DISCOVERING HIGHER ORDER RELATIONS FROM BIOMEDICAL TEXT

(Thesis format: Monograph)

by

Mohammad Syeed Ibn Faiz

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

THE UNIVERSITY OF WESTERN ONTARIO

School of Graduate and Postdoctoral Studies

CERTIFICATE OF EXAMINATION

Examiners:

.....

Dr. Sylvia Osborn

Supervisor:

.....

Dr. Robert E. Mercer

.....

Dr. Mark Daley

.....

Dr. Jacques Lamarche

The thesis by

Mohammad Syeed Ibn Faiz

entitled:

Discovering Higher Order Relations from Biomedical Text

is accepted in partial fulfillment of the

requirements for the degree of

Master of Science

.....

Date

.....

Chair of the Thesis Examination Board

Acknowlegements

I would like to take this opportunity to express my deepest gratitude to my supervisor Dr. Robert Mercer. He has been the constant source of help and encouragement for me. He was always there to give me every kind of support. I feel honoured that I have had the opportunity to work with him. He is one of the nicest persons I have ever met.

I would like to thank Emily Pitler and Ben Wellner for fruitful interactions while answering my queries.

I would like to thank Rushdi Shams and Sifta Ansari, my wonderful labmates, for their help and support.

I would also like to thank the University of Western Ontario, the Faculty of Science and the Computer Science Department for providing me financial support as a Teaching Assistant and a Research Assistant.

Finally, but foremost, I would like to thank my family. I would not be able to come this far without the support of my loving wife, Shabnam. And, I am forever thankful to my parents.

Abstract

A discourse relation is a coherence relation that connects discourse segments expressing abstract objects, i.e., events, facts, states or propositions. A biomedical relation, on the other hand, exhibits a relationship between biomedical entities. When the abstract objects involved in a discourse relation in a biomedical text correspond to biomedical relations (or biomedical events, facts, etc.), we can infer a higher order relationship between those biomedical relations. In this thesis, our goal is to extract such higher order relations from biomedical research articles. These higher order relations can be used for question answering, knowledge discovery or understanding reasoning in biomedical text. We have developed systems for parsing explicit discourse relations and extracting biomedical relations from biomedical research articles. We have evaluated these systems using public benchmark corpora and obtained promising results. Finally, we have presented an algorithm that can extract higher order relations leveraging the discourse relation parser and the relation extractor.

Keywords: Discourse, Relation Extraction, Biomedical Text

Contents

Certificate of Examination ii

Acknowlegements iii

Abstract iv

List of Figures viii

List of Tables ix

1 Introduction 1

2 Literature Review 4

2.1 Discourse Relations 5

2.1.1 The Rhetorical Structure Theory Discourse Treebank 5

2.1.2 The Penn Discourse Treebank 7

2.1.2.1 Discourse Parsing Using the PDTB 10

2.1.2.2 Identifying Explicit Discourse Connectives 10

2.1.2.3 Identifying Arguments of Discourse Connectives 12

2.1.2.4 Identifying Senses of Discourse Connectives 14

2.1.3 The Biomedical Discourse RelationBank 16

2.2 Biomedical Relation Extraction 18

2.3 Natural Language Processing Tools 23

3 Identifying Explicit Discourse Connectives 25

3.1 Features 26

3.1.1 Syntactic features 26

3.1.2 Surface Level Features . . . 29
3.2 Evaluation . . . 30
3.3 Discussion . . . 35

4 Identifying the Arguments of Explicit Discourse Connectives 38
4.1 Candidate Argument Head Selection . . . 40
4.2 ML Models and Features . . . 41
4.3 Evaluation . . . 44
4.4 Discussion . . . 45

5 Identifying the Senses of Explicit Discourse Connectives 47
5.1 Features . . . 51
5.2 Evaluation . . . 52
5.3 Discussion . . . 55

6 Biomedical Relation Extraction 57
6.1 Protein-Protein Interaction (PPI) Corpora . . . 58
6.2 Rule-based Relation Extraction . . . 60
6.2.1 Rules . . . 60
6.2.2 Evaluation . . . 65
6.2.3 Discussion . . . 66
6.3 Machine Learning-based Relation Extraction . . . 68
6.3.1 Features . . . 69
6.3.2 Evaluation . . . 71
6.3.3 Discussion . . . 71

