can be viewed as the process of finding more detailed and finer information such as names and relations, IR is the process of providing entire documents relevant to a query. So IR is generally token as a step precedentto the fine information extraction process.

Before introducing our relation extraction system, in this chapter we propose a biomedical IR system[1] as a coarse level of relation extraction. The system participated in the TREC 2005 ad-hoc retrieval task in the Genomics track, at which it attempted to find documents relevant to answers of complex questions. Example 3.1 shows one of such questions.

Example 3.1: *Provide information about the role of the gene DRD4 in the disease Alcoholism.*

The main approach taken in the IR system is to expand synonyms by exploiting a fusion of a set of biomedical and general ontology sources, and apply machine learning and natural language processing techniques to re-rank retrieved documents. In our

[1]This is a joint work with Baohua Gu, at the Natural Language Processing Lab, Simon Fraser University.

system, we integrate EntrezGene[2], HUGO[3], Eugenes[4], ARGH[5], GO[6], MeSH[7], UMLS[8] and WordNet[9] into a large reference database and then use a conventional Information Retrieval (IR) toolkit, the Lemur toolkit [58], to build an IR system. In the post-processing phase, we applied a boosting algorithm [53] that captures natural language sub-structures embedded in texts to re-rank the retrieved documents. Experimental results show that the boosting algorithm works well in cases where a conventional IR system performs poorly, but this re-ranking approach is not robust enough when applied to broad coverage task typically associated with IR.

3.1 Introduction

The TREC 2005 Genomics track consists of the ad-hoc retrieval task and the categorization task. We were participating in the ad-hoc retrieval task only, due to the considerable effort we spent on building the framework of the biomedical IR system.

The ad-hoc retrieval task aims at the retrieval of MEDLINE records relevant to the official topics. In constrast with the free-form topics of the 2004 task, the 2005 topics are more structured and better defined. A set of 5 generic topic templates (GTTs) was developed following the analysis of the the 2004 topics and the information needs from 25 biologists[10]. Ten topic instances were then derived from each of GTTs. As with the ad-hoc retrieval task in 2004, the document collection of the 2005 task is a 10-year MEDLINE subset (1994-2003), about 4.6M records and 9.6G bytes in total. The relevance judgement was made by the same pooling method used in the 2004 task, where top ranking documents of every topic from all submitted runs are given

[2]EntrezGene at National Center of Biotechnology Information, http://www.ncbi.nlm.nih.gov/entrez/.

[3]HUGO at Gene Nomenclature Committee, http://www.gene.ucl.ac.uk/nomenclature/.

[4]Eugenes: Genomic Information for Eukaryotic Organisms, http://eugenes.org/.

[5]ARGH: Biomedical Acronym Resolver, http://invention.swmed.edu/arth/.

[6]GO: The Gene Ontology. http://www.geneontology.org/.

[7]MeSH: Medical Subject Headings, http://www.nlm.nih.gov/mesh/meshhome.html.

[8]Unified Medical Language System at National Institute of Health, http://www.nlm.nih.gov/research/umls/.

[9]WoreNet: a lexicon database for English language, http://www.cogsic/princeton.edu/wn/.

[10]http://ir.ohsu.edu/genomics/2005protocol.html

to human experts, who then determined if each document is either definitely relevant (DR), possibly relevant (PR) or not relevant (NR) to the topic.

Three run types were accepted in the Genomics track: automatic, manual and interactive, which differed depending on how the queries were constructed. Each participant was allowed to submit up to two runs. Our submission was in the manual category, since our queries were manually constructed. One of our goals was to determine how natural language processing (NLP) techniques could be used for re-ranking in a post-retrieval step. In our current system, we only apply such techniques for re-ranking. In the future we plan to apply similar techniques towards query expansion.

3.2 System Architecture

In general, the performance of an IR system largely depends on the quality of the query expansion. Most participants of the ad-hoc retrieval task in previous years applied reference database relevance feedback, a technique that finds synonyms and relevant terms from the outside term databases and adds them in the query. Over the past decade, the biomedical databases have evolved dramatically in terms of both the number and the volume, but from the reviews of previous work in this task, most of participants only employed a couple of them to build the reference database. In our system, we collect terms from EntrezGene, HUGO, euGenes, ARGH, MeSH, GO, UMLS and WordNet, and integrate them into a large reference database, which we then use in our system.

Traditional NLP techniques have been generally un-successful in improving retrieval performance [116], but there is still interest in examining how the linguistic and domain specific knowledge contained in NLP models and algorithms might be applied to specific IR subtasks to improve performance. In this work, we applied a classification technique: a boosting algorithm to capture sub-structures embedded in texts [53] in the second phase of our IR system. Different from the typical bag-of-words approach, the algorithm takes each sentence as a labeled ordered tree and classifies it by assigning a relevance score as either relevant (positive) or not (negative). The relevance of each document is then calculated from relevance scores of the sentences in the document.

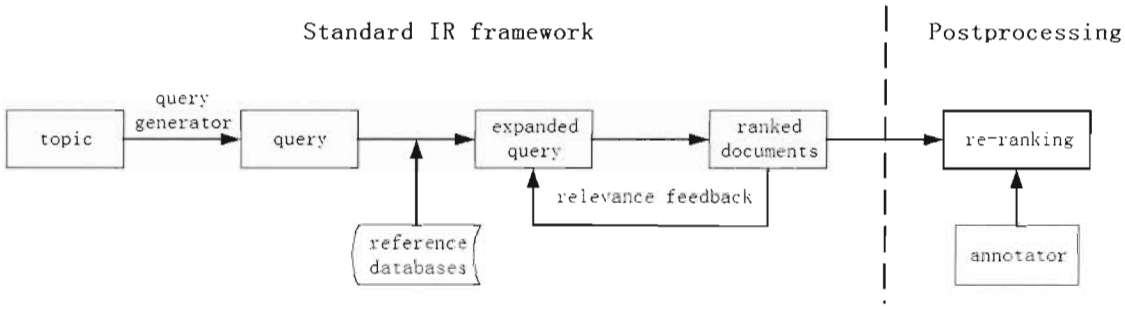

Figure 3.1: The system architecture

Our system consists of two major phases, shown in Figure 3.1. In the first phase (left of the dashed line in Fig 3.1), we applied extensive synonym expansion with a conventional IR system, the Lemur toolkit 4.1 [58]. The details of our synonym expansion phase and reference database construction are introduced in Section 3.3. The second phase is a post-processing step, in which the boosting classification algorithm [53] was used to re-rank the list of retrieved documents from the first phase. Section 3.4 describes its implementation details, experiments and evaluations of the boosting-based classification.

3.3 Conventional IR Module

3.3.1 Extensive Synonym Expansion

Our system involves the manual selection of key words from the official topics (for most topics the key words were already given in the tabular version of topics) according to the given GTTs. The names and symbols related to each key word, for instance, synonyms, acronyms, hyponyms and similar names, were then matched with the public biomedical and generic databases that include synonyms and relevant terms. Specifically, for gene/protein names, we automatically integrated EntrezGene, HUGO, Eugenes and ARGH into a large gene/protein database with 1,620,947 entries, each of which consists of names and symbols that represent the same biomedical substance, and then matched them with each key word in the topics. Similarly, for diseases, organisms and drugs, related names and symbols were automatically matched

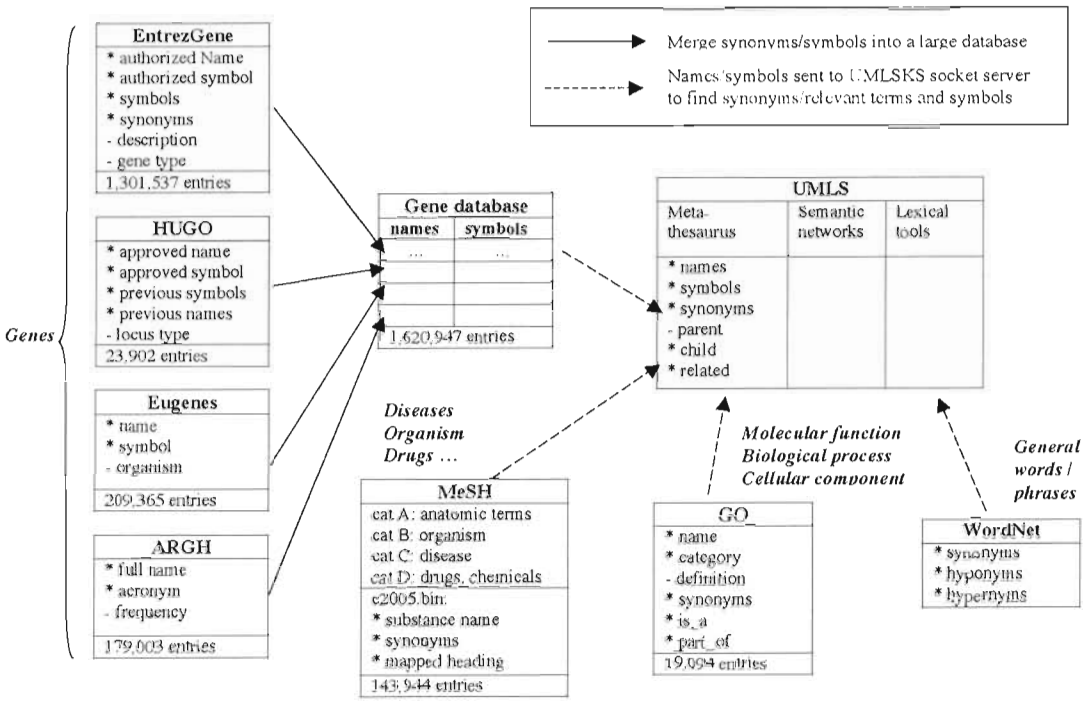

Figure 3.2: Extensive synonym expansion

with entries in MeSH; molecular functions, biological processes and cellular components made use of GO, and general words/phrases were matched (manually so far) in WordNet. In addition, all sets of related names and symbols were further expanded by searching via the UMLS Knowledge Source (UMLSKS) Socket Server. Figure 3.2 illustrates the procedure of constructing the reference databases.

3.3.2 Document Retrieval

In this project, we use the Lemur Language Modeling Toolkit 4.1. The Lemur system was designed to facilitate research in language modeling and information retrieval (IR), such as the ad-hoc and distributed retrieval, structured queries, cross-language document retrieval, summarization, filtering, and categorization.

We use the following three modules provided in Lemur 4.1:

1. Parsing Query module

2. Building index module

3. Structured Query Retrieval Module

In the following subsections, we will briefly describe how each module was used in our system.

Parsing Query

The Parsing Query module contains two utilities to handle different types of queries: ParseQuery and ParseInQueryOp. ParseQuery handles queries written in NIST's Web or TREC formats, while ParseInQueryOp is used to parse structured queries written in a structure query language. Both types of queries are then converted into the BasicDocStream format, an document format used inside Lemur. In our experiments, we tried both types of queries and found that the structured queries generally provided better results. Therefore, we used the structured queries in our submitted run.

The structure query language used in Lemur can be found on its web site[11]. Briefly, it allows a user to define various AND/OR/NOT relations, and it provides for weights of sums (WSUM) among the terms. It even allows a user to consider a sequence of single terms by defining them as a phrase. Hence, the structured query enables more precise query definition. A sample structured is shown in Example 3.2.

Example 3.2

$q135 = \#band($
$\#or($
$\#phrase(cellgrowth)$
$\#phrase(cellexpansion)$
$\#phrase(CellularExpansion)$
$\#phrase(CellularGrowth)$
$)$
$\#or($
$\#phrase(Bop)$

[11] http://www.lemurproject.org/lemur/StructuredQuery.html

$\#phrase(bacterio - opsin)$
$\#phrase(bacterioopsin)$
$\#phrase(bacterio - opsingene)$
$\#phrase(bop)$
$\#phrase(BiocompatibleOsteoconductivePolymer)$
)
);

Building the Index

Lemur's BuildIndex module supports constructiion of four types of indices, specifically: InvIndex, InvFPIndex, KeyfileIncIndex, and IndriIndex[12]. We used the KeyfileIncIndex, which includes the position information of a term and can be loaded faster than InvIndex and InvFPIndex while using less disk space than IndriIndex.

Retrieving Structured Query

The structured queries were passed to the StructQueryEval module, which ran retrieval experiments to evaluate the performance of the structured query model using the inquery retrieval method. Note that for structured queries, relevance feedback was implemented as a WSUM of the original query combined with terms selected using the Rocchio implementation of the TFIDF retrieval method [96]. In our official runs, the parameters (feedbackDocCount, feedbackTermCount, feedbackPosCoeff) for relevance feedback are: (100, 100, and 0.5).

3.3.3 Evaluation

Among all the official runs submitted to the ad-hoc task of the TREC-2005 Genome Track, 48 are using automatic retrieval methods and 12 including ours are manual ones. Figure 3.3 shows MAP (upper), P10 (middle) and P100 (lower) scores of the manual runs. Three runs are shown in the figure: the best, the worst and ours on each topic. To better illustrate the performance of our system among others, we plot each value in the figure as the differenece between the actual score and the median score.

[12]http://www.lemurproject.org/lemur/indexingfaq.html

Although we do not know the evaluation results of every other system, Figure 3.3 seems to indicate that our system is above the average. For instance, for the P10 scores of our system on all 49 topics, 36 are above the median and 10 of them are the best; for the MAP scores, 32 are above the median and 2 are the best. The automatic runs perform better than the manual runs on the whole and our system is around the average of the automatic runs. Our future research will involve the invetigation of how our system performs on each topic and each template, looking for insights to further tune our system.

3.4 Post-processing Module

3.4.1 Boosting-based Classification

Traditional NLP techniques, such as word sense disambiguation resolution, chunking and parsing, were examined in the IR community at the TREC-5 NLP track, but few of them were shown successful for good retrieval performance. The reasons may lie in the broad coverage of the typical retrieval task, the lack of good weighting schemes for compound index terms and the statistical nature of the NLP techniques [116].

However, the attempts of applying NLP and machine learning techniques to the IR tasks are still attractive, since a good understanding of the documents could be a breakthrough to the IR tasks. In this project, we adopted Taku Kudo's Boosting Algorithm for Classification of Trees (BACT), a classification method that captures the sub-structures embedded in texts. We use the method and implementation described in [53]. BACT takes a set of all subtrees as the feature set, from which it iteratively calls a weak learner to produce weak hypotheses. The strong hypothesis is finally generated by a linear combination of weak hypotheses.

We incorporated BACT into the post-processing step, where the list of retrieved documents from Lemur was re-ranked by taking the classification of the documents into account, as shown in the Figure 3.4. The documents in the training data were parsed using Charniak's parser [15] and then classified by BACT in terms of relevant (positive) or irrelevant (negative). A re-ranking mechanism made the final relevance decision by combining the relevance scores from both Lemur and BACT.

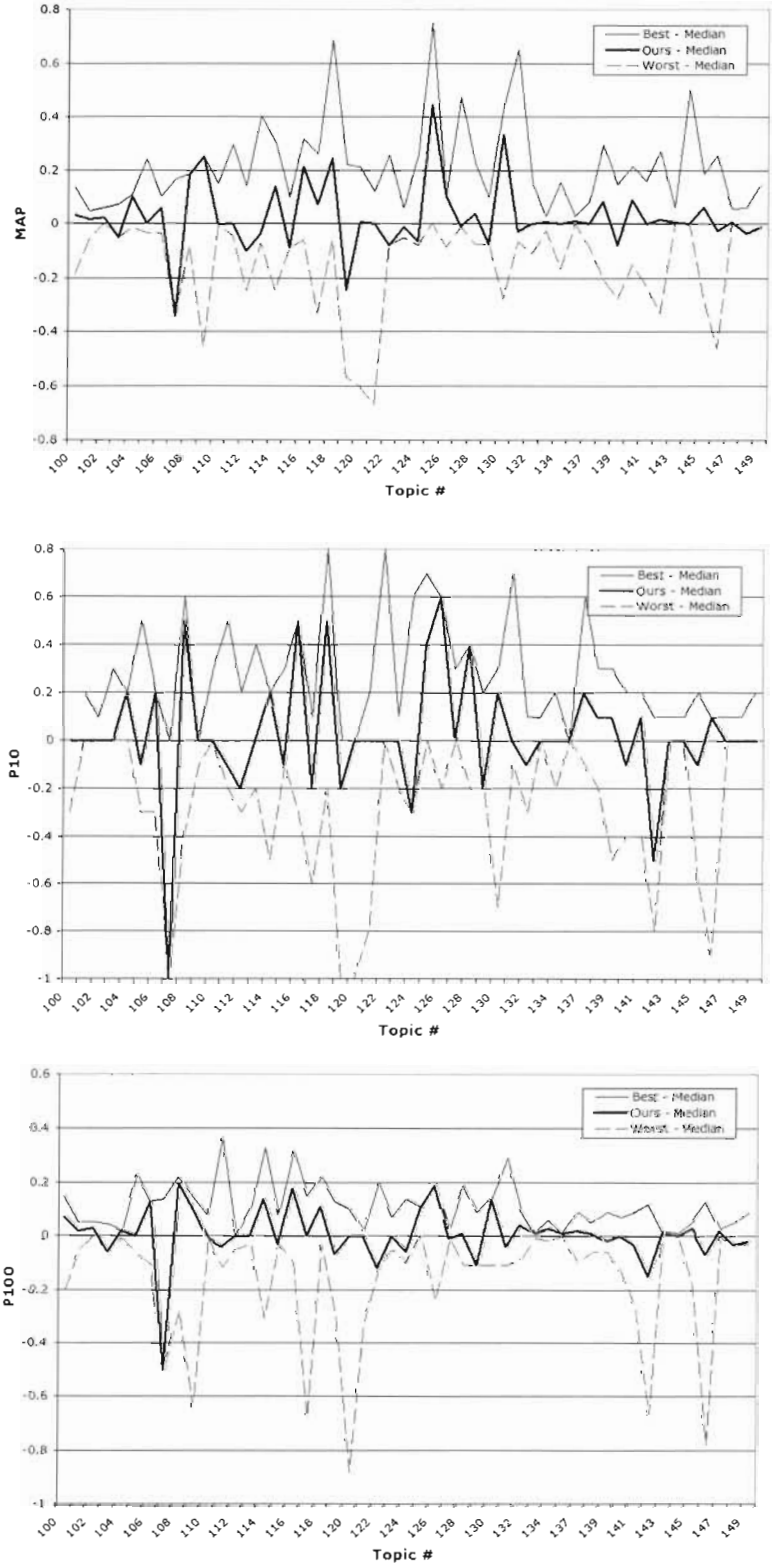

Figure 3.3: The MAP, P10 and P100 scores of the best, worst manual runs and our system on each topic. Each value is the actual score minus the median.

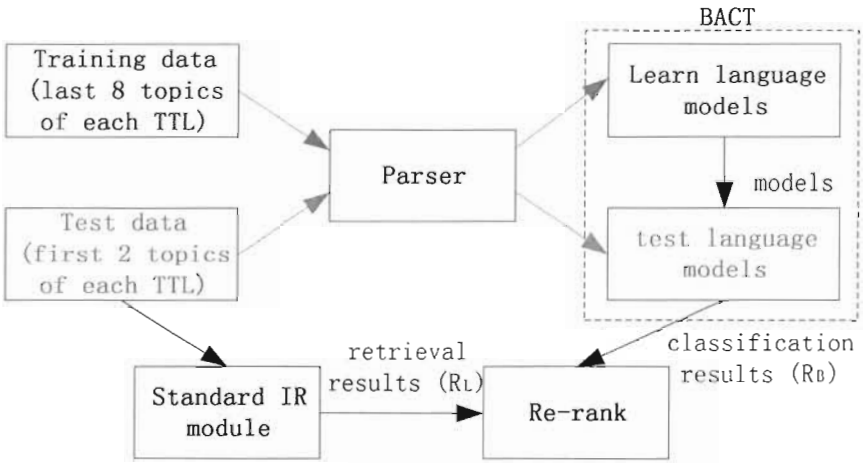

Figure 3.4: The post-processing phase

The major difficulty of applying BACT in this task is that it assigns a classification score (positive or negative) to each sentence rather than assigning a score to a document. This results in two issues: 1) the lack of the training data with the label for each sentence; 2) the lack of a mechanism for combining sentence scores into a document score.

Since we lacked training data of sufficient quality and quantity for the classification task, we were not able to submit the post-processing results to the TREC in time for the initial deadline. After the results of the ad-hoc retrieval task were announced (on Sept. 30, 2005), we were able to test the performance of the post-processing, by taking the following steps to prepare the training and test data for BACT:

1. The retrieved documents in the first two topics in each TTL were taken as the test data and those in the remaining topics as the training data.

2. The irrelevant documents in the training data were removed due to the unbalance of the training data (irrelevant documents are much more common than relevant ones).