7 Extracting Higher Order Relations from Text 74
7.1 Higher Order Relations . . . 74
7.2 Extracting Higher Order Relations . . . 77
7.3 Evaluation . . . 80
7.4 Discussion . . . 83

8 Conclusions and Future Work **86**

8.1 Contributions 86

8.2 Future Work 88

Bibliography **92**

A Relation Terms **97**

A.1 Relation Terms for Protein-Protein Interaction 97

B Connective Category **101**

B.1 Category for Discourse Connectives in BioDRB 101

C Higher Order Relations **108**

C.1 Examples of Higher Order Relations 108

D Abbreviations **113**

D.1 List of Abbreviations 113

E Stanford Typed Dependencies **115**

E.1 Stanford Typed Dependencies 115

Curriculum Vitae **119**

List of Figures

2.1 Discourse Structure in RST 6
2.2 A dependency tree and its corresponding chunk dependency tree 19

3.1 A syntax tree showing discourse and non-discourse usage of *and* 27

4.1 Dependency structure 44

5.1 Hierarchy of the sense tags in the PDTB. This figure is reproduced from [27]. . 48

6.1 Dependency representation for sentence (6.1) 61
6.2 Dependency representation for sentence (6.3) 64
6.3 Dependency representation for sentence (6.2). 64
6.4 Error in dependency representation 67
6.5 Dependency representation for a sentence from the AIMed corpus. 69

7.1 Dependency Representation for sentence (7.1). 79
7.2 Graphical representation of a higher order relation. 80

List of Tables

2.1 Performance (F-score) of the sentence level discourse parser with human-level accuracy for syntactic trees and discourse boundaries on RST-DT 7
2.2 Results of discourse versus non-discourse usage reported in [26] 11
2.3 Results achieved for connective identification by [17] 12
2.4 Results of argument identification reported in [43] 13
2.5 Argument identification accuracy reported in [11] 14
2.6 Results (F-scores) for sense classification reported in [42] 15
2.7 Performance of different models for identifying discourse connectives reported in [32] 17
2.8 Accuracies of PDTB-BioDRB and BioDRB. This table is reproduced from [32]. 17
2.9 Results on the AIMed dataset reported in [14] 21
2.10 Results of the 10-fold abstract-wise CV on five PPI corpora reported in [2] . . . 23

3.1 Distribution of positive and negative examples in the PDTB 31
3.2 Comparison of results on the PDTB with different feature sets 33
3.3 Results of the evaluation of the feature set on the BioDRB 34
3.4 Connective type-wise results on the PDTB 36

4.1 Feature set for discourse argument identification 43
4.2 Comparison of results for argument identification on the PDTB 45

5.1 BioDRB sense classification. This table is reproduced from [28]. 50
5.2 Results of sense classification on the PDTB 53
5.3 Results of sense classification on the PDTB considering only the first sense to be correct. 53

5.4 Grouping of BioDRB sense types into PDTB generalized classes. This table is reproduced from [28] 54
5.5 Results of sense classification on the BioDRB 54
5.6 Confusion matrix for sense classification on the PDTB 56

6.1 Summary of the five PPI corpora. This table is reproduced from [29]. 59
6.2 Statistics of positive and negative instances in the PPI corpora. 65
6.3 Results of PPI extraction on five corpora 65
6.4 Performance of RelEx reproduction on five corpora as reported in [29]. 66
6.5 Performance of the PPI extraction algorithm reported in [2]. 66
6.6 Performance of the binary maximum entropy classifier on the PPI corpora. . . . 72
6.7 Results of the hybrid PPI extraction method reported in [2]. 72

B.1 Classification of the discourse connectives in BioDRB into categories. 107

D.1 List of abbreviations 114

E.1 Stanford typed dependencies 118

Chapter 1

Introduction

Biomedical relation extraction is a well-known text mining problem. The goal of this problem is to extract relationships between biomedical entities from text. In this thesis we also aim at extracting a kind of relation from text that we call *higher order relations*. A higher order relation connects two biomedical relations or biomedical observations (e.g., a property of an entity). This kind of relationship exists at a higher level than each of the elements that it connects, hence our terminology. To extract the higher order relation, an understanding of the linguistic devices that provide cohesiveness to the text is required. We focus in particular on *explicit discourse connectives*. To our knowledge, we are the first to combine lower-level biomedical relations and observations with explicit discourse connectives to mine higher order relations from biomedical texts.