3. In the training data, sentences were given "approximate" labels by matching them against a disjunction of all terms in the corresponding query as either matched ($+1$) or unmatched (-1).

BACT assigned a real number as the classification score to each sentence, with a larger score corresponding to a more relevant sentence. We took the mean of all sentence scores in each document as the document score.

3.4.2 Re-ranking

The goal of re-ranking is to combine R_L and R_B, the ranks from Lemur and BACT respectively, such that the rank R' maximizes the evaluation scores, for example, MAP, P10 and P100. R_L, R_B and R' are score vectors of retrieved documents. We assumed that such a combination was linear, i.e.:

$$R' = R_L + i * R_B \tag{3.1}$$

We thus looked for i' that maximizes the evaluation function $E(R')$:

$$i' = \text{argmax}_i \, E(R_L + i * R_B) \tag{3.2}$$

3.4.3 Evaluation

As described in Section 3.4.1, we extract relevance scores of our retrieved documents (by Lemur) from the evaluation results of 2005 ad-hoc retrieval task. For each TTL, the retrieved documents of the first two topics were taken as the test data, and those of the remaining topics as the training data.

Table 3.1, 3.2 and 3.3 list the MAP, P10 and P100 before ($i = 0$) and after the re-ranking for the TTL #1, #2 and #3. A linear combination coefficient i' was predicted for each TTL following Equation 3.2. For the TTL #2, i' converges at 15 and the linear combination model significantly improves the IR performance: MAP increases from 0.0012 to 0.0024 for the topic #110 and from 0.0492 to 0.1602 for the topic #111; Same situations for P10 and P100. However, for the TTL #1 and #3, no linear combination model can improve the IR performance, i.e., $i' = 0$. The scores at $i = 10$ are also listed in Table 3.1 and 3.3 to show that the performance dropped when the linear combination models were applied.

Topic #	Metrics	$i = 0(i')$	$i = 10$
100	MAP	0.2221	0.1785
	bpref	0.8649	0.8649
	P10	0.4	0.3
	P100	0.28	0.22
101	MAP	0.0685	0.0195
	bpref	0.75	0.75
	P10	0	0
	P100	0.07	0.01

Table 3.1: Performances of re-ranking on the TTL #1

Topic #	Metrics	$i = 0$	$i = 15(i')$
110	MAP	0.0012	0.0024
	bpref	0.25	0.25
	P10	0	0
	P100	0	0.01
111	MAP	0.0492	0.1602
	bpref	0.4356	0.4356
	P10	0.1	0.7
	P100	0.1	0.4

Table 3.2: Performances of re-ranking on the TTL #2

Topic #	Metrics	$i = 0(i')$	$i = 10$
120	MAP	0.6113	0.2410
	bpref	0.8145	0.8145
	P10	1	0.3
	P100	0.88	0.29
121	MAP	0.6697	0.0328
	bpref	0.8810	0.8810
	P10	0.8	0
	P100	0.34	0

Table 3.3: Performances of re-ranking on the TTL #3

3.4.4 Discussion

Our experiments show that BACT as the post-processing does help when *bpref* (proportion of judged relevant documents that are retrieved) of the conventional IR system is low, for instance, 0.25 and 0.4356 in the TTL #2. For the TTL #1 and #3 where BACT failed, the average *bpref* is very high, above 0.8.

It seems as if our current use of BACT for re-ranking cannot scale to the broad coverage of relevant documents in the retrieved document set, especially in the case where *bpref* is high. This is a common problem of NLP techniques when applied to the IR task. However, employing machine learning and NLP techniques such as BACT as the post-processing step may help the conventional IR system when the recall is low, by re-ranking the retrieved documents towards a better performance.

The biomedical IR system introduced in this chapter can be taken as a coarse level of information extraction. Starting from the next chapter, we will introduce the task of biomedical relation extraction and propose a system, which extracts more detailed and finer information relevant to a specific molecular biological relation from the biomedical text.

Chapter 4

Task Descriptions

4.1 Introduction

From Chapter 4 to 7, we will introduce the extraction of a specific biomedical relation: subcellular localization, and propose a system that includes various Natural Language Processing (NLP) and machine learning techniques to extract subcellular localizations from biomedical text.

Subcellular Localization (SCL) is one typical biomedical relation. Bacterial SCL states where proteins locate in bacteria. For example:

Example 4.1: E. Coli produces [$_{LOCATION}$ membrane-bound] [$_{PROTEIN}$ lytic transglycosylase] and localizes it in [$_{BACTERIUM}$ murein sacculus].

indicates a *membrane-bound* localization relation between the said bacterium and protein. The bacterial SCL is a key functional characteristic of proteins, since a protein has to be translocated to the correct intra- or extra-cellular compartments or attach to a membrane in order to function properly. This characteristic is essential to the understanding of the functions of different proteins and the discovery of suitable drugs, vaccines and diagnostic targets. Locations of bacterial proteins are listed in Figure 4.1 for Gram+ and Gram- bacteria, which was originally introduced in Figure 1.2.

This thesis introduces our approach for identifying bacterial SCLs from MEDLINE articles. Specifically, our task is to extract from biomedical articles a relation among: a

LOCATION, e.g., *membrane-bound*, a particular BACTERIUM, e.g. *murein sacculus*, and a PROTEIN name, e.g. *lytic transglycosylase*. Therefore, the task is to identify a BPL function, a relation among bacterium, protein and location.

Examples of expected system output are shown in Table 4.1. In many circumstances, such relations are not explicitly given and thus this task may require some level of induction from the context. For instance, "binding of [$_{PROTEIN}$Trpl] to the site I" infers a *cytoplasmic* localization from the fact that Site I is the site to which a DNA molecule binds and that the DNA locates at the cytoplasmic layer.

This work is motivated by our collaboration with molecular biologists, who have built an BPL database for bacterial proteins. These BPLs are either curated manually by the biologists or predicted by an automatic BPL prediction model from the most recent NCBI Taxonomy dataset[1] of completely sequenced genomes. Our task is however to extract BPLs from MEDLINE articles.

The task is new to BioNLP in terms of the specific biomedical relation being sought. Therefore, we have to build an annotated corpus from scratch and we are unable to use existing BioNLP shared task resources in our experiments.

The BPLs extracted by our proposed approach will be ultimately examined by human experts, populated into the SCL database and then used by the biologists to improve the accuracy of the SCL prediction model. We have worked closely with them to ensure that the output produced by our system is directly useful in expanding their protein localization database.

Organism	Localization	Protein	Relevant Sentence	Pubmed ID
Metha. fervidus	CW	slgA	The genes (slgA)...	1712296
Halo. salinarium	C	Hp71	The samples were...	9396829

Table 4.1: Examples of the BPL output

[1]http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Taxonomy

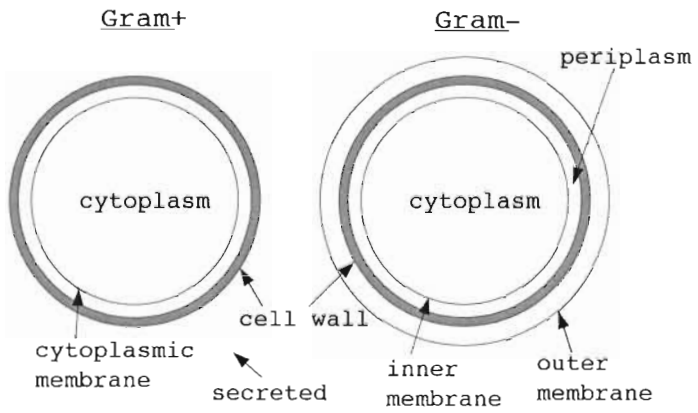

Figure 4.1: Illustration of locations of proteins with respect to the bacterial cell structure.

4.2 Description of Data Sets

Our corpus of biomedical abstracts was obtained by collecting results from several queries to the PubMed Central [12]. Two sample queries are provided in 4.2 and 4.3.

Example 4.2: "pseudomonas aeruginosa", extracellular, protein

Example 4.3: "pseudomonas aeruginosa", "outer membrane", protein

where "pseudomonas aeruginosa" is a BACTERIUM Named-Entity (NE). Each search retrieved thousands of full papers in XML or plain text format. However we only use the abstracts of 12,143 papers, because abstracts are shorter and generally contain denser information, thus taking less processing time.

We are also provided an initial set of training data, consisting of 132 BPL examples with relevant BACTERIUM, PROTEIN LOCATION NEs, passages, PubMedCentral article ID (PMID), from biologists. To expand this small training set, we perform a bootstrapping-based data curation process on the corpus. In each iteration, we apply a sentence classification algorithm to coarsely predict BPL relations and give them to biologists to review. Four iterations have been performed and 333 positive and 1059 negative examples were obtained by this process. Details of the curation process will be described in 5.2.

We randomly separated the data set into training and test sets. Table 4.2 lists numbers of sentences, relation instances, and relevant NEs in each data set. This table will be shown again in Section 5.2.3. Note that the small size of our dataset is a common problem of most of bio-medical information extraction tasks. One of the contributions of this paper is that we show how we can get around this lack of training data by using semi-supervised methods.

Several biomedical ontology sources are available for looking up biomedical terms. The NCBI taxonomy database[2] contains names of all organisms and their taxonomical information. It defines standard and variant genetic code tables for nuclear and organelle genomes, as well as their placements in the taxonomic tree. To date, the NCBI taxonomy database contains 232,631 taxonomy nodes including 772 archaea and 26,142 bacteria.

The Unified Medical Language System (**UMLS**) provides biomedicine and health knowledge sources and associated software tools for building or enhancing electronic information systems as well as for computational research of knowledge representation and information retrieval[3]. UMLS consists of three Knowledge Sources: 1) the Metathesaurus, a large multi-lingual vocabulary database that includes biomedical and health related concepts, their various terms and relations among them; 2) the Semantic Network, an ontology of concepts and their relations; 3) the SPECIALIST lexicon, which provides lexical information needed for the Natural Language Processing systems.

4.3 Evaluation metrics

Relations can be either **Partially Extracted** or **Fully Extracted**. A relation is partially extracted if any relevant NE is recognized. Correspondingly in the full relation extraction, all relevant NEs are recognized. Since full and partial relation extractions provide different level of information about the target relations, they should be distinguished and evaluated differently. In this thesis, evaluations are all based on full relation extraction by default.

[2]http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Taxonomy

[3]http://www.nlm.nih.gov/research/umls/

Data set	**Sentences**	**BPL Instances**	**Relevant NEs**
Positive training set	505	333	768
Positive test set	100	65	160
Negative training set	653	649	0
Negative test set	146	145	0

Table 4.2: Training and test sets: numbers of sentences, BPL instances and relevant NEs

Similarly, NEs can be either **Partially Recognized** or **Fully Recognized**. An NE is partially recognized if any word in the NE is recognized. Correspondingly in the full NE recognition, all words in an NE are recognized. It is also worth emphasizing the difference between full and partial NE recognitions, in terms of the quality of information they provide. In this thesis, evaluations are all based on full NE recognition by default.

Standard definitions for Precision, Recall and F-score for the relation extraction are defined as follows:

$$Precision = TP/(TP + FP) \tag{4.1}$$

$$Recall = TP/(TP + FN) \tag{4.2}$$

$$F1 = 2PR/(P + R) \tag{4.3}$$

where TP is the number of examples with correctly extracted relations, FP is the number negative examples from which the system mistakenly extracts relations, FN is the number of positive examples from which the system fails to extract relations and TN is the number of examples that the system correctly predicts as negative.

The evaluation metrics can be measured against either 1) only examples that contain relations, or 2) any examples. Many relation extraction tasks, such as [76, 77, 31], only evaluate against examples containing relations. However, in real application situations, we need to analyze all examples, in which only a subset of examples have BPL relations. Therefore, we define **Standard Evaluation** method, which consists of the evaluation measures of Definition 4.1 - 4.3 against all test examples. We use this standard method to evaluate all models proposed in this report by default, except

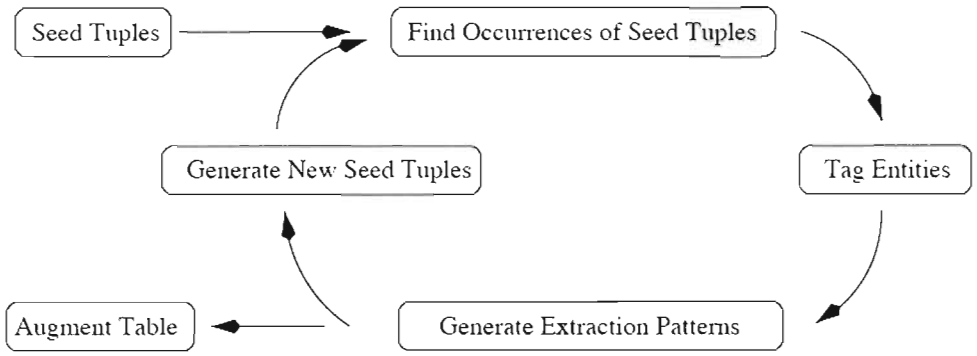

Figure 4.2: High level architecture of Snowball system [2].

some new metrics to be introduced in Section 5. The generative model we propose in Section 5 is trained on positive examples only, and thus it would not be reasonable to test it against negative examples.

4.4 Baseline Systems

Three baseline systems are introduced in this section. The first baseline system is a naive approach, which basically assumes any example containing BACTERIUM, PROTEIN and LOCATION NEs has a BPL relation. The second one is a well-known information extraction system called Snowball. Experiments with these two baseline systems allow us to compare a naive approach to a well-established system. The third one is a discriminative model characterized by word-based features.

4.4.1 Baseline 1: NE Co-occurrence

Since the task is to identify relations among BACTERIUM, PROTEIN and LOCATION NEs, a simple method would be assuming that **any sentence having all three NEs contains the BPL relation**. This NE co-occurrence method would achieve very high recall, since intuitively a BPL relation could not exist without mentions of the relevant NEs. However, our observation is that any sentence having all three NEs does not necessarily contain a BPL relation. It would be interesting to experiment with this method and to measure its performance on precision.

We implemented the NE-co-occurrence method and call it **baseline 1**.

4.4.2 Baseline 2: Snowball

To compare our proposed system with only the naive approach introduced above does not seem convincing. We choose Snowball [2] as the **baseline 2**, which is a well-established relation extraction system. Similar to our proposed system, Snowball extracts relations from text by starting from a small set of training examples of relations. However, Snowball is a bootstrapping-based system and applies only word surface patterns to identify relations, utilizing no information from linguistic analysis. Examples are used to generate patterns, which in turn result in new tuples being extracted from a collection of documents. Figure 4.2 illustrates how Snowball works in the bootstrapping fashion between the pattern extraction and tuple extraction.

4.4.3 Baseline 3: Word-based Discriminative model

In Section 6, we will propose some discriminative models to extract BPL relations. We will also introduce a baseline system, which is a Support Vector Machine featured by uni-grams and bi-grams between PROTEIN and LOCATION/BACTERIUM NEs. It is used exclusively for comparison with the proposed discriminative models.

Chapter 5

Generative Model

5.1 Introduction to the BPL Relation Extraction System

Starting from this chapter, we propose our BPL Relation Extraction system: **BPLRE**. This section provides a high-level overview of BPLRE.

We believe that sentences containing biomedical functional relations can be distinguished from others by certain syntactic and semantic patterns that could be learned statistically. These patterns represent linkages among entities and contain syntactic and semantic information, such as POS, chunking, parsing, named entities and ontologies, all of which are used in this work. Furthermore, the implicit relations across articles would be identified by integrating partial relation information from individual articles.

Our system takes structured MEDLINE documents (e.g., XML documents) and predicts BPL relations from the documents. A preprocessing module is applied to perform syntactic and shallow semantic analysis, including syntactic parsing and biomedical NER, as illustrated in the left side of Figure 5.1. Within the module, a sentence classification coarsely identifies sentences with relations. The module also includes an expert curation process, in which biologists annotate those classified sentences that actually contain BPL relations and indicate related BACTERIUM, PROTEIN and LOCATION NEs. The preprocessing module will be described in Section 5.2.

We then extract BPL relations with a 3-tier approach as shown in the right side of Figure 5.1. Each tier will provide more accurate and complete relation information

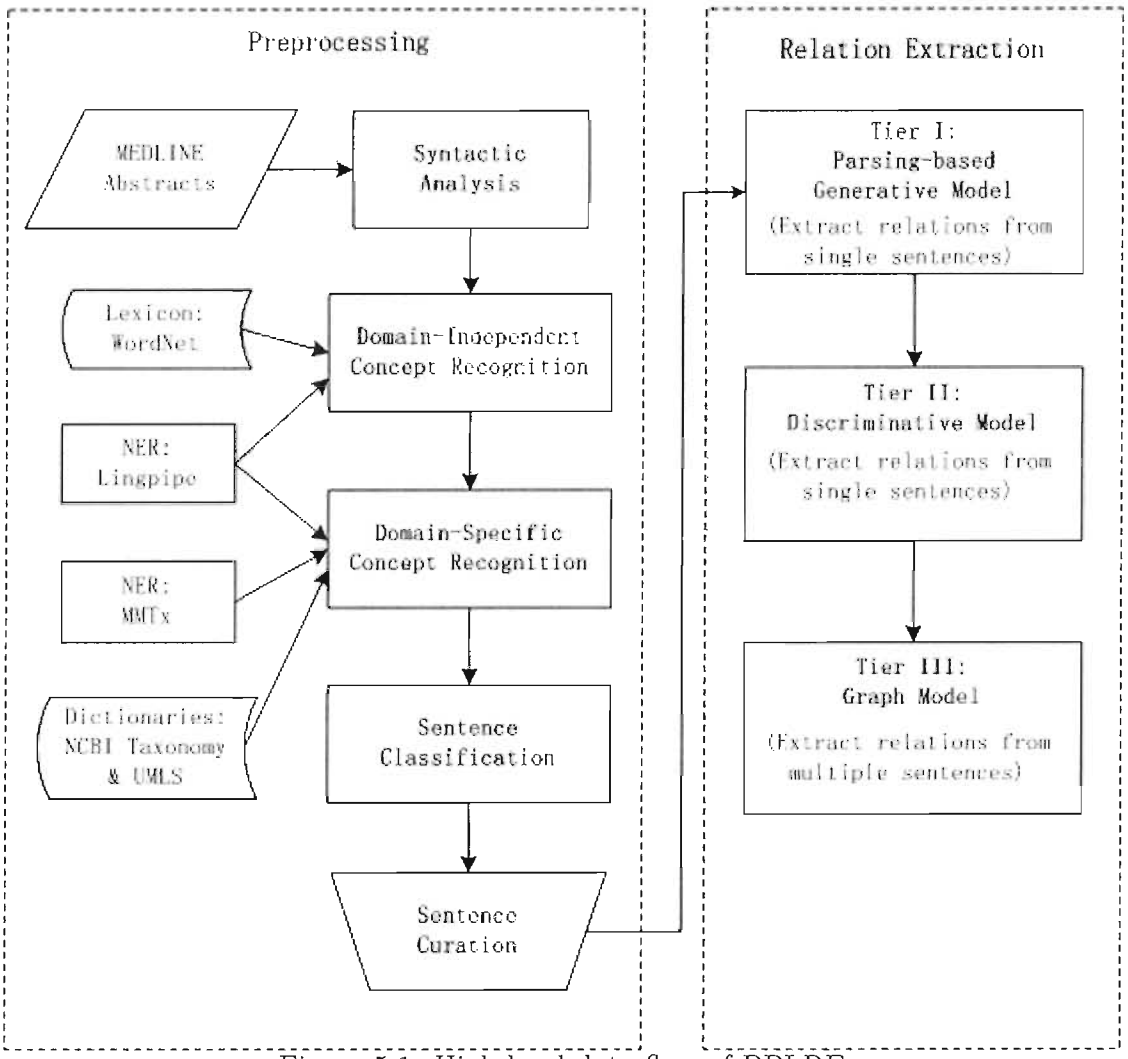

Figure 5.1: High-level data flow of BPLRE

than the previous tier.

- Tier I: A parsing-based generative model to extract relations from single sentences, by analyzing deep syntactic and shallow semantic information (see the rest of Chapter 5).
- Tier II: A discriminative model that integrates rich syntactic features from parse trees to extract relations from single sentences (see Chapter 6). Systems that combine the generative model and the discriminative model are also proposed in Chapter 6, in order to further improve the overall system performance.
- Tier III: Build a graphical representation of relations, find global and hidden relations from multiple sentences and documents, using a graph mode that will be introduced in Chapter 7.

Finally, in Chapter 8 we draw conclusions on the proposed models and also discuss our contributions to this research.

5.2 Preprocessing

In the preprocessing phase, we collect and annotate MEDLINE abstracts, from which we create a small training set for the BPL extraction task, by inviting human experts (curators) to review a set of candidate examples.

5.2.1 Annotation

Documents downloaded from PubMedCentral are in either XML or plain text format. We first extract titles, author information, publication information, abstracts, contents, references and the like from these documents. The Charniak-Johnson reranking parser [14] is then applied for sentence boundary detection and fully syntactic parsing of the abstracts.

The conceptual information consists of both general and domain-specific concepts. WordNet[1] provides us base forms: general words and ontological relations among

[1]Wordnet is a large semantic lexicon database for English, http://wordnet.princeton.edu/.

them, including synonyms, hypernyms and holonyms. Domain-specific concepts are collected by a majority voting from the following two methods:

- Dictionary lookup from the UMLS Metathesaurus, the NCBI Taxonomy, SWISSPROT and GO. This method introduces word sense ambiguity problem and does not identify new entities. We collect NEs related to the task from these sources and build our own dictionary. Numbers of entries of the dictionary from each source are in Table 5.1.

- Applying existing Named-Entity Recognition (NER) tools: Lingpipe[2] and MetaMap Transfer (MMTx)[3], to training the entity identification model. Lingpipe is a free package that performs various tasks including language identification, sentence detection, Part-Of-Speech tagging, text clustering and NER. MMTx is a tool to map arbitrary text to concepts in UMLS Metathesaurus.

Dictionary Sources	UMLS	NCBI	SwissProt	GO
Number of protein names	434,324	-	139,412	-
Number of bacterium names	94,020	384,915	-	-
Number of location names	-	-	-	75

Table 5.1: Types and numbers of entries from dictionary sources

We could also apply other machine learning methods to the identification of domain-specific concepts, but this is not the focus of our research. Our major contribution will be the extraction of functional relations.

5.2.2 Sentence Classification

The initial set of the training data is 132 BPL examples with relevant BACTERIUM, PROTEIN LOCATION NEs, passages, PubMedCentral article ID (PMID) from biologists. However, most of these relevant passages are found in the body of the papers, while we only have annotations of the abstracts.

[2]http://www.alias-i.com/

[3]http://mmtx.nlm.nih.gov/

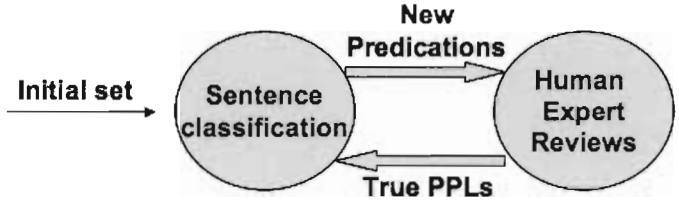

Figure 5.2: The bootstrapping strategy to collect training data

In order to get the first set of training data from the abstracts, for each BPL example, we took sentences including both PROTEIN and LOCATION NEs from abstracts of relevant articles as relevant sentences (please note that these sentences may not be truely relevant) and remaining sentences as irrelevant. The ORGANISM name is excluded from the matching pattern in order to end up with more relevant sentences, since we assume all sentences are more or less talking about pseudomonas aeruginosa. At this point we got 72 relevant and 1197 irrelevant sentences from 148 abstracts.

We trained a classification model with these sentences by a Boosting Algorithm for Tree Classification (BACT) [53], which then predicted 614 sentences as relevant from the whole corpus.

5.2.3 Training Set Curation

From this point a bootstrapping strategy (as illustrated in Figure 5.2) is adopted to build up the training set and experiment with the relation extraction methods: in each round we passed predicted sentences/passages to the curator and then tested our proposed relation extraction method with the curated data.

The curation interface is shown in Figure 5.3. The curator determined the sentence relevance, validations of BACTERIUM, LOCATION and PROTEIN names, as well as appending valid PROTEIN names that are not identified by the Annotator.

Together with four biologists, we made four rounds of prediction and curation. The first set $v1.0$ contains 75 predictions (chosen from 614 sentences), each includes one sentence and all BACTERIUM, PROTEIN and LOCATION NEs identified by the Annotator. 32 predictions are verified true.

In the second set $v1.1$, 87 out of 149 predictions are true. The major differences

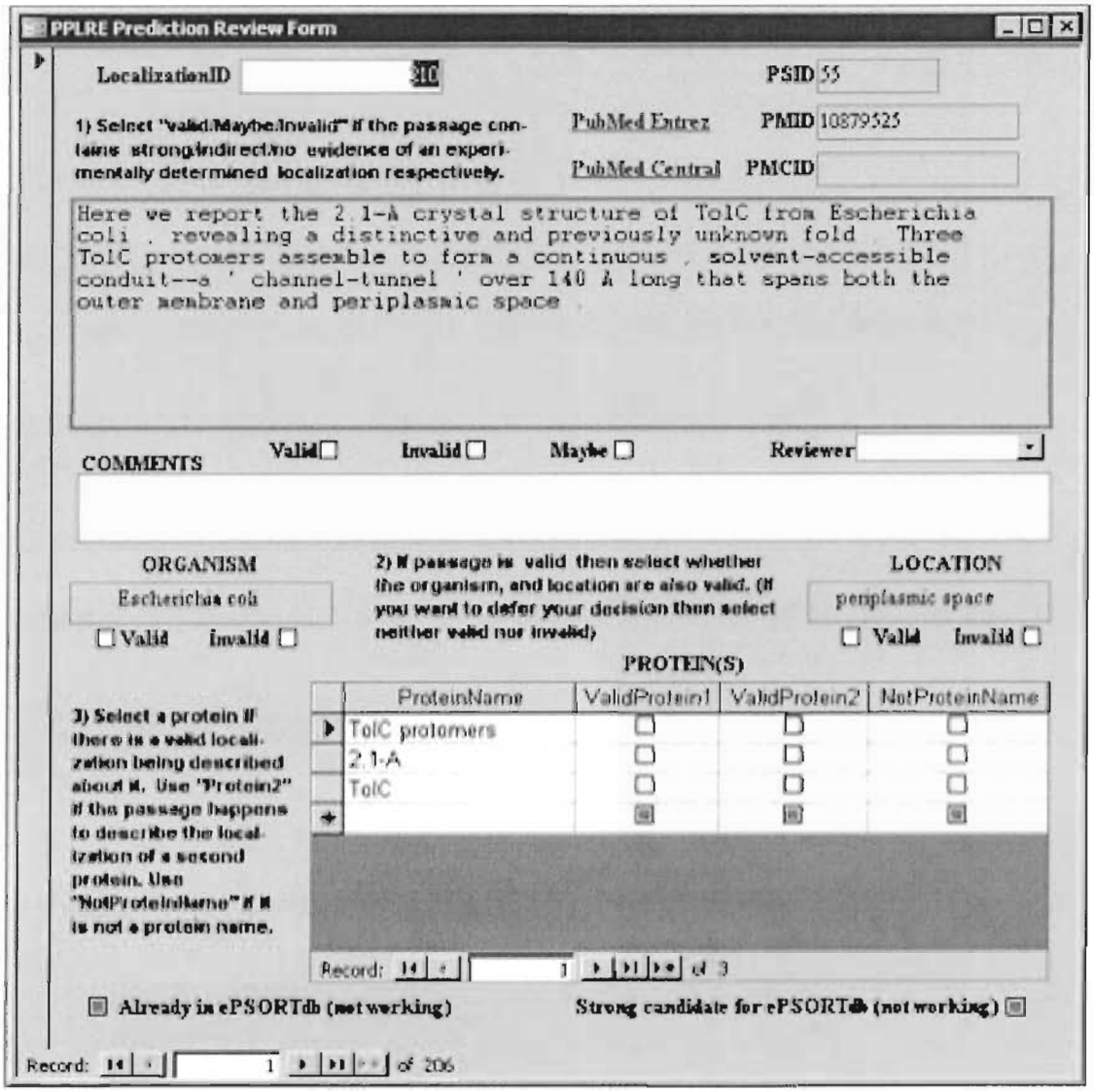

Figure 5.3: The curation interface

from the first set are:

- Each prediction includes a passage with two consecutive sentences.
- A *Maybe* choice is added to the passage relevance metric for the indirect evidence of an experimentally determined localization. Curators label a passage as *Maybe* when the relation was experimentally determined but not explicitly indicated in the example.
- The curators are able to label the false positive and false negative identifications of PROTEIN names.

In the third set $v1.2$ each predication contains multiple sentences which do not have to be consecutive. This set consists of 319 predictions, out of which 110 contain BPL relations. The fourth review set $v2.0$ contains 300 single-sentence predictions, among which 61 predictions are found true. Table 5.2 lists the numbers of BPL positive and negative predictions.

Table 5.2: Numbers of predictions in the curated set

Number of examples	**v1.0**	**v1.1**	**v1.2**	**v2.0**	**Total**
Positive	75	87	110	61	333
Negative	539	62	219	239	1059
Total	614	149	319	300	1392

A BPL prediction may contain multiple BPL relations. For instance, the following example shows two BPL tuples in a single sentence: (*Erwiniaamylovora*, *levansucrase*, *extracellular*) and (*Pseudomonassyringae*, *levansucrase*, *extracellular*). In this case, we duplicate the sentence as two positive examples in the curated set, with different relation annotations, each of which indicates one BPL tuple.

Example 5.1: (*Valid*)

The EPS levan is synthesized by the [$_{LOCATION}$ *extracellular*] *enzyme* [$_{PROTEIN}$ *levansucrase*] *in* [$_{BACTERIUM}$ *Pseudomonas syringae*] , [$_{BACTERIUM}$ *Erwinia amylovora*], *and other bacterial species.*

Data set	**Sentences**	**BPL Instances**	**Relevant NEs**
Positive training set	505	333	768
Positive test set	100	65	160
Negative training set	653	649	0
Negative test set	146	145	0

Table 5.3: Training and test sets: numbers of sentences, BPL instances and relevant NEs

Example 5.2 shows an example of *Maybe* relation, in which *glucan* is a SUGAR not a PROTEIN NE. Example 5.3 is an *invalid* example which does not contain a valid PROTEIN name.

Example 5.2: (*Maybe*)
The [$_{LOCATION}$ *periplasmic*] *cyclic beta-1,2-glucan of* [$_{BACTERIUM}$ *Agrobacterium tumefaciens*] *is believed to maintain high osmolarity in the periplasm during growth of the bacteria on low-osmotic-strength media.*

Example 5.3: (*Invalid*)
Collectively, these data suggest that the C. jejuni Cia proteins are secreted from the flagellar export apparatus.

The positive and negative examples are then split into training and test sets as listed in Table 4.2, which is re-listed in Table 5.3 below, for the purposes of training and evaluating our proposed models. The training set is four times larger than the test set. We use these examples as **standard** training and evaluation sets for models proposed in this report.

5.3 Generative Model

5.3.1 Introduction

Conventionally, parsing techniques are used to find the grammatical structure of a sentence given a formal grammar, while Miller et al. [72] integrate both syntactic and semantic interpretations into the parsing process of a lexicalized probabilistic context-free parser (LPCFG). This generative model not only proves capable of performing

both syntactic and semantic processing, but also limits the propagation of errors by the mutual influence of syntactic and semantic interpretations. The parser is applied to the Template Entities and Template relation tasks of MUC-7 and achieved F-score at 83.49% and 71.23% respectively.

A Link Parser [101] is used to build a syntactic bi-gram model, which represents the relation (or *link*) between two syntactic constituents in the sentence [12]. These bi-grams are then classified (using Naive Bayes and SVMs) based upon the relevance to the gene/protein interactions. Compared to LPCFG, the link parser is able to interpret larger sub-structures of sentences, for instance, subject to verb, verb to object. However, the method proposed by the authors limited itself to a few types of links (e.g, subject and verb, verb and object, noun and its modifier, etc.) and does not involve the semantic information. Some recent work on applying parsing techniques to the information extraction task use dependency tree parsing. Compared to dependency tree, the link parser only produces partly derivable dependencies (the notion of head is essentially missing). In addition, since links are undirected and the notion head is abandoned, the link parser lacks of the central dependency and cannot generate hierarchy of dependencies.

Culotta et al. proposed a kernel method for to the detection and classification of relations between entities [24]. This was done by estimating the similarity between the dependency trees between sentences in the training vs. the test data. Each node of the dependency tree consists of the word and its syntactic and semantic information (e.g., POS, entity type, WordNet hypernyms, semantic role labels, etc.) and matches against others in the kernel similarity function.

Bunescu et al. applied the same kernel method as Culotta and Sorensen's, except that they argued that the relation information is concentrated on the shortest path in the dependency tree [11]. They extracted the path between two entities along predicates and arguments in the dependency tree and enriched the path by the syntactic and semantic information of the passed-through nodes. The shortest path is therefore an even smaller representation of the dependency tree than the smallest common subtree of two entities, thus more efficient as a tree kernel approach.

All methods introduced above are seeking binary relations, which is generally

less challenging than the ternary BPL relation. In addition, most of them work on the newswire, which is generally much less dependent on domain knowledge than biomedical articles. We introduce a parsing-based generative approach that integrates syntactic parsing, entity type, WordNet ontological annotation and domain-specific information into the parse tree and trains a semantic parser with MEDLINE articles curated by domain experts. In addition, we propose the following methods to minimize the problems of the generative model introduced in Section 6.2.1.

- **Sparse data problem**. The idea is to double the training set size by replacing protein, organism and location names with their NE tags. The other advantage of this method is that we can make use of results of the NE recognizer in the test phase, in order to largely increase accuracy of protein name identification.

- **Very local relation patterns**. After the prediction phase, we apply a discriminative model on features of the parse trees (mainly from links among entities) to capture more global patterns.

- In order to take the existing annotations as the additional information instead of constraints during the prediction phase, each prediction of the trained generative model will be slightly adjusted by its agreement with the existing annotations. For instance, if both the trained model and NER agree that "IL-2" is a PROTEIN, the prediction probability will increase, and otherwise will decrease.

In this section we propose a statistical parsing technique that simultaneously identifies biomedical named-entities (NEs) and extracts subcellular localization relations for bacterial proteins from the text in MEDLINE articles. We build a parser that derives both syntactic and domain-dependent semantic information and achieves an F-score of 16.7% for the BPL extraction. This performance is generally not acceptable, since half of BPLs predicted by this parser is incorrect (the precision is only 50%) and only 10% of actual BPLs are extracted. We propose a semi-supervised approach that incorporates automatically labeled data with noise to improve the F-score of our parser to 40.5%.

Our goal is to automatically extract BPL predictions, which will then be reviewed by human experts and populated into the relation database. As a result, precision of the predictions is more important than recall in order to save human efforts, providing a reasonably high recall. Therefore, the semi-supervised approach greatly improves the overall performance of the BPL extraction. The precision is significantly increased to from 50% to 88.9%, while the recall is improved to 26.2%.

Evaluation metrics are described in Chapter 3. Since the parser is trained on the positive examples only, it would be reasonable to evaluate it on positive test examples only. In this Chapter, all proposed parsing-based methods are evaluated against positive examples only, in order to compare them with each other. The best performing method is then also evaluated against all test examples.

5.3.2 Description of the Statistical Parser

Similar to the approach in [72] and [54], our parser integrates both syntactic and semantic annotations into a single annotation as shown in Figure 5.4. A lexicalized statistical parser [5] is applied to the parsing task. The parse tree is augmented by two types of semantic annotations:

1 Annotations on relevant PROTEIN, BACTERIUM and LOCATION NEs. Tags are *PROTEIN_R*, *BACTERIUM_R* and *LOCATION_R* respectively.

2 Annotations on paths between relevant NEs. The lower-most node that spans both NEs is tagged as *_LNK* and all nodes along the path to the NEs are tagged as *_PTR*.

5.3.3 Ternary vs. Binary relation

Since binary relations are more feasible to represent on the parse tree, the BPL relation is split into two binary relations:

- BP: BACTERIUM and PROTEIN
- PL: PROTEIN and LOCATION

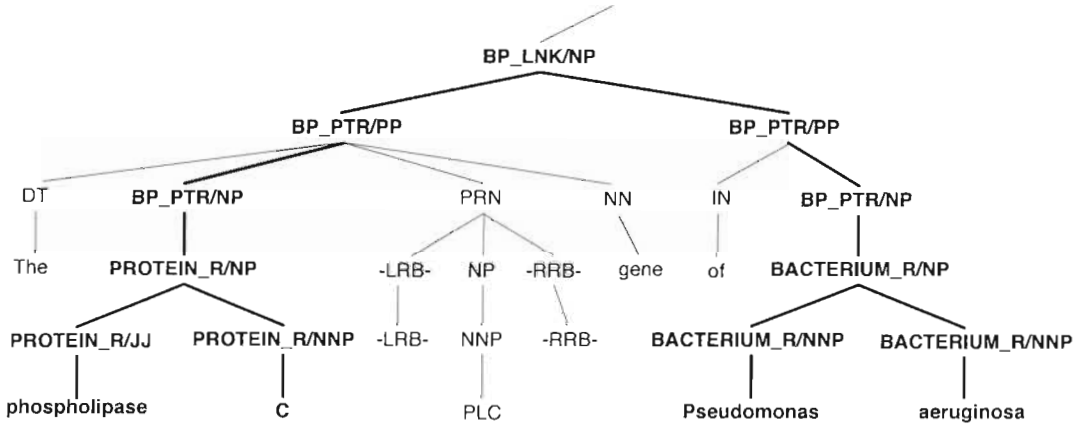

Figure 5.4: An example of parsing results

The BACTERIUM-LOCATION relation is ignored since it is given knowledge as illustrated in Figure 1.2. This dual-binary-relation approach also makes it easier to capture BPLs across sentences (i.e., a BPL occurs in multiple sentences), since in these cases it is more likely to see two names (e.g., only protein and prokaryote names) instead of all three in a single sentence.

The BPL relation can be predicted by a fusion of BP and PL once they are extracted. The fusion of PL and BP relations is a process to generate the target BPL relation from already extracted PL and BP relations. A PL and a BP relation will be combined if they appear in the same abstract and both PROTEIN names refer to the same protein.

For instance (PMID: 15868041):

Example 5.4: [*PROTEIN KatB*] *was localized to the* [*LOCATION cytoplasm*]*, while KatA, the "housekeeping" enzyme, was detected in both cytoplasmic and periplasmic extracts. A* [*BACTERIUM P. aeruginosa*] [*PROTEIN katB*] *mutant demonstrated 50% greater sensitivity to hydrogen peroxide than wild-type bacteria, suggesting that KatB is essential for optimal resistance of P. aeroginosa to exogenous hydrogen peroxide.*

Suppose a BP relation between *KtaB* and *cytoplasm* is identified from the first sentence and a PL relation between *KtaB* and *P. aeruginosa* from the second sentence. These

two binary relations would imply a *BPL(KtaB, P. aeruginosa, cytoplasm)*.

One problem with this "dual-binary-relation" approach is that when a protein associates with more than one organism, we may end up with the wrong BACTERIUM-LOCATION relation. An analysis on how the problem affects the overall performance will be discussed later.

5.3.4 Recovery of Incomplete Parse

The initiative of predicting binary relations is straightforward: a PROTEIN and a BACTERIUM (or LOCATION) NEs compose a BP (or PL) relation if all nodes on the path between them are annotated with _PTR labels. However, a sentence with relation may not be correctly parsed but we still wish to find the right relation from its parse. We apply two techniques in a row to make predictions from incomplete parses: **Relation Recovery** and **NE Recovery**. With Relation Recovery, any nodes on the path between PROTEIN and BACTERIUM (or LOCATION) NEs are annotated as BP(or PL)_PTR if they are not. With the NE Recovery, an NP is annotated as an NE (i.e., all non-leaf nodes under the NP are annotated with NE tags) if:

- any of its descendent nodes is annotated with the NE tag, or
- it is located at one end of a path that, except for this NP, is fully annotated with NE and relation tags.

For example, Figure 5.5 shows the actual parsing results of the sentence in Figure 5.4. Three nodes in circles are incorrectly annotated: 1) the relation tag is missing on PP; 2) "C" is not predicted as part of a PROTEIN NE; 3) "gene" is mistakenly identified as a PROTEIN NE. We attempt to recover the incomplete parse tree in the following steps corresponding to the two techniques introduced above.