To illustrate this concept in more detail let's consider a hypothetical sentence: "ProteinX is likely to interact with ProteinY, because we previously observed that it interacts with ProteinZ". From this sentence we can easily recognize two biomedical relations: one between ProteinX and ProteinY and another between ProteinX and ProteinZ. Closer observation reveals that these two biomedical relations do not appear in isolation in the sentence. They in fact appear in two discourse segments which are explicitly connected by the discourse connective *because*. Because of this discourse level connection, we can infer a causal relationship between the two biomedical relations residing in the sentence shown above. We call such a relationship a higher order relation. Basically, we have exploited the idea that a discourse level connection can in fact connect two pieces of biomedical information together.

Extracting higher order relations will give us more information about biomedical relations in a similar way that extracting biomedical relations gives us more information about biomedical entities. We can use our knowledge of biomedical relations to answer queries regarding biomedical entities. For example, from the hypothetical example we can provide an answer to the query "Does ProteinX interact with ProteinY?". In a similar manner, knowledge about higher order relations will enable us to answer to queries like "Why does ProteinX interact with ProteinY?" or "Is there evidence that ProteinX interacts with ProteinY?".

Another application of higher order relations involving biomedical relations would be knowledge discovery. For example, if we mine a large number of biomedical articles and extract higher order relations from them, we would be able to form a large set of such relations. From that set we can infer complex relationships between biomedical entities. Higher order relations that involve biomedical facts or observations can also be very useful. A higher order relation often expresses part of a logical argument or reasoning from the author of the text. Understanding and combining such relations can assist in automatically understanding the reasoning or arguments presented by the text.

From the hypothetical sentence that we have shown above, it is possible to automatically extract a higher order relation. But to do that, we must be able to perform two other automatic extractions: parsing discourse relations (finding discourse connectives, their sense and their arguments) and extracting biomedical relations from text. In this thesis we propose a method to extract higher order relations from text by combining discourse relations with biomedical relations. We have developed systems for parsing explicit discourse relations and extracting biomedical relations from text. These two systems were then integrated to extract higher order relations. Our evaluation of each of these systems has indicated very promising results.

More specifically, our contributions in this thesis are the following:

- Based on previous work, we have developed a system for parsing explicit discourse relations from text. Explicit discourse relation parsing involves three steps: identifying explicit discourse connectives, identifying their arguments and identifying their sense. We achieved new state-of-the-art results on identifying discourse connectives on the Penn Discourse Treebank. We have obtained promising results on argument and sense identification. We have developed a discourse relation parser explicitly for the biomedical

domain. To our knowledge, we are the first to do this. With this parser, we have achieved significantly better results on identifying discourse connectives on the Biomedical Discourse Relation Bank than that reported in the literature so far.

- We developed two systems for extracting biomedical relations, specifically protein-protein interactions (PPI). One is a rule-based approach and the other uses a machine learning method (logistic regression). We obtained promising results using our rule-based system on standard PPI corpora.

- We have introduced the concept of higher order relations. We have proposed an algorithm that extracts higher order relations from text using the systems just mentioned dealing with discourse and biomedical relations.

The remainder of the thesis is organized as follows: Chapter 2 will provide a literature review of the problems of discourse parsing and biomedical relation extraction plus a short introduction to the tools used in our research. Chapters 3, 4 and 5 are devoted to the three steps involved in discourse parsing. Chapter 6 will describe our work on biomedical relation extraction. Our algorithm for extracting higher order relations and its implementation is presented in Chapter 7. Finally, Chapter 8 summarizes and concludes the work presented in this thesis and discusses possible future work.

Chapter 2

Literature Review

This thesis develops an algorithm for building higher order relations that involve biomedical and discourse relations. To create a higher order relation, the algorithm must have access to the discourse and biomedical relations. A significant research literature exists studying methods to extract discourse relations from text. However, very few of them have considered discourse relations in the biomedical domain. We investigated the problem of discourse relation parsing with an aim of improving the state-of-the-art and developing a discourse relation parser for the biomedical domain. For extracting biomedical relations we improved on ideas previously proposed in the research literature.