Firstly, with the relation recovery, the node PP in the circle is added with the relation tag BP_PTR. Now two BP relations can be predicted from this example: BP_1*(phospholipase, Pseudomonas aeruginosa)* and BP_2*(gene, Pseudomonas aeruginosa)*, as linked by dashed lines in Figure 5.5. Secondly, with the NE recovery, we add the PROTEIN NE tag to the word "C", since "phospholipase" is annotated with the NE tag by the parser, so is the NP "phospholipase C". In addition, the word

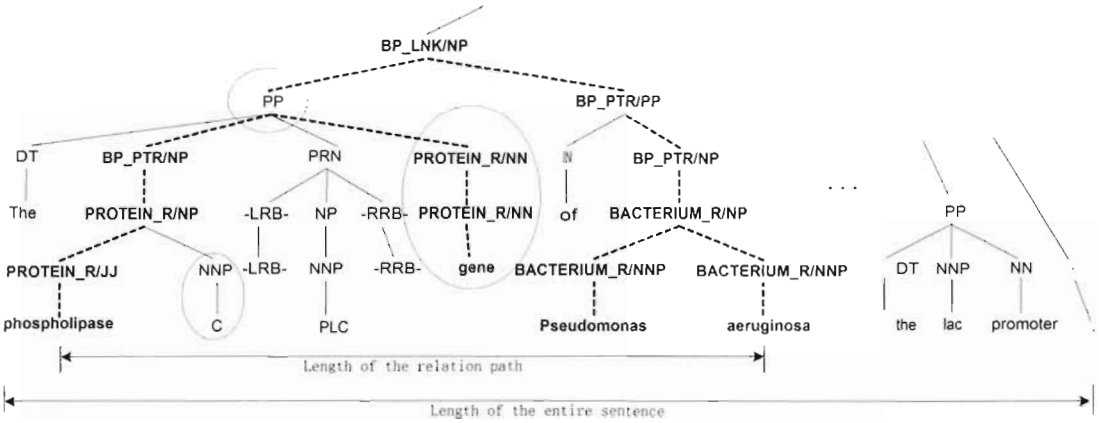

Figure 5.5: An example of parsing results

"gene" is in our hand-built stop-word list of PROTEIN NEs and therefore the relation BP_2 is then removed. The final relation prediction we make from this example is *BP(phospholipase C, Pseudomonas aeruginosa)*.

5.3.5 Confidence of Relation Prediction

Bikel's parser produces a log probabilistic confidence score c_T for each parse T. However, c_1 does not measure the confidence of a relation prediction, which only covers a sub-tree of the entire parse tree. Instead of modifying Bikel's parser to produce the log probability of any sub-tree, we approximately assign a confidence score of a sub-tree t from c_T:

$$c_t = c_T + log\frac{l(t)}{l(T)} \tag{5.1}$$

where $l(t)$ and $l(T)$ denote the number of words covered by the sub-tree t and the entire tree T respectively, as illustrated in Figure 5.5.

In addition, a penalty is applied to c_t of relations recovered by the techniques introduced in Section 5.3.4. A penalty coefficient p_t is defined as:

$$p_t = \frac{n_{tags}}{n_{path}} \tag{5.2}$$

Data set	Sentences	BPL Instances	Relevant NEs
Positive training set	505	333	768
Positive test set	100	65	160
Negative training set	653	649	0
Negative test set	146	145	0

Table 5.4: Training and test sets: numbers of sentences, BPL instances and relevant NEs

where n_{path} denotes the number of nodes on the path, and n_{tags} the number of nodes being annotated with NE or relation tags by the parser (before any recovery is applied) along the path. For instance, the relation BP_1*(phospholipase C, Pseudomonas aeruginosa)* in Figure 5.5, $n_{tags} = 9$ and $n_{path} = 11$. The new definition of c_t now is:

$$c_t = c_T + log\frac{L(t) \times n_{tags}}{L(T) \times n_{path}} \tag{5.3}$$

When fusing two binary relations (from the same sentence with the same PROTEIN NE) into a BPL relation, we add confidence scores of BP and PL as the confidence score of BPL. Our experiments show that the confidence score of the majority of predicted BPL relations is between $(-3.0, -1.0)$. An observation indicates that any two binary relations containing the same PROTEIN NEs from the same sentence have great chance to form a valid BPL relation. Therefore, we set a threshold -2.5 on the confidence score of BPL predictions, such that any BPL predictions with confidence score less than -2.5 will be ignored.

5.3.6 Extraction Using Supervised Parsing

We first experiment with a fully supervised approach by training the parser on the BP/PL training set and evaluating the parser on the test set. Table 4.2 that summarizes the training and test sets is listed again in Table 5.4 for an easy reference.

Evaluation results in Table 5.5 show low precision and recall on binary predictions. When combining binary relations, precision on ternary predictions increases but recall drops.

Method	**Performance** (Precision/Recall/F-score)(%)		
	BP	**PL**	**BPL**
Baseline 1	27.1/56.5/36.6	338.9/69.0/49.8	14.1/61.5/23.0
Supervised (curated data only)	15.8/10.3/12.5	37.9/25.0/30.1	50.0/10.0/16.7
Semi-Supervised (curated data + newswire)	8.7/7.1/7.8	28.0/15.5/20.0	16.7/3.3/5.5
Semi-Supervised (noisy data only)	86.7/46.4/60.5	69.2/40.9/51.4	83.3/16.7/27.8
Semi-Supervised (curated data + noisy data)	85.7/66.7/**75.0**	73.3/50.0/**59.5**	88.9/26.2/**40.5**

Table 5.5: Evaluation results of supervised and semi-supervised parsing-based methods. The training data is described in Table 4.2.

Looking into the results of the parser, we find that the quality of the syntactic annotations is poor. Figure 5.6 shows the parse tree of the sentence in Example 5.5 generated by the supervised parser, in which some major syntactic constituent dependencies are incorrect as highlighted by dashed regions. The correct dependencies are indicated by dashed arrows. Moreover, recall of the PROTEIN NER is only 13.5% due to 1) too few PROTEIN NEs in the training set; 2) not using available protein name sources.

Example 5.5: *We now show that* [$_{PROTEIN}$ *pagP*] *and its* [$_{BACTERIUM}$ *Escherichia coli*] *homolog (crcA) encodes an unusual enzyme of lipidA biosynthesis localized in the* [$_{LOCATION}$ *outer membrane*].

We also find that most paths between PROTEIN and BACTERIUM/LOCATION NEs are not complete, as shown in Figure 5.6, due to incorrectly identified and missed NEs. The lack of syntactic and semantic information in the training set is also one of reasons that result in incomplete paths.

In summary, a major reason for above observed problems is the lack of training data, which as we said earlier is a common problem in the bio-NLP area.

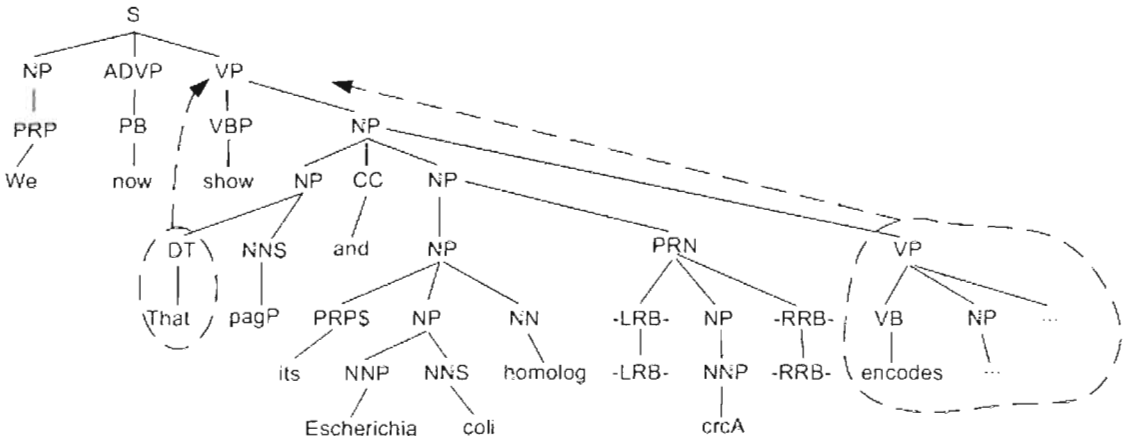

Figure 5.6: An example of parsing results from the supervised parser. The parse tree includes both syntactic and semantic (NEs and relations) annotations.

5.3.7 Extraction Using Semi-supervised Learning

Training Set Expansion with Newswire Data

In order to improve syntactic parsing performance, we included the Penn Treebank corpus[4] into our training set. The Penn Treebank contains nearly 1 million syntactically parsed sentences. Evaluation results are shown in Table 5.5.

We observed a few typical sentences and found the accuracy of syntactic annotations is highly improved. For instance, the sentence in Example 4.5 is correctly parsed in terms of syntactic dependencies among constituents, as shown in Figure 5.7. However, the overall performance is significantly worse than the supervised system. The reason is that the feature distributions of PROTEIN, ORGANISM and LOCATION names significantly decrease due to the large size of non-biomedical articles.

Training set expansion with noisy data

Experiments with purely supervised learning show that our generative model requires a large curated set to minimize the sparse data problem, but domain-specific annotated corpora are generally rare and expensive. However, there is a huge source of

[4]The Penn Treebank Project, http://www.cis.upenn.edu/ treebank/.

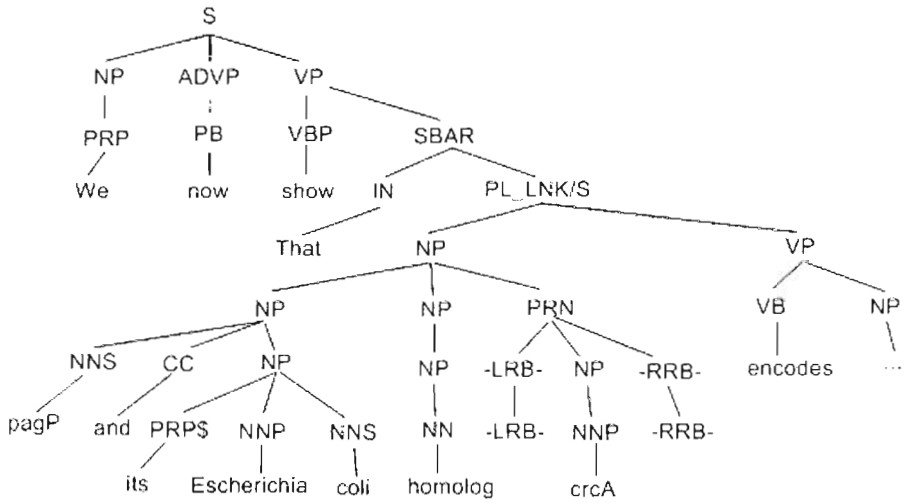

Figure 5.7: An example of parsing results by the supervised parser with additional newswire training examples. Note that the parse tree fails to include any semantic (NEs and relations) annotations.

unlabeled MEDLINE articles available that may meet our needs, by assuming that any sentence containing BACTERIUM, PROTEIN and LOCATION NEs has the BPL relation. We then choose 14,008 such sentences from a subset of the MEDLINE database as the training data. These sentences, after being parsed and BPL relations inserted, are in fact the very noisy data when used to train the parser, since the assumed relations do not necessarily exist. The reason this noisy data works at all is probably because we can learn a preference for structural relations between entities that are close to each other in the sentence, and thus distinguish between competing relations in the same sentence.

Two experiments were carried out corresponding to choices of the training set: 1) noisy data only, 2) noisy data and curated training data.

Evaluation results in Table 5.5 show that, compared to supervised parsing, our semi-supervised methods dramatically improve precision and recall for both binary and ternary predictions. For ternary predictions of the semi-supervised method trained on the curated data and noisy data, recall increases from 10% to 26.7% and

precision increases from 50.0% to 88.9%. Evaluation results suggest that BP predictions benefit more from the semi-supervised learning. They also show that the inclusion of curated data in the semi-supervised method also improves the overall performance.

We also experimented with training the semi-supervised method using noisy data alone, and testing on the entire curated set, i.e., 333 and 286 sentences for BP and PL extractions respectively. Note that we do not directly train from the training set in this method, so it is still "unseen" data for this model. The F-score of ternary predictions is 25.1%.

We name the best-performing parser, the one trained on noisy data and curated training data, **ZParser**, for easy reference later on.

As we discussed before, evaluation results given above are based on positive examples only. We also test the best-performing parser, ZParser, against all test examples (see Table 4.2). BPL prediction results of ZParser and two baseline systems introduced in Section 4.4: NE co-occurrence and Snowball, are listed in Table 5.6.

	NE co-occurrence	**Snowball**	**ZParser**
Precision (%)	5.9	**66.6**	58.6
Recall (%)	**61.5**	18.2	26.2
F-score (%)	10.8	28.6	**36.2**

Table 5.6: Evaluation results on BPL predictions of the NE-co-occurrence baseline system, Snowball and the best-performing ZParser against all test examples

The NE co-occurrence baseline takes BACTERIUM-PROTEIN-LOCATION tuples that are identified by the automatic NER as BPL candidates and therefore achieves the highest recall among all three systems listed in Table 5.6. Reasons that this baseline does not receive 100% recall are: 1) the performance of NER, especially PROTEIN NER, is not significantly good; 2) not all true NEs are relevant to BPL relations; 3) the system cannot extract BPLs across multiple sentences. Due to the large number of FPs, the NE co-occurrence system of course gets very low precision and accuracy, which make its F-score the lowest among all three systems.

Snowball seems very careful when making predictions, thus obtaining highest precision but lowest recall. Our best-performing ZParser achieves the best F-score among all the systems. Its precision is close to that of Snowball, while the recall outperforms Snowball by 7.6%.

To assess the statistical significance of the improvements achieved by ZParser, we also perform a two-tailed significance test on the results of Baseline 1 and ZParser, using an implementation of a computer-intensive, stratified approximate-randomization test [120]. The significance test on positive and negative test data shows that the improvements on F-score are statistically significant with p value of 8×10^{-4}.

5.3.8 Bio-NER Shared Task

The NER module of the Annotator consists of existing NER tools (Lingpipe and MMTx) and dictionary-lookup from UMLS, NCBI Taxonomy, SwissProt and GO. However, it introduces many false NEs thus performing poorly on precision. Table 5.7 lists types and numbers of enties from each dictionary source.

Dictionary Sources	UMLS	NCBI	SwissProt	GO
Number of protein names	434,324	-	139,412	-
Number of bacterium names	94,020	384,915	-	-
Number of location names	-	-	-	75

Table 5.7: Types and numbers of entries from dictionary sources

The parsing-based relation extraction method introduced in this chapter actually identifies NEs as the same time, since the semantic parser implemented to identify NEs produces a parse tree with PROTEIN NE annotations. Therefore, it would be capable of improving the performance of the current NER module.

To the initial training and test sets (2,000 and 404 abstracts respectively) are from the COLING/BIONLP shared task[5].

A few experiments on the parsing-based NER with different methods were carried out as listed below. Evaluation results are in a form of Recall/Precision/F-Score of

[5]http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/ERtask/report.html

full name matching. The best performing system in the shared task are: 69.01% / 79.24% / 73.77%. F-score of the worst system is around 48%.

1 Baseline. Dictionary-lookup only: 54.31% / 31.65% / 40.00%. The dictionary was collected from UMLS, SwissProt and iProClass.

2 Fully Supervised learning. The training set was from BIONLP only. 61.52% / 61.93% / 61.72%.

3 Method 1 + 2. The training set: 1) the BIONLP training set + 2) the BIONLP test set with dictionary-lookup. 57.89% / 42.13% / 48.77%.

Method 4 and 5 train the parser with the additional noisy data as introduced in Section 5.3.7 :

4 The training set consists of 1) the BIONLP training set, 2) the noisy data set. 63.58% / 47.7% / 54.51%.

5 Same as Method 4, except multiplying the BIONLP training set 5 times to "emphasize" the human-curated examples. 63.76% / 51.29% / 56.85%.

The methods introduced above take advantage of BPL relations to find PROTEIN NEs. However, other types of NEs may also indicate occurrences of PROTEIN NEs. Method 6 and 7 build paths between PROTEIN and another NEs (CELL_TYPE and DNA respectively).

6 Based on Method 2, we build additioned paths between PROTEIN NEs and CELL_TYPE NEs on the parse tree in the training phase. 63.54% / 60.44% / 61.95%.

7 Same as Method 6, except that we build paths by linking DNA NEs and PROTEIN NEs. 62.41% / 60.12% / 61.24%. (CELL_TYPE and DNA are largest NE types after PROTEIN in the BIONLP training set)

The dictionary-lookup results in both low precision and recall. One of the reasons is that NE annotations in the BIONLP sets are not at all consistent. For instance, "cytokine" appears 98 times in the training set, while only 55 of them are annotated as PROTEIN NEs. This word-sense disambiguation problem has to be solved by looking at the contexts in which the NEs occur.

Furthermore, noisy data slightly improves recall but largely decreases precision. It seems that the BIONLP data sets are more sensitive to the noisy data than our own corpus, on which our previous experiments with noisy data have shown a great improvement on NER.

Finally, other types of NEs do not provide much help in recognizing protein names. The reason is that we do not have a meaningful relation between these NE types from the molecular biology point of view, and this is also why we succeeded with our own corpus - with the BPL relation.

5.3.9 Discussion

Methods and experiments described in this section have already shown promising and exciting results of the proposed semantic parsing approach on BPL relation extraction. At this stage each BPL relation is extracted from a single document.

In this chapter we introduce a statistical parsing-based method to extract biomedical relations from MEDLINE articles. We make use of a large unlabeled data set to train our relation extraction model. Experiments show that the semi-supervised method significantly outperforms the fully supervised method with F-score increasing from 16.7% to 40.5%.

However, generative models are generally limited by the sparse data problem due to the significant number of zero occurrences of joint events. In addition, since ZParser is built from very local contexts: head constituents, modifiers and parent nodes, it may not be able to capture more complex relation patterns. Discriminative models, on the other hand, are not limited by local characteristics, but usually have error propagation problems. In the next chapter, we will introduce a discriminative model [65] which takes as input the examples with gold named entities and identifies BPL relations on them. A combination of the generative model and the discriminative model, which

attempts to make use of advantages of both models, is also implemented to further refine the relation extraction results.

Chapter 6

Discriminative and Hybrid Models

6.1 Discriminative Model

In this chapter, we propose a ternary relation extraction method[1] primarily based on rich syntactic information. We extract BACTERIUM-PROTEIN-LOCATION (BPL) relations from the text of biomedical articles. Different kernel functions are used with an SVM learner to integrate two sources of information from syntactic parse trees: (i) specific syntactic features extracted along the path between entities, and (ii) features from entire trees using a tree kernel. We use the large number of syntactic features that have been shown to be useful for Semantic Role Labeling (SRL) and apply them to the relation extraction task. Our experiments show that the use of rich syntactic features outperforms shallow word-based features.

6.1.1 Introduction

Previous work in the biomedical relation extraction task [97, 6, 32] suggested the use of predicate-argument structure by taking verbs as the center of relation expressions[2]. In contrast, in this chapter we directly link PROTEIN NEs to their locations. Claudio et al. [17] proposed an approach that solely considers the shallow semantic features extracted from sentences.

[1]This is a joint work with Yudong Liu at the Natural Language Processing Laboratory, Simon Fraser University.

[2]There are several other papers that exploit predicate-argument structure for relation extraction, but since our approach is substantially different and due to lack of space, we do not cite them all here.

For relation extraction in the newswire domain, syntactic features have been used in both a generative model [71] and a discriminative log-linear model [49]. In contrast, we use a much larger set of syntactic features extracted from parse trees, many of which have been shown useful in SRL.

Kernel-based methods have been used for relation extraction on various syntactic representations. Culotta and Sorensen applied a kernel method to the detection and classification of relationships between entities [24], by estimating the similarity between the dependency trees of sentences. Each node of the dependency tree consists of the word and its syntactic and semantic information (e.g., POS, entity type, WordNet hypernyms, semantic role labels, etc.) and matches against others in the kernel similarity function. Bunescu and Mooney applied the same kernel method as Culotta and Sorensen's, except that they argued that the relation information is concentrated on the shortest path in the dependency tree [11]. They extracted the path between two entities along predicates and arguments in the dependency tree and enriched the path with the syntactic and semantic information of the passed-through nodes. The shortest path is therefore an even smaller representation of the dependency tree than the smallest common subtree of two entities, thus more efficient as a tree kernel approach.

In contrast we explore a much wider variety of syntactic features in this work. To benefit from both views a composite kernel [122] integrates the flat features from entities and structured features from parse trees. In our work, we also combine a linear kernel with a tree kernel for improved performance.

Our proposed discriminative models that use rich syntactic features will also be compared with bag-of-word approaches. Nair and Rost build a subcellular classifier on keywords of functional annotations of proteins in the SWISS-PROT database [76]. They first retrieve keywords from the protein annotations and then map each protein annotation onto a keyword-based vector space. Stapley et al. represent yeast proteins as vectors of weighted terms from Medline documents mentioning their respective genes [105]. The term weights of a vector are functions of their frequencies within the document collection as a whole and the frequency within the relevant documents. SVMs are applied in both papers as the classifier using bag-of-word features.

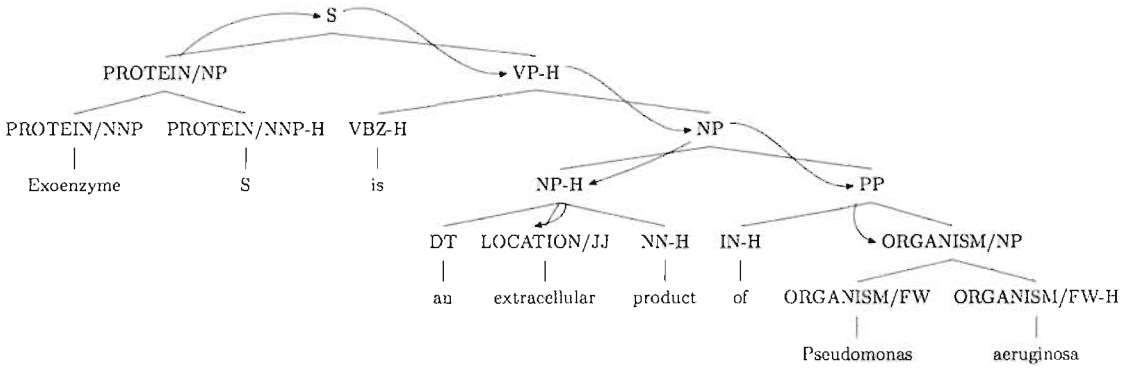

Figure 6.1: An example of BPL ternary relation in a parse tree

To compare with the proposed discriminative models, we implement a bag-of-word model, the **Baseline3**. Experiments in Section 6.1.4 will show that models using rich syntactic features outperform word-based models.

6.1.2 SRL Features for Information Extraction

Figure 6.1 shows one example illustrating the ternary relation we are identifying. In this example, "Exoenzyme S" is a PROTEIN NE, "extracellular" a LOCATION NE and "Pseudomonas aeruginosa" a BACTERIUM NE. Again our task is to identify whether there exists a BPL relation among these three NEs.

To simplify the problem, as was done in Chapter 5, we first reduce the BPL ternary relation extraction problem into two binary relation extraction problems. Specifically, we split the BPL ternary relation as follows:

- BP: PROTEIN and BACTERIUM
- PL: PROTEIN and LOCATION

Notice that the BACTERIUM-LOCATION relation is ignored because it is irrelevant to PROTEIN and less meaningful than BP and PL relations. Based on this simplification, and following the idea of semantic role labeling, we take the PROTEIN NE in the role of the predicate (verb) and the BACTERIUM/LOCATION NE as its argument candidates in question. Then the problem of identifying the binary relations of BP and PL has been reduced to the problem of argument classification given the

predicate and the argument candidates. The reasons that we pick PROTEIN NEs as predicates are:

- We assume PROTEIN NEs play a more central role in linking the binary relations to the final ternary relations.
- Neither BACTERIUM nor LOCATION NE is appropriate for being predicate, since the relation BACTERIUM-LOCATION is known and not our interest.

Compared to a corpus for the standard SRL task, there are some differences: first is the relative position of PROTEIN and BACTERIUM/LOCATION NEs. Unlike the case in SRL, where arguments locate either before or after the predicate, in this application it is possible that one NE is embedded in another. A second difference is that a predicate in SRL scenario typically consists of only one word; here a PROTEIN NE can contain up to 8 words.

Note that we do not use the PropBank data set in our model at all, since it is not a biomedical corpus. All of our training data and test data is annotated by domain expert biologists and parsed by Charniak-Johnson's parser [14]. When there is a misalignment between the NE and the constituent in the parse tree, we insert a new NP parent node for the NE.

6.1.3 System Description

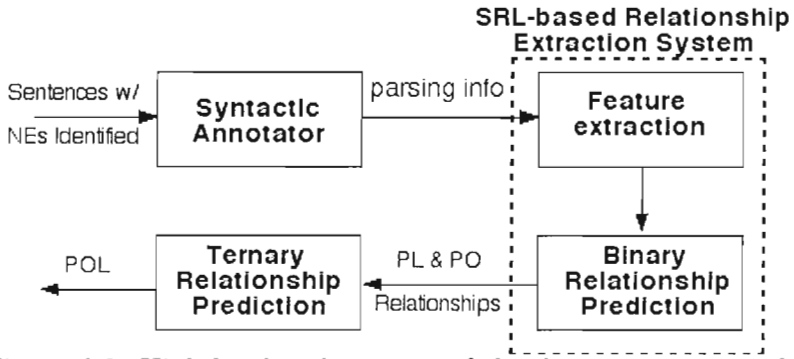

Figure 6.2: High-level architecture of the discriminative model

• each word and its Part-of-Speech (BPS) tag of the PRO NE
• head word (hw) and its BPS of the PRO NE
• subcategorization that records the immediate structure that expands from the PRO NE. Non-PRO daughters will be eliminated
• BPS of parent node of the PRO NE
• hw and its BPS of the parent node of the PRO NE
• each word and its BPS of the ORG NE (in the case of "BP " relation extraction).
• hw and its BPS of the ORG NE
• BPS of parent node of the ORG NE
• hw and its BPS of the parent node of the ORG NE
• BPS of the word immediately before/after the ORG NE
• punctuation immediately before/after the ORG NE
• feature combinations: hw of PRO NE_hw of ORG NE, hw of PRO NE_BPS of hw of ORG NE, BPS of hw of PRO NE_BPS of hw of ORG NE
• path from PRO NE to ORG NE and the length of the path
• trigrams of the path. We consider up to 9 trigrams
• lowest common ancestor node of the PRO NE and the ORG NE along the path
• LCA (Lowest Common Ancestor) path that is from the ORG NE to its lowest common ancestor with PRO NE
• relative position of PRO NE and ORG NE. In parse trees, we consider 4 types of positions that ORGs are relative to PROs: before, after, inside, other
• LTAG-based features along the path from PRO NE to ORG NE*

Table 6.1: Features adopted from the SRL task. PRO: PROTEIN; ORG: BACTERIUM

Figure 6.2 shows an overview of the discriminative model. The input to our system consists of titles and abstracts that are extracted from MEDLINE records. These extracted sentences have been annotated with the NE information (PROTEIN, BACTERIUM and LOCATION). The syntactic Annotator inserts the head information into the parse trees by using the Magerman/Collins head percolation rules. The main component of the system is our SRL-based relation extraction module, where we first manually extract features along the path from the PROTEIN NE to the BACTERIUM/LOCATION NE and then train a binary SVM classifier for the binary relation extraction. Finally, we combine the extracted binary relations to the ternary relations, the same process as introduced in Section 5.3.3.

• subcategorization that records the immediate structure that expands from the ORG NE. Non-ORG daughters will be eliminated
• BPS of the word immediately before/after the PRO NE*
• punctuation immediately before/after the PRO NE*
• if there is an VP node along the path as ancestor of the PRO NE
• if there is an VP node as sibling of the PRO NE
• if there is an VP node along the path as ancestor of the ORG NE*
• if there is an VP node as sibling of the ORG NE*
• path from PRO NE to LCA and the path length (L1)
• path from ORG NE to LCA and the path length (L2)
• combination of L1 and L2
• sibling relation of PRO and ORG
• unigrams between PRO NE and ORG NE; stop words are selectively filtered.
• bigrams between PRO NE and ORG NE*
• distance between PRO NE and ORG NE in the sentence. (3 valued: 0 if nw (number of words) = 0; 1 if 0 < nw <= 5; 2 if nw > 5)
• combination of distance and sibling relation

Table 6.2: New features used in the SRL-based relation extraction system.

As a discriminative feature-based relation extraction system, identification of important features is crucial to the task. After a series of feature calibrations, we proposed features for BP/PL relation extraction that are listed in Table 6.1 and Table 6.2. Features marked with an asterisk are used for BP but not for PL relation extraction. Features with no mark are used for both. **LTAG** (Lexicalized Tree-Adjoining Grammar) based features are extracted in the same way as described in [64].

6.1.4 Experiments and Evaluation

Experimental Results

The data set used to train and evaluate proposed models was shown in 4.2, which is re-listed here in Table 6.3. To ensure significance of our results, we do 5-fold cross validation[3] in all our experiments.

[3]While some researchers use 10-fold cross validation, we have sufficient data for 5-fold cross validation, an evaluation scheme which is often used by other researchers in the field [76, 21].

Data set	**Sentences**	**BPL Instances**	**Relevant NEs**
Positive training set	505	333	768
Positive test set	100	65	160
Negative training set	653	649	0
Negative test set	146	145	0

Table 6.3: Training and test sets: numbers of sentences, BPL instances and relevant NEs

Method	**PL**			**BP**			**BPL**		
	Prec	**Rec**	**F**	**Prec**	**Rec**	**F**	**Prec**	**Rec**	**F**
Baseline1	9.6	69.0	16.9	5.0	56.5	9.2	5.9	61.5	10.8
Baseline3	62.4	61.0	61.7	57.9	59.7	58.8	57.1	67.1	61.7
PAK	71.0	52.4	60.3	69.0	49.2	57.5	66.0	51.6	57.9
SRL	72.9	56.9	63.9	66.0	52.4	58.4	70.6	49.8	58.4
TRK	69.8	60.2	64.7	64.2	62.1	63.1	79.6	48.9	60.5
TRK+SRL	74.9	58.6	**65.8**	73.9	57.6	**64.8**	75.3	55.0	**63.6**

Table 6.4: Percent scores of Precision/Recall/F-score for PL, BP and BPL relation predictions.

We built several models to compare the relative utility of the various types of rich syntactic features we can exploit for relation extraction. For various representations, such as feature vectors, trees and their combinations, we applied different kernels in a Support Vector Machine (SVM) learner [21]. Specifically, we use Joachims' SVM_light[4] with default linear kernel to feature vectors and Moschetti's SVM-light-TK-1.2[5] with the default tree kernel. The models are Baseline1 and Baseline3 as introduced in Section 4.4. Baseline2 is Snowball, an established data-mining system.
Baseline1 is a naive approach that assumes that any example containing PROTEIN, LOCATION NEs has the PL relation. The same assumption is made for BP and BPL relations.
Baseline3 is a purely word-based system, where the features consist of the unigrams and bigrams between the PROTEIN NE and the BACTERIUM/LOCATION NEs

[4] *http://svmlight.joachims.org/*
[5] *http://ai-nlp.info.uniroma2.it/moschitti/TK1.2-software/Tree-Kernel.htm*

inclusively.

The **PAK** system uses the predicate-argument structure kernel (PAK) based method. PAK was defined in [75] and only considers the path from the predicate to the target argument, which in our setting is the path from the PROTEIN to the BACTERIUM or LOCATION NEs.[6]

The **SRL** is an SRL system which is adapted to use our new feature set. A default linear kernel is applied with SVM learning.

The **TRK** system is similar to PAK system except that the input trees are entire parse trees instead of PAK paths.

The **TRK+SRL** combines full parse trees and manually extracted features and uses the kernel combination.

The SVM_light produces a **Confidence Score** for each relation prediction. We take relations with a confidence score larger than 0 as positive predictions, otherwise as negative predictions.

We name the best-performing discriminative system, TRK+SRL, as **YSRL** for easy reference.

6.1.5 Discussion

Table 6.4 shows the results we obtained when running on our data set with 5-fold cross validation. We evaluated the system performance for ternary relation extraction as well as the extraction of the two binary relations. The total accuracy of finding the correct ternary relation in the test data using rich syntactic features is 63.6% and we can see that it outperforms shallow word-based features, which obtains 61.7% accuracy.[7]

By comparing the **PAK** model and **SRL** model, we observe that with the same path, a system based on manually extracted features significantly boosts precision and accuracy and therefore obtains a significantly better overall performance than

[6]We also experimented with **PAK+SRL** but since the results were similar to using only **SRL**, we do not discuss it here.

[7]We highlight precision and accuracy of finding the ternary relations in text, i.e. distinguishing between positive and negative examples of relations as these are the most important figures for the domain expert biologists.

the **PAK** model. In contrast to the baseline systems, the **TRK** model obtains the highest precision but lower recall on ternary predictions. This means that the sub-structures considered by **TRK** are good in discriminating the instances but may not be necessary. In YSRL, the addition of SRL features boosts the overall performance of **TRK** system to the best overall F-score by moderating precision and recall.

The gaps between models reinforce the fact that the path between NEs in the parse tree is very important for the relation extraction task. In particular, it illustrates that along this path, SRL-based syntactic features are discriminative as well as necessary for this task; In addition, our experiments showed that some features outside this path can contribute to the task to some extent. In **Baseline1** all examples in the test set are predicted to have the BPL relation and thus the Recall is always 73.8% (the remaining 26.2% are across multiple sentences). However, negative examples in our test set normally contain all NEs annotated by the automatic annotator, thus causing from a few to tens of false predictions, which result in very low precision and F-score.

We also attempted to train YSRL with noisy data we trained, but its performance dramatically decreased. This experiment indicates that YSRL does not work as well as ZParser does with noisy data.

6.1.6 Conclusion

In this section we explored the use of rich syntactic features for the relation extraction task. In contrast with the previously used small set of syntactic features for this task, we use a large number of features originally proposed for the SRL task. We provide comprehensive experiments using many different models that exploit syntactic information. The total accuracy of finding the correct ternary relation in the test data using rich syntactic features is 63.6%, and we can see that it outperforms shallow word-based features which obtains 61.7% accuracy.

We built a BPL ternary relation extraction system primarily by exploiting rich syntactic information from parse trees. In particular, we extracted features mainly based on the SRL framework. We also proposed some new features to address the peculiarities of this particular application. For comparison purposes, we built up 6 different systems and applied different kernels in conjunction with SVM learners.

Experiments show that these manually-extracted features under SRL framework play an important role in discriminating relation instances as well as providing necessary information for complementing tree kernel based system.

Even though there exists some gap between the performance of the YSRL and TRK system, we still can claim that the path between the NEs in the parse trees plays a critical role in relation extraction. Moreover, the idea of SRL can be applied successfully in guiding feature extraction along the path, which therefore could make SRL systems more general and adaptable to other types of relation extraction tasks.

6.2 Hybrid Models

6.2.1 Generative vs. Discriminative

In sections 5.3 and 6.1 we introduced a generative model, ZParser, and a discriminative model, YSRL, for BPL relation extraction. ZParser captures both syntactic and semantic information and simultaneously identifies NEs and relations. However, it is limited by the sparse data problem due to the significant number of zero occurrences of joint events. Moreover, ZParser is built from very local contexts: head constituents, modifiers and parent nodes, therefore may not be able to capture more complex relationship patterns. In addition, ZParser is capable of predicting all types of annotations indicated by the training model, but only providing the size of the training set be large enough.

The discriminative model YSRL is basically a classification model based on relation patterns extracted from syntactic parsing information. In contrast to the generative model, the discriminative approach usually has error propagation problems. For instance, an annotation error could be enlarged in the classification phase.

In this section we now aim to integrate ZParser and YSRL, with the hope that they could help each other to further improve their performance on the relation extraction task. The other reason to combine these two models is that YSRL itself cannot identify NEs. But given NEs identified by ZParser, YSRL is able to participate in our relation extraction system.

6.2.2 Pipelined System

As introduced in Section 5.3, the generative model ZParser identifies relevant NEs and extracts BPL relations at the same time. ZParser significantly outperforms two baseline systems: NE co-occurrence and Snowball. In contrast to ZParser, the discriminative model, tree-kernel-based YSRL, was shown effective on the BPL extraction, but does not identify NEs. Therefore, YSRL cannot be applied to the BPL prediction task directly, unless relevant NEs are predicted and provided to YSRL.

The idea of the pipelined system is running ZParser and YSRL in series, so that NEs predicted by ZParser are taken by YSRL to predict BPL relations. We use standard evaluation metrics as described in Section 4.3, which are precision, recall and F-score based on the full NE recognition and full relation extraction. The data sets are positive and negative examples split into training and test sets, as shown in Table 4.2.

There are two ways to make BPL predictions from the pipelined system:

1 predictions made by YSRL only

2 union of predictions of ZParser and YSRL. The **Confidence Score** of the final prediction is obtained by Equation 6.1.

$$Confidence(r\_UNION) = Confidence(r\_YSRL) + e^{Confidence(r\_ZParser)} \quad (6.1)$$

Note that a relation predicted by ZParser but not YSRL is given $Confidence(r\_YSRL) = 0$. Since $Confidence(r\_ZParser)$ is a log linear likelihood, a relation that is not predicted by ZParser is assigned $Confidence(r\_ZParser) = -\infty$.

Table 6.5 shows evaluation results of ZParser and the pipelined system with the above two ways of making BPL predictions.

Taking "noisy" NEs from ZParser, YSRL makes much fewer predictions than ZParser does, which results in a large decrease of recall. This again suggests that noisy data may not work effectively for YSRL on the relation extraction task. However, YSRL extracts some new BPL relations and the combined results outperform ZParser alone by 2.6% on F-score.

Measure	ZParser	Pipeline System	
		YSRL	ZParser+YSRL
Precision (%)	58.6	47.1	**59.4**
Recall (%)	26.2	12.3	**29.2**
F-score (%)	36.2	19.5	**39.0**

Table 6.5: Comparison of evaluation results of ZParser and the pipelined system.

6.2.3 Co-training System

Co-training is a bootstrapping-based algorithm [9], in which there are two different "views" on training examples, and two learners corresponding to these views label unlabeled training examples for each other. The idea of co-training may apply to our relation extraction task, such that, in each iteration, ZParser and YSRL label a large set of unlabeled data, which are then added into the training set for each other.

In this section, we propose four variations of the co-training algorithm and evaluate them with the standard data sets listed in Table 4.2 and evaluation metrics described in Section 4.3.

Co-training Algorithm 1

Figure 6.3 illustrates a high-level architecture of the co-training algorithm that alternatively calls ZParser and YSRL. The algorithm makes PL and BP binary predictions separately and combines them into ternary relations when evaluated.

Specifically, in each iteration, YSRL is trained on the curated training set and examples predicted by ZParser in the last iteration, if any. The trained YSRL is then used to classify a development set, which is the noisy data set with PROTEIN, BACTERIUM and LOCATION NEs appearing in each sentence, as introduced in Section 5.3.7. We then pick the top N_1 predictions from the development set and build the path between NEs on the top of the automatically annotated parse tree, the same tree augmentation method described in Section 5.3.2.

Following a similar process, ZParser is trained on the curated training set and the top N_1 positive predictions by YSRL and then parses the development set. Parse trees of the top N_2 positive and top N_2 negative predictions from the development set

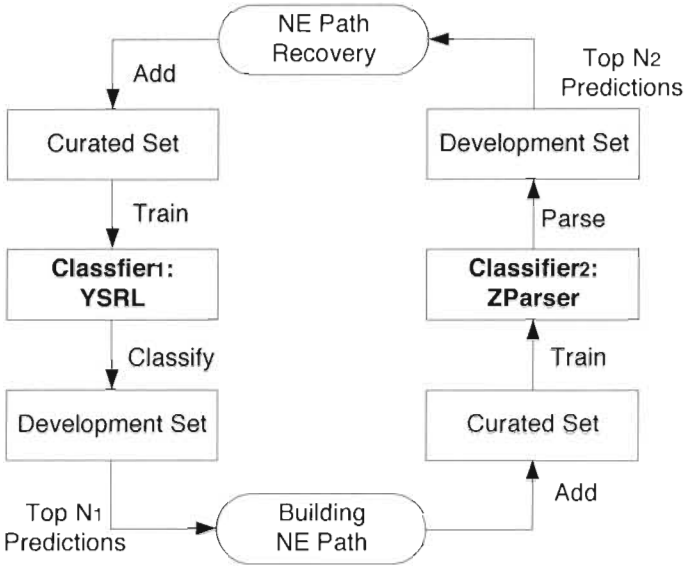

Figure 6.3: High-level architecture of the co-training algorithm that integrates ZParser and YSRL

are recovered to include PROTEIN NEs that are not identified, the same process as described in Section 5.3.4. The top N_2 positive/negative predictions are then added into the training set for YSRL in the next iteration.

The pseudo-code of the co-training algorithm is shown in Appendix B.

It is clear that the choice of N_1 and N_2 is crucial to the effectiveness of the algorithm, since we expect to remove the noise from the development set by making a small set of predictions, while keeping the prediction list growing in each iteration.

In this section, we experiment with four different settings on N_1 and N_2. In Co-training Algorithm 1, we grow the prediction list by add 20 more predictions in each iteration. For instance, at the ith iteration, the number of predictions $P_i = 20 \times i$. The reason we choose 20 is that the size of the curated training set is about $1,600$ (for both PL and BP) and 1/80 would be a reasonable growth rate for the training set.

In each iteration, ZParser and YSRL are evaluated against the standard test set (see Table 4.2). In addition, we combine the predictions of ZParser and YSRL and find

the combined results improve the overall performance. Table 6.6 shows the evaluation results of ZParser, YSRL and combined predictions of Co-training Algorithm 1 in its first ten iterations.

Iteration ID	YSRL (%)			ZParser (%)			Combined Results (%)		
	P	R	F1	P	R	F1	P	R	F1
1	40.0	3.1	5.7	41.7	7.7	13.0	31.3	7.7	12.3
2	40.0	3.1	5.7	46.2	9.2	**15.4**	37.5	9.2	14.8
3	57.1	6.2	11.1	41.7	7.7	13.0	43.8	10.8	**17.3**
4	100.0	6.2	**11.6**	50.0	7.7	13.3	50.0	7.7	13.3
5	75.0	4.6	8.7	50.0	7.7	13.3	45.5	7.7	13.2
6	75.0	4.6	8.7	15.4	3.1	5.1	31.3	7.7	12.3
7	0	0	0	11.1	1.5	2.7	10.0	1.5	2.7
8	0	0	0	57.1	6.2	11.1	40.0	6.2	11.7
9	0	0	0	11.1	1.5	2.7	11.1	1.5	2.7
10	0	0	0	45.5	7.7	13.2	38.4	7.7	12.8

Table 6.6: Performance of Co-training Algorithm 1: results of ZParser, YSRL and combined predictions of ZParser and YSRL in first ten iterations.

In the first three iterations, the F-score of combined predictions increases gradually from 12.3% to 17.3%, but then goes down. The performance decreases dramatically at the 6th iteration and YSRL produces no true positive predictions thereafter. A major reason of the decrease is that, as the number of predictions is getting larger, the classifiers cannot effectively make predictions with good quality, due to the very small size of training data.

Co-training Algorithm 2

The failure of the Co-training Algorithm 1 reminds us of a similar situation when we experimented with the generative model in Section 5, where a large set of noisy training data compensated for the lack of curated data.

According to the principles of semi-supervised learning, we implement the Co-training Algorithm 2 by adding as many predictions as possible to the curated training set, and hope that the quality of predictions would get better after each iteration. Specifically, all positive examples predicted by YSRL are added into the training set of Zparser; all predictions made by ZParser are added into the training set of YSRL in the next iteration.

Table 6.7 shows the performance of the Co-training Algorithm 2 in first two iterations. It is not surprising that YSRL does not perform well with the large set of noisy data, since discriminative models have been shown unsuitable for outlier detection [63].

Iteration ID	YSRL					ZParser				
	P	R	F1	# of PL	# of BP	P	R	F1	# of PL	# of BP
1	0	0	0	645/987	1565/3085	31.3	7.7	12.3	29/0	333/0
2	0	0	0	649/1169	2474/3683	28.6	3.1	5.6	32/0	265/0

Table 6.7: Evaluation results of Co-training Algorithm 2. The table also contains the number of positive/negative examples added into the training set of ZParser and YSRL.

Co-training Algorithm 3

Previous experiments show that the inclusion of noisy data by setting N_1 as large as possible works well for ZParser, but not for YSRL. Therefore, in this experiment, we treat YSRL and ZParser differently by introducing noisy data to ZParser and maintain a reasonable growth rate of YSRL's training set.

Specifically, $N_2 = 20$ for both positive and negative predictions made by ZParser, similar to that of Co-training Algorithm 1. Moreover, we add as many as possible positive and negative predictions from YSRL to the training set of ZParser, on the condition that ratio of positive predictions, N_1^+, to negative predictions, N_1^-, is $1:2$. NE and relation annotations of these negative predictions are removed from parse trees. The reason we set the ratio is that the ratio of positive to negative examples in the curated training set is $1:2$. In summary:

- $N_2 = 20$ for both positive and negative examples
- Assuming YSRL makes p positive predictions and q negative predictions. If $p > q/2$, then $N_1^+ = q/2$ and $N_1^- = q$; otherwise, $N_1^+ = p$ and $N_1^- = 2p$.

Evaluation results of the Co-training Algorithm 3 listed in Table 6.8, however, indicate that the inclusion of negative predictions in the training set significantly

decreases the performance of ZParser. Evaluation results of the first two iterations show that ZParser produces a very small amount of NEs and relation annotations and thus makes few BPL predictions, because probability distributions of NE and relation tags are "diluted" by the negative examples.

Iteration ID	YSRL			ZParser				
	P	R	F1	P	R	F1	# of PL	# of BP
1	40.0	3.1	5.7	0	0	0	30/60	334/668
2	0	0	0	0	0	0	33/66	266/532

Table 6.8: Evaluation results of Co-training Algorithm 3. The table also contains the number of positive/negative examples added into the training set of ZParser.

Co-training Algorithm 4

Evaluation results of the co-training algorithm 4 indicate that ZParser does not prefer negative examples. Therefore, in the last experiment experiment on the co-training system, we run the Co-training Algorithm 3, except that no negative examples are feed to ZParser. The results are shown in Table 6.9.

Iteration ID	YSRL			ZParser			
	P	R	F1	P	R	F1	# of PL / BP
1	40.0	3.1	5.7	29.4	7.7	12.2	30 / 334
2	16.7	1.5	2.8	42.9	9.2	15.2	33 / 266
3	20.0	1.5	2.9	30.0	4.6	8.0	33 / 236
4	20.0	1.5	2.9	13.3	3.1	5.0	25 / 215
5	0	0	0	20.0	4.6	7.5	22 / 194
6	0	0	0	25.0	4.6	7.8	23 / 184
7	0	0	0	25.0	4.6	7.8	18 / 176
8	0	0	0	13.3	3.1	5.0	12 / 196
9	0	0	0	15.4	3.1	5.1	9 / 168
10	0	0	0	23.1	4.6	7.7	10 / 59

Table 6.9: Evaluation results of Co-training Algorithm 4. The table also contains the number of positive examples added into the training set of ZParser.

This variation of the co-training system is referred as Co-training Algorithm 4. It is similar to Co-training Algorithm 1, except that ZParser was trained on many more positive examples. However, YSRL with both algorithms makes no correct predictions

after a few rounds. Besides, ZParser with Co-training Algorithm 4 performs even worse.

Summary

We introduced the co-training system that integrates the generative model ZParser and the discriminative model YSRL, such that data labeled by one model are used to train the other model.

YSRL is trained on positive and negative examples, while ZParser is trained on the positives ones. Experiments were carried out on the co-training system with respect to different growth rates of the training set.

In Co-training Algorithm 1, the top 20 predictions made by one model are added into the training set of the other model; while in Co-training Algorithm 2, all predictions are picked. Co-training Algorithm 3 combines ideas of two forerunners, by adding top 20 predictions of ZParser to the training set of YSRL and as many as possible predictions of YSRL to the training set of ZParser. In addition, negative predictions of YSRL are also picked by ZParser, on the condition that predictions being picked and the original training set have the same positive/negative ratio. We then found ZParser does not work well at all with the negative training examples. Co-training Algorithm 4 is the same as Algorithm 3 but the ZParser does not take negative predictions from YSRL. Table 6.10 lists a summary of the four co-training algorithms in terms of their growth rates of the training set.

Co-training Algorithm	**YSRL**		**ZParser**	
	Positive	**Negative**	**Positive**	**Negative**
1	20	20	20	-
2	all	all	all	-
3	20	20	all*	all*
4	20	20	all	-

Table 6.10: A summary of the four co-training algorithms in terms of their growth rates of the training set. *: negative examples is twice as large as positive examples in the Co-training Algorithm 3.

The evaluation results of the first two algorithms suggest that YSRL favors small

growth rates on its training set and ZParser prefers a large amount of training data, even with noise. However, providing ZParser with as much training data as possible does not make YSRL learn effectively after the first few rounds. Co-training Algorithm 3 also finds that training ZParser on negative examples results in very few BPL predictions. In summary, none of the proposed co-training algorithms outperforms the semi-supervised ZParser as introduced in Chapter 5.

We believe that the idea of the co-training system would be the right direction for further progress in integrating generative and discriminative models and in enabling them to compensate each other for general classification tasks. The major reason our proposed co-training algorithms do not compete with the semi-supervised method is that the curated data set is too small to effectively identify NEs and relations. Previous experiments indicate that the semi-supervised ZParser may not benefit much from adding more curated training data, but we would expect a significant improvement to the co-training system by a larger training set.

Both the generative and discriminative models introduced in previous and current chapters work for relation extraction from single sentences only. However, a considerably large portion of BPL relations in our corpus exists in multiple sentences. In the next chapter, we will purpose a graph-based model that extracts BPLs from multiple sentences, based on predictions made by the generative and discriminative models, to further improve the overall performance of the system.

Chapter 7

Biomedical Relation Networks

7.1 Introduction

Both the parsing-based method and the discriminative model limit themselves to relation extraction from single sentences only. However, in our training and test sets, about 26% of BPL relations are from multiple sentences. In other words, single-sentence relation extraction cannot achieve a recall over 74%.

The following example shows a relation, BPL($Bacillussubtilis$, $TreA$, $Cytoplasm$), from two consecutive sentences, in which each sentence contains a binary relation. A BPL may be extracted from non-consecutive sentences of a single document or from multiple documents[1].

Example 7.1: *A 2.5 kb DNA fragment contain a gene encoding a* [$_{PROTEIN}$ *phospho-alpha-(1-1)-glucosidase*] ([$_{PROTEIN}$ *phosphotrehalase*]), *designated* [$_{PROTEIN}$ *treA*], *was isolated from a* [$_{BACTERIUM}$ *Bacillus subtilis*] *chromosomal library by complementation of the tre-12 mutation. The major* [$_{PROTEIN}$ *TreA*] *activity was found in the* [$_{LOCATION}$ *cytoplasm*].

A Biomedical Relation Network (BRN) is a graph model that represents relations from multiple sentences and even multiple documents. In general, a BRN is a data structure that stores relations between biomedical substances as a directed weighted

[1]Our training and test sets currently do not contain BPL relations across multiple documents.

cyclic graph, in which each node represents a biomedical NE (i.e., PROTEIN, BACTERIUM or LOCATION NE in this task) that is the name of the node, and each link represents the relation (i.e., BP or PL) between two NEs being linked. Each node has a weight indicating the confidence of its name being a correct NE. Each link is also associated with a weight, which is the probability of the relation being correct.

There are two aspects of BRNs that we will consider:

1. the construction of BRNs from sentences,

2. the utilization of BRNs: extracting relations from them.

Figure 7.1 shows a high-level illustration of these two aspects, with Sections 7.2 and 7.3 below providing details. The work related to BRNs are discussed in Section 7.4. Experiments and evaluations are described in Section 7.5. Discussion about BRNs is given in Section 7.6.

7.2 BRN Construction

A BRN contains two types of links. One is associated with the functional relation r, while the other represents the ontological relation. As shown in Figures 7.2 and 7.3, ontological links connect NEs with their ontologically related NEs, e.g., parents, children, hypernyms, hyponyms, similar NEs; while functional links connect pairs of NEs with a designated biomedical relation, e.g., up-regulation or BPL. A *link weight* (valued between -1 and 1) is assigned to each link representing the similarity or probability of two NEs being related.

In our task, a BRN is constructed from NEs and binary relations identified by the ZParser+YSRL pipelined model as introduced in Section 6.2.2. The weight of each link in the BRN is the confidence score as defined in Equation 6.1. Note that a link weight can be negative if a linguistic negation is detected in the sentence. However, the negation detection is not our research focus and thus not included in the proposed system. In addition, each node could also be assigned a *node weight*, in which case the NE represented by the node is recognized as a biomedical NE with certain probability.

In summary, the following steps are carried out to build a BRN:

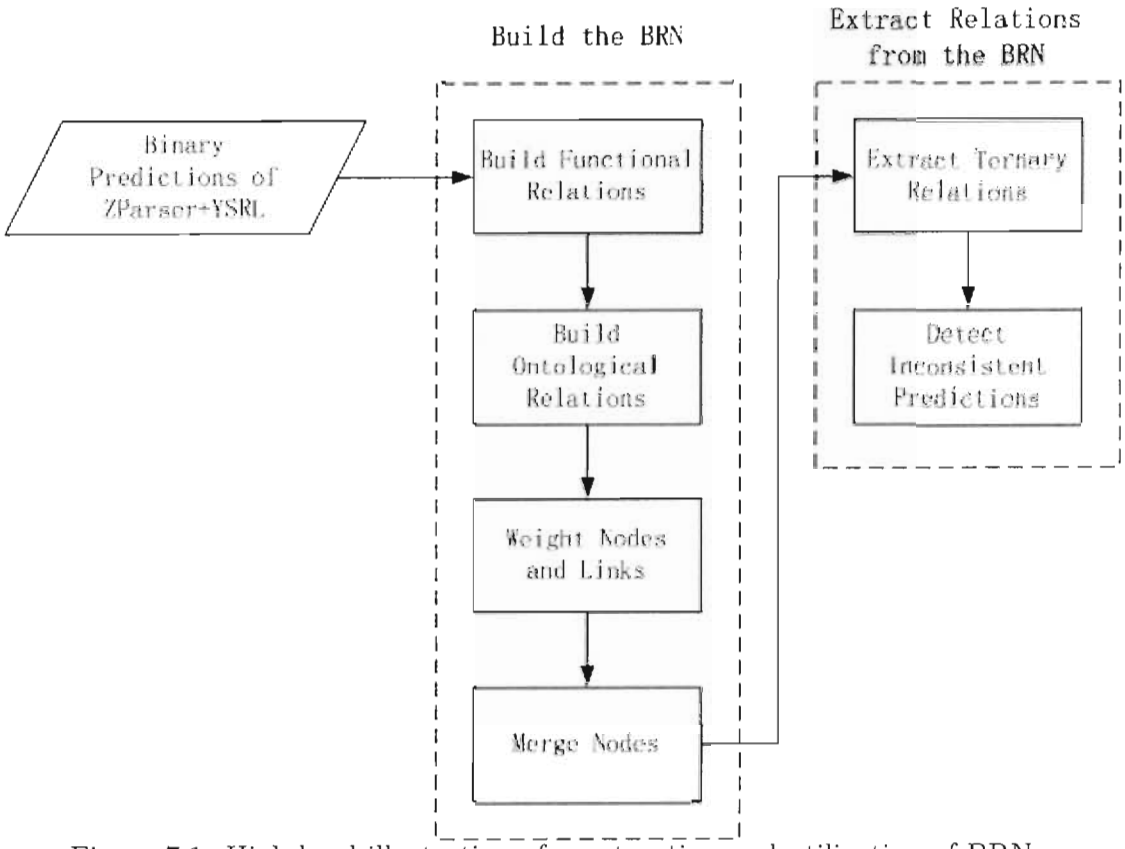

Figure 7.1: High-level illustration of construction and utilization of BRNs.

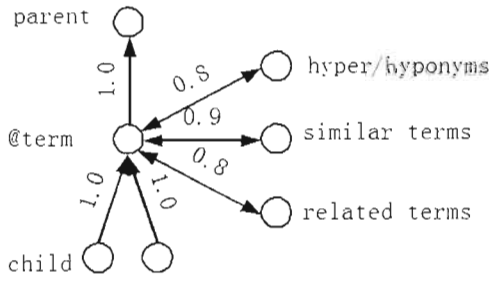

Figure 7.2: Ontological relations

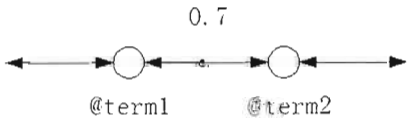

Figure 7.3: Functional relations

- Connecting each pair of nodes with the functional relations as shown in Figure 7.3. As introduced above, these relations are binary predictions (PL and PO) from the ZParser+YSRL pipelined model as described in Section 6.2.2, so far the best performing system. The nodes correspond to PROTEIN, BACTERIUM and LOCATION NEs.

- Building and connecting nodes with the ontological relations as shown in Figure 7.2. We search the NE of each node d in the UMLS Metathesaurus and look for its children, hyponyms and similar NEs. Each of these related NEs is then represented by a node in the BRN and linked to the node d.

- Weighting each node and link. The weights of links are confidence scores produced by the ZParser+YSRL model. Ideally the weights of nodes would be probabilities given by an NE Recognizer, but since the ZParser+YSRL model only produces one confidence score for each predicted relation, we assign 1 to the weights of all nodes for now, including the ones found in the dictionary.

- Inter-connecting links by merging nodes that represent the same NE. The node weight of merging two nodes w_1 and w_2 is: $Merged(w_1, w_2) = w_1 + w_2 - w_1 * w_2$. The purpose of the inter-connection is to increase the confidence of an NE that occurs multiple times within the context, when in the future the node weights are given by some NE recognizer and thus are not necessarily 1.

Figure 7.4 illustrates a portion of the BRN consisting both ontological and functional links to represent the example in Section 7.1. Note that, during the evaluation to be described in Section 7.5, we do not include the related NEs obtained from the dictionary in the final predictions, since the gold answer generally does not contain these NEs and thus we are unable to evaluate them.

A **window** is applied when two nodes are merged. It also controls the distance of two sentences, across which a BPL is extracted. The window size w denotes the distance of two sentences. $w = 1$ indicates that two sentences are consecutive; $w = n$ indicates that there are $n - 1$ sentences between the two sentences from which the BPL is extracted. $w = \infty$ when these two sentences belong to different documents.

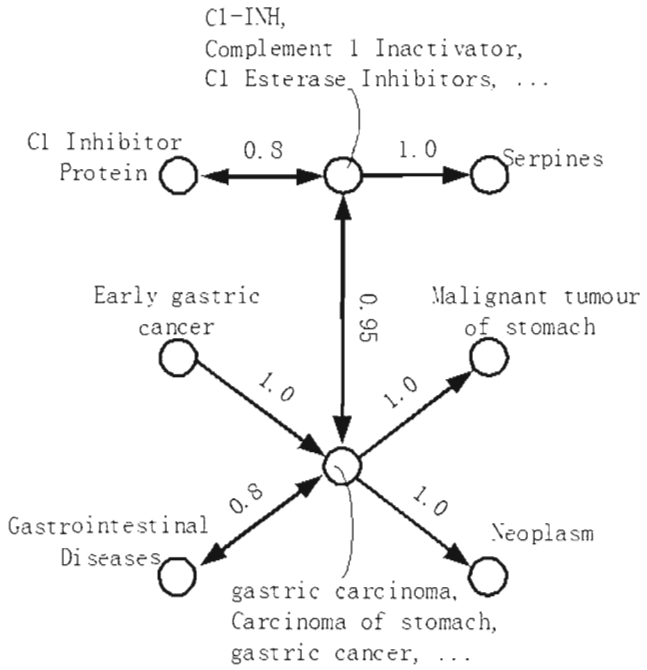

Figure 7.4: A portion of BRN

7.3 Relation Extraction from BRN

Once the BRN for the entire corpus is constructed, we can predict BPL relations from multiple sentences and documents by finding a sub-graph that contains a BACTERIUM node, a PROTEIN node and a LOCATION node in the BRN, as illustrated in Figure 7.5. Assuming two binary relations that have been extracted at single-sentence level: *PL(PROTEIN_1, LOCATION_1)* and *BP(, BACTERIUM_1, PROTEIN_1)*, a new prediction can be made on *BPL(BACTERIUM_1, PROTEIN_1, LOCATION_1)*, along with a significance score, i.e., 0.8*0.5 = 0.4. This new BPL relation is predicted from multiple sentences in single or multiple documents within the specified window size. Generally speaking, the larger the window size, the better the recall. However, the precision suffers as a result. In Section 7.5 we will experiment with a set of window sizes and show how the window size impacts on the overall performance.

Moreover, inconsistent BPL predictions can be detected on top of the constructed

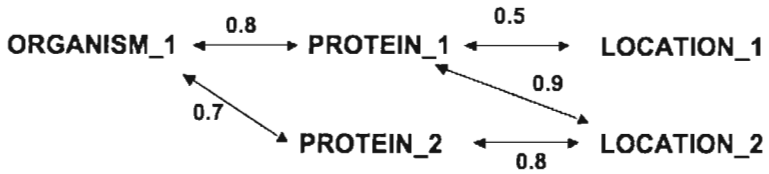

Figure 7.5: An illustration of BPL relation identification

BRN, with an assumption that one protein only is present at one location in the specific bacterium (this is mostly true). Suppose two BPLs are discovered: *BPL(ORGANISM_1, PROTEIN_1, LOCATION_1)* and *BPL(ORGANISM_1, PROTEIN_1, LOCATION_2)*, as shown in Figure 7.5. The one with lower significance score can then be removed.

7.4 Related Work

Graph models applied to the general relation extraction task were described in Section 2.2.6. Here we reiterate relevant graph-based approaches and compare them with BRNs.

Concept Chain Graphs (CCGs) for discovering unknown associations between concepts are introduced in InfoXtract [103, 102]. A chain graph is a probabilistic network model that mixes undirected and directed graphs to give a probabilistic representation that includes Markov random fields and Markov models. It is a hybrid probabilistic IR framework combining a traditional bag-of-words model with higher-level concepts and relations provided by an IE system. The CCG is implemented as a multilevel index where the highest level represents relations, the middle level represents concepts, and the lowest level represents a word index. However, these relation, concepts and word indexes are weighted based on the frequency of occurrence and predicted relations may not reflect actual meanings. Furthermore, in contrast to BRNs, CCGs are built on the top of the bag-of-words model thus unable to represent linguistic assets to the relation extraction task.

The Concept Space introduced in [60] provides a co-occurrence network of semantically related concepts that form relations containing two noun phrases (NPs). It

provides a network of semantically related concepts that form relations for the entire collection. Each relation is directional and contains two NPs and a weight of co-occurrence analyzed based on the asymmetric cluster function [16] to indicate its strength of relevance. The operation on the Concept Space is a bottom-up technique that captures relations between pairs of NPs from large collections of text. However, in contrast to BRNs, the concept space cannot represent named relations and is unable to represent relations such as pathway and negation.

Mack et al. proposed the *Lexical Networks* [67], which apply data-mining techniques to graphs that are derived from syntactic parse trees, where the nodes in a graph represent proteins, and the links represent relatively strong co-occurrences between these proteins within a sentence or paragraph. Strong co-occurred proteins are linked with strength. In contrast, BRNs are built on the top of proposed generative and discriminative models, which provide state-of-the-art methods to extract BPLs from syntactic and semantic characteristics of text. Moreover, although the pair-wise term relations of the lexical Networks can be compiled into longer sequences that span multiple documents, lexical networks are basically the visualization tool of unnamed relation between proteins and are not used to extract hidden information.

Similarly, PubGene [47] and the inference network introduced in [115] are built on co-occurrences of gene pairs, therefore, edges are still the representations of un-named relations and may not reflect actual meanings.

Mcdonald et al. present a two-stage method for n-ary relation extraction from Medline abstracts in [70]. Similar to building and utilizing BRNs, their first stage creates a graph from pairs of named entities that are likely to be related, and the second stage finds potential n-ary relation instances in that graph. In contrast, they predict binary relations using a maximum entropy classifier and get around the named-entity recognition problem by assuming named entities are known. The potential n-ary relation instances are predicted by finding maximal cliques[2], which are however always exponentially many, since the graph is fully connected.

[2]A *clique* C of G is a subgraph of G in which there is an edge between every pair of vertices. A *maximal clique* of G is a clique $C = (V_C, E_C)$ such that there is no other clique $C' = (V_{C'}, E_{C'})$ such that $V_C \subset V_{C'}$.

7.5 Experiments and Evaluations

We build a BRN from the binary predictions of ZParser+YSRL (described in Section 6.2.2) on the standard data sets (see in 4.2), which are positive and negative curated data split into the training and test sets. BPL relations are then predicted from the BRN corresponding to a set of different window size w: 1, 2 and 5. Evaluation results of the BRN with different window sizes are listed in Table 7.1. The reason we do not choose w equal to ∞ is that the test set includes no BPL relations from multiple documents.

Measure (%)	**ZParser**	**Pipelined System (ZParser+YSRL)**	**BRN**		
			w=1	**w=2**	**w=5**
Precision	58.6	59.4	**64.7**	61.1	51.2
Recall	26.2	29.2	**33.8**	33.8	33.8
F-score	36.2	39.0	**44.4**	43.5	40.7

Table 7.1: Evaluation results (in percent) of ZParser, pipelined system and the BRN with the window size $w = 1$, 2 and 5.

When $w = 1$, the system correctly extracts 4 BPL relations from consecutive sentences in the BRN. When enlarging the window size, the system makes more predictions, which are all incorrect. The fact is that BPL relations across non-consecutive sentences in our test set are infrequent. So as the window size is increased, there is a dramatic increase in the number of predictions, very few of which will have a chance to be correct. The experiment suggests that $w = 1$ would the best choice for this task.

The system performs effectively on the inconsistency detection. It finds that two BPL relations identified by ZParser+YSRL are inconsistent with each other as listed in Table 7.2: They have the same PROTEIN and BACTERIUM NEs, but predicted LOCATION NEs are different. The system then changes the BPL prediction with the lower confidence score to negative and thus slightly improves the precision and F-score.

BACTERIUM	PROTEIN	LOCATION	Confidence Score
T. pallidum	lipoprotein	periplasmic	0.676
T. pallidum	lipoprotein	cytoplasmic membrane	0.703

Table 7.2: Two BPL relations predicted by ZParser+YSRL are found inconsistent.

7.6 Discussion

BRNs make it possible to identify the BPL relations from multiple sentences and documents. In addition, the nature of BRNs would make the following problems easier to handle:

- Answering questions, such as, which genes are most likely relevant to specific cancers, and vice versa. In general, correctly identifying NEs and relations among them would be one of the crucial tasks of a question-answering system. The BRNs could be one of the right tools to identify NEs and relations and thus could be a solution to the question answering task.

- Finding the biological pathways between any medical substances by choosing a path between them with the largest weight. In this BPL extraction task, we have only illustrated how to find ternary relations from the BRNs. The same idea can be applied to find biological pathways, which are basically n-ary relations, from the BRNs. Then the task would be finding a path of nodes and links on the BRN between two unknown nodes. The types of intermediate nodes and links may be known or unknown, depending on the requirements of the specific task.

- Identifying the relation between any pair of NEs also indicates relations between their ontologically related NEs.

- Exploiting relations among biomedical articles by clustering a BRN. Nodes in the BRN connect two articles if their NEs occur in both articles. By clustering the BRN based on which articles the nodes occur in, the links between two clusters would indicate the relations of the corresponding articles.

- Performing multiple document summarization. Once a BRN is built from multiple documents, it would be possible to choose paths with large weights on nodes and links along the paths in the BRN. These paths are supposed to be more important than other paths and could be taken as a summary of multiple documents.

Generally speaking, the designated relation between arbitrary biomedical substances can be found in BRNs and may indicate useful information that has not previously been recognized. The process of relation extraction from BRNs may be easily applied to the extraction of other relations from multi- sentences and documents, for instance, to find the ternary relation $(COMPANY, TITLE, PERSON)$.

Our analysis of prediction results suggests that co-reference resolution would be very helpful to this graph-based model. However, we have not found any well-performing co-reference resolution systems and co-reference resolution is not our research focus. Leaving this topic for future research is a reasonable approach for now.

Chapter 8

Conclusion and Future Work

In this dissertation we introduce the task of biomedical function relation extraction from MEDLINE articles. The specific relation we are working on is Bacterial Protein Subcellular Localization, a ternary relation among a bacterium, protein and location. Specifically, the task is to identify BACTERIUM, PROTEIN and LOCATION Named Entities (NEs) from the articles and determine whether they interact with each other to achieve certain biomedical functionality.

Before introducing the relation extraction system, we described our biomedical IR system, which in generally is taken as a coarse level of information extraction. The system participated in the TREC 2005 ad-hoc retrieval task in the Genomics track, where it attempted to find documents relevant to answers complex questions. Built on top of a conventional information retrieval toolkit, the system applies a synonym-based query expansion from various biomedical sources to retrieval relevant documents. It then re-rank the retrieved documents using a boosting-based algorithm, which is capable of capturing natural language sub-structures embedded in text. Experiments show that the algorithm works well in cases where the conventional information retrieval system performs poorly.

Our relation extraction system takes structured MEDLINE articles and predicts *BPL(BACTERIUM, PROTEIN, LOCATION)* relations from them. A preprocessing module is applied to automatically annotate these articles with syntactic and shallow semantic analysis, including syntactic parsing and biomedical NER. We went through a four-round curation process. In each round, a boosting algorithm on syntactic

subtree features was used to coarsely classify sentences; biologists then reviewed the classified sentences and review results were added into the training set for the next round. This curation process enabled us to obtain more curated data with the help of automatic machine learning algorithm and avoided laborious manual curation. The number of positive examples increases from 72 to 333. Finally, a three-tier relation extraction module predicts BPLs from the annotated articles.

For the first tier, a parsing-based generative model extracts relations from single sentences, by performing syntactic and shallow linguistic analysis. The parser integrates domain-independent syntactic information and domain-specific semantic information on parse trees, and is capable of identifying NEs and extracting relations simultaneously. We propose a semi-supervised generative model that takes advantage of noisy data generated by the automatic annotator and greatly improves the overall performance compared with the supervised alternative. The semi-supervised model also significantly outperforms a naive NE-co-occurrence baseline system by 13.2% of F-score and a well-known text mining system (Snowball) by 7.6% of F-score.

For the second tier, a discriminative model integrates rich syntactic features from parsing trees to extract relations from single sentences. To further improve the overall system performance, we combine the generative model and the discriminative model in a pipelined system and a co-training system. The pipelined system improves both precision and recall of the semi-supervised generative model and increases F-score by 2.8%.

Lastly, a graphical representation of BPL relations, a Biomedical Relation Network (BRN), is proposed to find global and hidden relations from multiple sentences and articles. The BRN integrates ontological and functional relations in a directed weighted cyclic graph. Based on binary predictions of the generative and discriminative models, the BRN is capable of extracting BPLs and detecting inconsistent predictions. Our evaluation shows that BRNs increase F-score of the pipelined system by 2.2%. Due to the limitation of our test data, we only evaluate the performance of BRNs predicting relations from a single article.

We summarize our contributions to the task of extracting biomedical functional relations from text as follows:

- Protein subsellular extraction has been studied in the BioNLP community for years, but researchers mainly focused on the binary relations (e.g., between organism and protein) of the eukaryotic SCLs. The BPL relation is fairly new to both the molecular biology and the BioNLP research.

- One of major problems of most BioNLP tasks is the lack of curated data. In addition to a very limited amount of curated data, we made use of a large data set with noise to train the relation extraction model, i.e., the semi-supervised generative model, which significantly improves the overall performance.

- We proposed a parsing technique that integrates syntactic and semantic information and identifies NEs and relations simultaneously. The approach is different from the bootstrapping technique in that the information is not extracted iteratively and thus there is no error propagation to the next iteration. Identifying NEs and relations at the same time would be new in the biomedical information extraction research area.

- We studied the relevance of the Semantic Role Labeling (SRL) task and the BPL extraction task, and applied features normally used by SRL systems to a discriminative model for the BPL extraction.

- We proposed a graphical representation for BPL relations and utilized the representation to effectively predict relations from multiple sentences.

- We had a set of MEDLINE articles curated by biologists, in order to train and evaluate various models. The proposed system will finally make predictions from the whole MEDLINE database, which contains over 12 million articles. The predictions will then be judged by biologists and added into the curated corpus, which would be an new asset to the NLP community. To build a publicly available corpus would greatly benefit the BioNLP research on relation extraction.

Our experiences in this research suggest that the BPL relation extraction is a very hard task. The system should be able to not only identify NEs, but also tell which NEs

occurring together can perform certain biomedical functionalities. Furthermore, the lack of curated data makes it very difficult to train an effective system. Our hope is that, with unremitting efforts on researching various models and gradually increasing the size of the curated data set, the BPL and other relation extraction tasks would be more and more attractive to the BioNLP community and beneficial to other relevant research.

Appendix A

Biomedical data sources

Most biomedical relationship identification systems make use of some publicly accessible data sources, such as annotated articles, ontologies and other databases. These data sources were generated, collected or curated by professionals and have been opened to the public with two major purposes. One is to help the IE systems save the time and efforts on building or collecting data. The other purpose is to make evaluations among IE systems that use the same data sources easier.

The most popular biomedical data sources are briefly introduced as follows.

- **Medical Literature Analysis and Retrieval System Online** (**MEDLINE**[1]) is the bibiographic database at National Library of Medicine (NLM[2]). It includes approximately 13 million references to biomedical articles from 4,800 journals in more than 70 countries from the year 1950. MEDLINE uses a language called Medical Subject Headings (MeSH) to index the articles in the database. The volume of biological and medical research literature grows so rapidly that over 2,000 new articles are being added in MEDLINE everyday. It is now the largest biomedical information source in the world. MEDLINE is accessible via PubMed[3].

- **GENIA** corpus is a semantically annotated molecular and biological corpus

[1] http://www.nlm.nih.gov/pubs/factsheets/medline.html

[2] http://www.nlm.nih.gov/

[3] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?

based on the GENIA ontology, the taxonomy of a subset of the substances and the biological locations involved in reactions of proteins, developed by Tsujii laboratory of the University of Tokyo[4]. The corpus is annotated by POS tags and biomedical entity tags. Its latest version consists of 2,000 MEDLINE abstracts. GENIA intends to incorporate parsing technique and has already provided corpus in the Penn TreeBank style with 200 abstracts[5]. Some GENIA-based corpora by third parties have been developed to provide more linguistic information, such as coreference by MEDCo project[6] and dependency by OntoGene project[7].

- **Yeast Proteome Database (YPD)** is a model for the organization and presentation of comprehensive protein information based on the detailed curation of the scientific literature for the yeast Saccharomyces cerevisiae, in Proteome BioKnowledge Library Databases[8]. YPD contains 6,100 yeast proteins with more than 50,000 annotations lines derived from the review of 8,500 research publications. The information concerning each protein is structured around a convenient one-page format, the Yeast Protein Report, with detailed information or descriptions.

- **Flybase** is a database of the drosophila genome[9] produced by a consortium of researchers funded by the National Institutes of Health (NIH), U.S.A., and the Medical Research Council, London. It includes a bibliography of more than 81,000 Drosophila citations, information on more than 38,000 alleles of more than 11,000 genes, descriptions of over 13,300 chromosomal aberrations, Drosophila genetic map information, information on the functions of gene products, etc.

- **BioCreAtIvE** corpus. In the challenge cup of 2004, BioCreAtIve provided a

[4]http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/

[5]http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/topics/Corpus/GTB.html

[6]http://nlp.i2r.a-star.edu.sg/medco.html

[7]http://www.ontogene.org/

[8]http://www.proteome.com/YPDhome.html

[9]http://flybase.org/

corpus of training data for the task of automatic functional annotation using the Gene Ontology annotations[10].

- **Databases of Interacting Proteins** (**DIP**) experimentally determined interactions between proteins. It combines information from a variety of sources to create a single, consistent set of protein-protein interactions. The data stored within the DIP database were curated, both, manually by expert curators and also automatically using computational approaches that utilize the knowledge about protein-protein interaction networks extracted from the most reliable, core subset of the DIP data[11]. DIP currently consists of 17,556 proteins and 46,463 interactions from 2,884 articles and 34 other data sources.

- NLM's **Unified Medical Language System** (**UMLS**) provides biomedicine and health knowledge sources and associated software tools for system developments in building or enhancing electronic information systems as well as for informatics research about investigating knowledge representation and retrieval questions[12]. There are three UMLS Knowledge Sources: the Metathesaurus, the Semantic Network, and the SPECIALIST lexicon. They are distributed with several tools (programs) that facilitate their use, including the MetamorphoSys install and customization program.

- **Saccharomyces Genome Database** (**SGD**) is an organized collection of genetic and molecular biological information about Saccharomyces cerevisiae, bakers' and brewers' yeast. It contains the sequences of yeast genes and proteins; descriptions and classifications of their biological roles, molecular functions, and subcellular localizations; links to literature information; links to functional genomics datasets; and tools for analysis and comparison of sequences[13].

- **Medical Subject Headings** (**MeSH**) is NLM's controlled vocabulary used

[10] *http : //www.pdg.cnb.uam.es/BioLINK/workshop$_B$ioCreative$_0$4/results/agreement.html*

[11] http://dip.doe-mbi.ucla.edu/

[12] http://www.nlm.nih.gov/research/umls/

[13] http://www.yeastgenome.org/

for indexing articles for MEDLINE/PubMed[14]. MeSH terminology provides a consistent way to retrieve information that may use different terminology for the same concepts. It consists of sets of terms naming descriptors in a hierarchical structure that permits searching at various levels of specificity. There are 22,568 MeSH descriptors, more than 139,000 supplementary headings and thousands of cross-references.

- **Gene Ontology (GO)** project was established to provide a common language to describe aspects of a gene product's biology[15]. The objective of GO is to provide controlled vocabularies for the description of molecular functions, biological processes and cellular components of gene products. It started as a collaboration between three model organism databases, the Saccharomyces Genome Database (SGD), FlyBase (Drosophila), and Mouse Genome Informatics (MGI).

- The **Reference Sequence (RefSeq)** database provides a foundation for the functional annotation of the human genome, which consists of a biologically non-redundant collection of DNA, RNA, and protein sequences[16]. Each RefSeq represents a single, naturally occurring molecule from a particular organism. RefSeqs are frequently based on GenBank records but differ in that each RefSeq is a synthesis of information, not a piece of a primary research data in itself.

[14]http://www.nlm.nih.gov/mesh/meshhome.html

[15]http://www.geneontology.org/

[16]http://www.ncbi.nlm.nih.gov/RefSeq/

Appendix B

Pseudo code of Co-training Algorithm

1: $Classifier_1$ =YSRL
2: $Classifier_2$ =ZParser
3:
4: $L_1^{+/-}(0)$ =positive/negative curated set for $Classifier_1$
5: $L_2(0)$ =curated set for $Classifier_2$
6:
7: U =large set of data from medline which have B, P, L named entities of interest
8: $U_c = U$ parsed with Charniak-Johnson parser
9:
10: **for** $t = 0$ to T **do**
11: # train YSRL, the $Classifier_1$, on $L_1^{+/-}(t)$, the curated set for YSRL at tth iteration
12: **for** each parsed sentence p in U_c **do**
13: **for** each (B, P) in p **do**
14: **if** $Classifier_1$ output $= +1$ **then**
15: $p_{with-links}$ = p with LNK/PTR annotation
16: add $p_{with-links}$ in $L_2(t+1)$
17: **end if**
18: **end for**
19: **for** each (P, L) in p **do**
20: **if** $Classifier_1$ output $= +1$ **then**

```
            p_with-links = p with LNK/PTR annotation
            add p_with-links in L_2(t+1)
          end if
        end for
      end for

      # train ZParer, the Classifier_2, on L_2(t), the curated set for ZParer at tth
      iteration
      for each sentence s in U do
        parse s with Classifier_2 to provide parse tree p
        for each (B, P) in p do
          p_with-NEs = p with B, P NE annotations only
          if Classifier_2 output has confidence > c_1 then
            add p_with-NEs in L_1^+(t+1)
          else if Classifier_2 output has confidence < c_2, where c_2 < c_1 then
            add p_with-NEs in L_1^-(t+1)
          end if
        end for
        for each (P, L) in p do
          p_with-NEs = p with P, L NE annotations only
          if Classifier_2 output has confidence > c_1 then
            add p_with-NEs in L_1^+(t+1)
          else if Classifier_2 output has confidence < c_2 then
            add p_with-NEs in L_1^-(t+1)
          end if
        end for
      end for
    end for
```

Bibliography

[1] ACE. The ace evaluation plan. In Automatic Content Extraction (04), 2004.

[2] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In Proceedings of the 5th ACM International Conference on Digital Libraries (DL), 2000.

[3] J. Allen. Natural Language Understanding. Benjamin Cummings Publishing Company, New York, 1994.

[4] A. R. Aronson, T. C. Rindflesch, and A. C. Browne. Exploiting a large thesaurus for information retrieval. In Proceedings of RIAO 94, pages 197–216, 1994.

[5] D. Bikel. A distributional analysis of a lexicalized statistical parsing model. In EMNLP 2004, pages 182–189, 2004.

[6] C. Blaschke, M. Andrade, C. Ouzounis, and A. Valencia. Automatic extraction of biological information from scientific text: Protein-protein interactions. In Proceedings of the AAAI Conference on Intelligent Systems in Microbiology (ISMB '99), page 60C77, Heidelberg, Germany, AAAI, Menlo Park, CA, 1999.

[7] C. Blaschke, R. Hoffmann, J. C. Oliveros, and A. Valencia. Extracting information automatically from biological literature. Comparative and Functional Genomics, 2:310–313, 2001.

[8] C. Blaschke and A. Valencia. The frame-based module of the suiseki information extraction system. IEEE Intelligent Systems, 17(2):14–20, 2002.

[9] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98), pages 92–100, Madison, Wisconsin, United States, 1998.

[10] R. Bunescu, R. Ge, R. J. Kate, R. J. Mooney, Y. W. Wong, E. M. Marcotte, and A. K. Ramani. Learning to extract proteins and their interactions from medline

abstracts. In Proceedings of the ICML-2003 Workshop on Machine Learning in Bioinformatics, pages 46–53, Washington DC, Augest 2003.

[11] R. C. Bunescu and R. J. Mooney. A shortest path dependency kernel for relation extraction. In HLT/EMNLP-05, pages 724–731, October 2005.

[12] M. F. Caropreso and S. Matwin. A text representation for sentence selection. In Canadian AI 2006, pages 324–335, 2006.

[13] X. Carreras and L. Marquez. Introduction to the conll-2004 shared task: Semantic role labeling. In Proceeding of Conference on Computational Lingustics Learning (CoNLL04), pages 89–97, 2004.

[14] E. Charniak and M. Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 173–180, 2005.

[15] Eugene Charniak. A maximum-entropy-inspired parser. In Meeting of the North American Chapter of the ACL, pages 132–139, 2000.

[16] H. Chen and K. J. Lynch. Automatic construction of networks of concepts characterizing document databases. IEEE Transactions on Systems, Man and Cybernetics, 22(5):885–902, 1992.

[17] G. Claudio, A. Lavelli, and L. Romano. Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. In Proc. EACL 2006, Trento, Italy, April 5–7 2006.

[18] N. Collier, C. Nobata, and J. Tsujii. Extracting the names of genes and gene products with a hidden markov model. In Proceedings of COLING 2000, pages 201–207, 2000.

[19] D. P. Corney, B. F. Buxton, W.B. Langdon, J. Charlwood, P.M. Woollard, and D. T. Jones. Extracting biological information from full-length papers. Rn/03/17, UCL-CS, 2003.

[20] D. P. A. Corney, Langdon W.B. Buxton, B. F., and D. T Jones. BioRAT: Extracting Biological Information from Full-length Papers, volume 20 of 17. Bioinformatics, 2004.

[21] K. Crammer and Y. Singer. The algorithmic implementation of multiclass kernel-based vector machines. Technical report, School of Computer Science and Engineering, Hebrew University, 2001.

[22] M. Craven. The genomics of a signaling pathway: A kdd cup challenge task. Technical report, University of Wisconsin, 2002.

[23] M. Craven and J. Kumlien. Constructing biological knowledge-bases by extracting information from text sources. Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, Heidelberg, Germany 1999.

[24] A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), pages 423–429, Barcelona, Spain, July 2004.

[25] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In Proceedings of the Third Conference on Applied Natural Language Processing, 1992.

[26] N. Daraselia, S. Egorov, A. Yazhuk, and S. Novichkova. Extracting human protein interactions from medline using a full-sentence parser. Bioinformatics, 19(0):1–8, 2003.

[27] G. Demetriou and R. Gaizauskas. Tuilizing text mining resuts: The pastaweb system. In Proceedings of the Association of Computational Linguistics Workshop on Natural Langauge Processing in the Biomedical Domain, pages 77–84, Philadelphia, US, July 2002.

[28] J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. Mining medline: Abstracts, sentences, or phrases? Pacific Symposium on Biocomputing, 7:326–337, 2002.

[29] I. Donaldson, J. Martin, B. Bruijn, and C. Wolting. Prebind and textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. BMC Bioinformatics, 4(11), 2003.

[30] O. Emanuelsson, H. Nielsen, S. Brunak, and G. Von Heijne. Predicting subcellular lolocalization of proteins based on their n-terminal amino acid sequence. Journal of Molecular Biology, 300(10051016), 2000.

[31] E. Eskin and E. Agichtein. Combining text mining and sequence analysis to discover protein functional regions. In Pacific Symposium on Biocomputing, volume 9, pages 288–299, 2004.

[32] R. Feldman, Y. Regev, M. Finkelstein-Landau, E. Hurvitz, and B. Kogan. Mining biomedical literature using information extraction. Current Drug Discovery, pages 19–23, October 2002.

[33] C. Friedman, P. Kra, M.Krauthammer, H.Yu, and A.Rzhetsky. Genies: a natural-langauge processing system for the extraction of molecular pathways from journal articles. Bioinformatics, 17(1):74–82, 2001.

[34] Y. Fu, T. Bauer, J. Mostafa, M. Palakal, and S. Mukhopadhyay. Concept extraction and association from cancer literature. In Proceedings of the 4th international workshop on Web information and data management, pages 100–103, McLean, Virginia, USA, 2002.

[35] K. Fukuda, A. Tamura, T. Tsunoda, and T. Takagi. Toward information extraction: identifying protein names from biological papers. In Proc. of the Pacific Symposium on Biocomputing 1998 (PSB98), pages 707–718, 1998.

[36] R. Gaizauskas, G. Demetriou, P. J. Artymiuk, and P. Willett. Protein structures and information extraction from biological texts: The pasta system. Bioinformatics, 19(1):135 – 143, 2003.

[37] B. J. Grosz, A. K. Joshi, and S. Weinstein. Centering: A framework for modeling the local coherence of discourse. Computational Linguistics, 21(2):203–225, 1995.

[38] U. Hahn and M. Romacker. Creating knowledge repositories from biomedical reports: The medsyndikate text mining system. In Proceedings PSB 2002, pages 338–349, 2002.

[39] T. Hasegawa, S. Sekine, and R. Grishman. Discovering relations among named entities from large corpora. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), pages 415–422, Barcelona, Spain, July 2004.

[40] A. Hoglund, T. Blum, S. Brady, P. Donnes, J. Miguel, M. Rocheford, O. Kohlbacher, and H. Shatkay. Significantly improved prediction of subcellular localization by integrating text and protein sequence data. In Pac. Sym. on Biocomp., volume 11, pages 16–27, 2006.

[41] A. Hoglund, P. Donnes, T. Blum, H. Adolph, and O. Kohlbacher. Using n-terminal targeting sequences, amino acid composition, and sequence motifs for predicting proteinsubcellular localization. In German Conference on Bioinformatics (GCB), 2005.

[42] R. Homayouni, K. Heinrich, L. Wei, and M. W. Berry. Gene clustering by latent semantic indexing of medline abstracts. Bioinformatics, 21:104–115, 2005.

[43] J. Hosaka, J. Koh, and A. Konagaya. Effect of utilizing terminology on extraction of protein-protein interaction information from biomedical literature. In Proceedings of the 10th Conference of the European Chapter of the Association for Computer Linguistics (EACL'03), page 107C110, Budapest, Hungary, 2003. ACL, East Stroudsburg, PA.

[44] D. Hristovski, B. Peterlin, J. A. Mitchell, and S. M. Humphrey. Improving literature based discovery support by genetic knowledge integration. Stud. Health Technol. Inform., 95:68–73, 2003.

[45] D. Hristovski, J. Stare, B. Peterlin, and S. Dzeroski. Supporting discovery in medicine by association rule mining in medline and umls. Medinfo., 10(Pt 2):1344–1348, 2001.

[46] K. Humphreys, G. Demetriou, and R. Gaizauskas. Two applications of information extraction to biological science journal articles. In Pacific Symposium on Biocomputing 5, pages 502–513, 2000.

[47] T. K. Jenssen, A. Laegreid, J. Komorowski, and E. Hovig. A literature network of human genes for high-throughput analysis of gene expression. Nature Genetics, 28:21–28, May 2001.

[48] Y. S. Hwang K. J. Lee and H. C. Rim. Two-phase biomedical ne recognition based on svms. In Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, pages 33–40, 2003.

[49] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In Proc. ACL-2004 (poster session), pages 178–181, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[50] J. Kazama, T. Makino, Y. Ohta, and J. Tsujii. Tuning support vector machines for biomedical named entity recognition. In Proceedings of the Natural Language Processing in the Biomedical Domain (ACL 2002), Philadelphia, PA, USA., 2002.

[51] J. D. Kim and J. Tsujii. Corpus-based approach to biological entity recognition. In Proceedings of the Second Meeting of the Special Interest Group on Text Data Mining of ISMB, 2002.

[52] M. Krauthammer, P. Kra, I. Iossifov, S. M. Gomeze, g. Hripcsak, and V. Hatzivassiloglou. Of truth and pathways: Chasing bits of information through myriads of articles. Bioinformatics, 18(Supplement 1):S249–S257, 2002.

[53] T. Kudo and Y. Matsumoto. A boosting algorithm for classification of semi-structured text. In Proceedings of Empirical Methods of Natural Language Processing (EMNLP 2004), 2004.

[54] S. Kulick, A. Bies, M. Libeman, M. Mandel, R. McDonald, M. Palmer, A. Schein, and L. Unga. Integrated annotation for biomedical information extraction. In HLT/NAACL, pages 61–68, Boston, May 2004.

[55] T. Landauer, P. W. Foltz, and D. Laham. Introduction to latent semantic analysis. Discourse Processes, 25:259–284, 1998.

[56] S. Lappin and H.J. Leass. An algorithm for pronominal anaphora resolution. Computational Linguistics, 20(4):535–561, 1994.

[57] M. Lease and E. Charniak. Parsing biomedical literature. In IJCNLP-05, 2005.

[58] Lemur. Language Modeling Toolkit 4.1. http://www.lemurproject.org/lemur/doc.html, 2005.

[59] G. Leroy and H. Chen. Filling preposition-based templates to capture information from medical abstracts. In Proceedings of the Pacific Symposium on Biocomputing '02 (PSB'02), page 350C361, Lihue, HI, 2002.

[60] G. Leroy and H. Chen. Genescene: An ontology-enhanced integration of linguistic and co-occurrence based relations in biomedical texts. In JASIST 2005 Special Issue on Bioinformatics, 2005.

[61] G. Leroy, H. Chen, and J. D. Martinez. A shallow parser based on closed-class words to capture relations in biomedical text. Journal of Biomedical Informatics, 36:145–158, 2003.

[62] G. Leroy, H. Chen, J. D. Martinez, S. Eggers, R. Falsey, K. Kislin, Z. Huang, J. Li, J. Xu, D. McDonald, and G. Ng. Genescene: Biomedical text and data mining. In Proceedings of the third ACM/IEEE-CS joint conference on Digital libraries, pages 116 – 118, Houston, Texas, 2003.

[63] L. C. Liu, H. Sako, and H. Fujikawa. Performance evaluation of pattern classifiers for handwritten character recognition. Number 191-204. International Journal on Document Analysis and Recognition, 2002.

[64] Y. Liu and A. Sarkar. Using ltag-based features for semantic role labeling. In Proceedings of the Eighth Workshop on Tree Adjoining Grammars and Related Formalisms: TAG+8, Poster Track, Sydney, Australia, July 2006.

[65] Y. Liu, Z. Shi, and A. Sarkar. Exploiting rich syntactic information for relation extraction from biomedical articles. In HLT/NAACL-07, poster track, Rochester, NY, April 2007.

[66] Z. Lu and L. Hunter. Go molecular function terms are predictive of subcellular localization. In Proceedings of the Pacific Symposium on Biocomputing (PSB), volume 10, pages 151–161, 2005.

[67] R. Mack, S. Mukherjea, A. Soffer, N. Uramoto, A. Coden E. Brown, J. Cooper, A. Inokuchia, B. Iyer, Y. Mass, H. Matsuzawa, and L. V. Subramaniam. Text analytics for life science using the unstructured information management architecture. IBM Systems Journal, 34(3):490–515, 2004.

[68] E. Marcotte, I. Xenarios, and D. Eisenberg. Mining literature for protein-protein interactions. Bioinformatics, 17(4):359C363, 2001.

[69] D.M. Mcdonald, H. Chen, and H Su. Extracting gene pathway relations using a hybrid grammar:the arizona relation parser. Bioinformatics, 20(18):3370–8, Dec 2004.

[70] R. McDonald, F. Pereira, S. Kulick, S. Winters, Y. Jin, and P. Whit. Simple algorithms for complex relation extraction with applications to biomedical ie. In 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), 2005.

[71] S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. A novel use of statistical parsing to extract information from text. Proc. NAACL-2000, pages 226–233, 2000.

[72] S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. A novel use of statistical parsing to extract information from text. In The 6th Applied Natural Language Processing Conference, pages 226–233, 2000.

[73] J. H. Moore and S. M. Williams. New strategies for identifying gene-gene interactions in hypertension. Ann. Med, 34:88–95, 2002.

[74] A. Morgan, L. Hirschman, A. Yeh, and M. Colosimo. Gene name extraction using flybase resources. In Sophia Ananiadou and Jun'ichi Tsujii, editors, Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, pages 1–8, 2003.

[75] A. Moschitti. A study on convolution kernels for shallow semantic parsing. In Proceedings of ACL-2004, 2004.

[76] R. Nair and B. Rost. Inferring subcellular localization through automated lexial analysis. In Bioinformatics, volume 18, pages 78–86, 2002.

[77] R. Nair and B. Rost. Mimicking cellular sorting improves prediction of subcellular localization. In Journal of Molecular Biology, volume 34, pages 85–100, 2005.

[78] S. Ng and M. Wong. Toward routine automatic pathway discovery from on-line scientific text abstracts. Genome Informatics, 10:104–112, 1999.

[79] C. Nobata, N. Collier, and J. Tsujii. Automatic term identification and classification in biology texts. In Proceedings of the fifth Natural Language Processing Pacific Rim Symposium (NLPRS), pages 369–374, Beijing, China, 1999.

[80] S. Novichkova, S. Egorov, and N. Daraselia. Medscan: a natural language processing engine for medline abstracts. Bioinformatics, 19(13):1699–1706, 2003.

[81] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. Automatic extraction of information on protein-protein interaction from scientific literature. Genome Informatics, Universal Academy Press, pages 296–297, 1999.

[82] Jong C. Park, Hyun Sook Kim, and Jung Jae Kim. Bidirectional incremental parsing for automatic pathway identification with combinatory categorial grammar. Pac Symp Biocomput, 6:296–407, 2001.

[83] K. J. Park and M. Kanehisa. Prediction of protein subcellular location by support vector machines using compositions of amino acids and amino acid pairs. Bioinformatics, 19:16561663, 2003.

[84] C. Perez-Iratxeta, P. Bork, and M. A. Andrade. Xplormed: a tool for exploring medline abstracts. TRENDS in Biochemical Sciences, 26(9):573–575, 2001.

[85] C. Perez-Iratxeta, P. Bork, and M. A. Andrade. Association of genes to genetically inherited diseases using data mining. Nature Genetics, 31:316–319, July 2002.

[86] C. Perez-Iratxeta, A. J. Perez, P. Bork, and M. A. Andrade. Update on xplormed: a web server for exploring scientific literature. Nucleic Acids Research,, 31(13):3866–3868, 2003.

[87] C. Plake, J. Hakenberg, and U. Leser. Optimizing syntax-patterns for discovering protein-protein-interactions. In Proc ACM Symposium on Applied Computing, SAC, Bioinformatics Track, Santa Fe, USA, March 2005.

[88] D. Proux, F. Rechenmann, and L. Julliard. A pragmatic information extraction strategy for gathering data on genetic interactions. In Proceedings of International Conference on Intelligent System of Molecular Biology, pages 279–285, 2000.

[89] J. Pustejovsky, J. Castano, J. Zhang, M. Kotecki, and B. Cochran. Robust relational parsing over biomedical literature: Extracting inhibit relations. In Proceedings of the Pacific Symposium on Biocomputing, pages 362–373, 2002.

[90] J. Pustejovsky, J. Castano, J. Zhang, R. Saur, and W. Luo. Medstract: Creating large-scale information servers for biomedical library. In ACL02. Association for Computational Linguistics, 2002.

[91] Soumya Ray and Mark Craven. Representing sentence structure in hidden markov models for information extraction. In IJCAI, 2001.

[92] F Rinaldi, G Schneider, K Kaljurand, J Dowdall, C Andronis, and A Persidis. Mining relations in the genia corpus. In Proceedings of the Second European Workshop on Data Mining and Text Mining for Bioinformatics, pages 61–68, 2004.

[93] T. Rindflesch, J. Rajah, and L. Hunter. Extracting molecular binding relationships from biomedical text. In Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-NAACL 2000), page 188C195, Seattle, WA, ACL, East Stroudsburg, PA, 2000.

[94] T. C. Rindflesch, L. Tanabe, J. N. Weinstein, and L. Hunter. Edgar: Extraction of drugs, genes and relations from the biomedical literature. In In Proc. 5th Pacific Symposium on Biocomputing, pages 514–525, 2000.

[95] A. Rzhetsky, I. Iossifov, T. Koike, M. Krauthammer, P. Kra, M. Morris, H. Yu, P. A. Dubou, W. Weng, W. J. Wilbur, V. Hatzivassiloglou, and C. Friedman. Geneways: a system for extracting, analyzing, visualizing,and integrating molecular pathway data. J. of Biomedical Informatics, 37(1):43–53, 2004.

[96] G. Salton and C. Buckley. Term eighting approaches in automatic text retrieval. Information Processing and Management, 1988.

[97] T. Sekimizu, H.S. Park, and J. Tsujii. Identifying the interaction between genes and gene products based on frequently seen verbs in medline abstracts. In Genome Informatics, volume Examples of sentences. Shallow parser. Noun phrase recognizer. Algorithm to identify the arguments of verbs. 62-71, 1998.

[98] H. Shatkay, S. Edwards, and M Boguski. Information retrieval meets gene analysis. Information Retrieval Meets Gene, 17(2):45–53, 2002.

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

[100] D. Shen, J. Zhang, G. Zhou, J. Su, and C. L. Tan. Effective adaptation of hidden markov model-based named entity recognizer for biomedical domain. In Sophia Ananiadou and Jun'ichi Tsujii, editors, Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, pages 49–56, 2003.

[101] D. Sleator and D. Temperley. Parsing english with a link grammar. In Third International Workshop on Parsing Technologies, 1993.

[102] R. K. Srihari, W. Li, C. Niu, and T. Cornell. Infoxtract: A customizable intermediate level information extraction engine. In HLT-HLT-NAACL 2003 Workshop: Software Engineering and Architecture of Language Technology Systems, pages 51–58, Edmonton, Canada, May-June 2003.

[103] R. K. Srihari, M. E. Ruiz, and M. Srikanth. Concept chain graphs: A hybrid ir framework for biomedical text mining. In Workshop on Text Analysis and Search for Bioinformatics, SIGIR'03, Toronto, Canada, 2003.

[104] B. J. Stapley and G. Benoit. Biobibliometrics: Information retrieval and visualization from co-occurrences of gene names in medline abstracts. In Proceedings of the fifth Pacific Symposium on Biocomputing (PSB 2000), pages 529–40, 2000.

[105] B. J. Stapley, L. A. Kelley, and M. J. Sternberg. Predicting the sub-cellular location of proteins from text using support vector machines. In Pac. Symp. on Biocomp., pages 374–385, 2002.

[106] M. Stephens, M. Palakal, S. Mukhopadhyaya, R. Raje, and J. Mostafa. Detecting gene relations from medline abstracts. In Pac Symp Biocomput, page 483C495, 2001.

[107] M. Strube and U. Hahn. Functional centering: Grounding referential coherence in information structure. Computational Linguistics, 25(3):309–344, 1999.

[108] Chang. J. T. and R. B. Altman. Extracting and characterizing gene-drug relationships from the literature. Pharmacogenetics, 14(9):577–586, Sept. 2004.

[109] T. Takahata, Y. Kouchi, K. Asano, and T. Takagi. Disease-associated genes extraction from literature database. Genome Informatics, 14:703–704, 2003.

[110] K. Takeuchi and N. Collier. Bio-medical entity extraction using support vector machines. In Sophia Ananiadou and Jun'ichi Tsujii, editors, Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, pages 57–64, 2003.

[111] L. Tanabe, U. Scherf, L. H. Smith, J. K. Lee, L. Hunter, and J. N. Weinstein. Medminer: an internet text-mining tool for biomedical information, with application to gene expression profiling. BioTechniques, 27:1210–1217, 1999.

[112] J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Carroll. Automatic extraction of protein interactions from scientific abstracts. In Pac Symp Biocomput., pages 541–552, 2000.

[113] TREC. Text Retrieval Conference. http://trec.nist.gov/overview.html, 2005.

[114] Y. Tsuruoka and J. Tsujii. Boosting precision and recall of dictionary-based protein name recognition. In Sophia Ananiadou and Jun'ichi Tsujii, editors, Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, pages 41–48, 2003.

[115] H. D. Turtle. Inference Networks for Document Retrieval. PhD thesis, University of Massachusetts, 1990.

[116] Ellen M. Voorhees. Natural language processing and information retrieval. In SCIE, pages 32–48, 1999.

[117] A. Voutilainen and P. Tapanainen. Ambiguity resolution in a reductionistic parser. Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics, pages 394–403, 1993.

[118] L. Wong. Pies, a protein interaction extraction system. In Proceedings of the sixth Pacific Symposium on Biocomputing (PSB 2001), pages 520–531, 2001.

[119] A. Yakushiji, Y. Tateisi Y. Miyao, and J. Tsujii. Event extraction from biomedical papers using a full parser. In Proceedings of the sixth Pacific Symposium on Biocomputing (PSB 2001), pages 408–419, 2001.

[120] A. Yeh. More accurate tests for the statistical significance of result differences. In Proceedings of COLING 2000, pages 947–953, 2000.

[121] A. Yeh, L. Hirschman, and A. Morgan. Background and overview for kdd cup 2002 task 1: Information extraction from biomedical articles. Technical report, The MITRE Corporation, Dec 2002.

[122] M. Zhang, J. Zhang, J. Su, and G.D. Zhou. A Composite Kernel to Extract Relations between Entities with Both Flat and Structured Features. In Proc. ACL-2006, Sydney, Australia, July 2006.