This chapter presents a survey of the background information required to understand the scope of the problem of extracting discourse and biomedical relations from text. In Section 2.1 we will have a look at discourse relations, and methods to recognize in text those phrases that are discourse relations, specifically data-driven approaches to discourse relation parsing. In order to generate data-driven machine learned models, corpora annotated with the appropriate information are needed. Three discourse-annotated corpora, namely the Rhetorical Structure Theory Discourse Treebank, Penn Discourse Treebank and Biomedical Discourse Relation-Bank are discussed along with the relevant works on discourse parsing leveraging these corpora. Then Section 2.2 will provide the necessary background to understand the problem of extracting biomedical relationships from text. Finally, Section 2.3 will introduce the natural language processing tools we used in our implementation.

2.1 Discourse Relations

A text is not just a collection of some isolated utterances. It contains various discourse coherence relations between its segments, i.e., sentences, clauses or phrases, to produce a meaning at a higher level than its constituents. To understand a discourse we need to comprehend the discourse relations it contains. Understanding discourse relations can be useful in many Natural Language Processing (NLP[1]) applications including question answering, semantic interpretation of natural language, textual entailment, anaphora resolution, etc. Discourse relations have been studied in the NLP literature at different levels of granularity and there are several discourse theories which propose different views on discourse structures [18, 41]. In this thesis our focus is on low-level discourse relations. At the low level, discourse relations hold between abstract objects [1], i.e., events, facts, states or propositions. Moreover, our focus is restricted to data-driven approaches to discourse relation parsing.

Data-driven approaches to discourse relation parsing have been facilitated by the advent of publicly available discourse-annotated corpora. Among these corpora the most noteworthy are the Rhetorical Structure Theory Discourse Treebank and the Penn Discourse Treebank. In the following two subsections we will look at some noteworthy works on discourse parsing leveraging these corpora. We will then look at another corpus, the Biomedical Discourse RelationBank, a discourse-annotated corpus for the biomedical domain.

2.1.1 The Rhetorical Structure Theory Discourse Treebank

The Rhetorical Structure Theory Discourse Treebank (RST-DT) consists of manually annotated discourse structures of 385 articles of the Wall Street Journal within the framework of Rhetorical Structure Theory. Rhetorical Structure Theory (RST) is a theory of discourse structure originated by Mann and Thompson [18]. In RST a discourse is represented by a tree, where the leaves are elementary discourse units (edu)s, which are mostly clauses or clause-like constructs, and intermediate nodes correspond to contiguous text spans. An example of a discourse structure in RST is shown in Figure 2.1.

A discourse tree gives a hierarchical view of the discourse under consideration. At the

[1]The list of abbreviations used throughout this thesis is provided in Appendix D

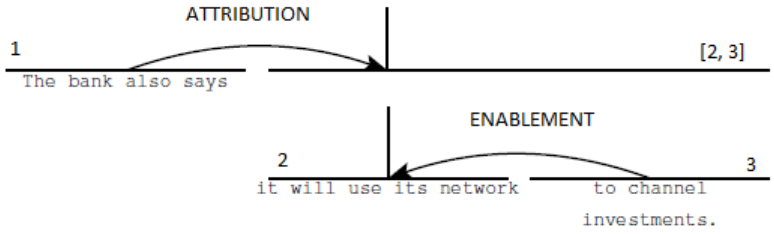

Figure 2.1: Discourse Structure in RST. A vertical line indicates the nuclear text span. ATTRIBUTION denotes an instance of reported speech, both direct and indirect. The satellite is the source of the attribution, and the nucleus is the content of the reported message. In an ENABLEMENT relation the nucleus denotes an action and the satellite provides information intended to aid the reader in performing that action.

top we have the whole text span and as we go down the tree we get smaller text spans being connected by rhetorical relations. Mann and Thompson proposed a carefully crafted set of 24 primary rhetorical relations in their original paper after analyzing a variety of texts including both dialogues and monologues. A rhetorical relation is mostly a binary relation (mononuclear relations hold between 2 text spans, whereas multi-nuclear relations hold between 2 or more text spans) having two argument text spans, one of which is called the nucleus and the other the satellite. The nucleus is the text span that is more essential to the writer's intended purpose and is marked with a vertical line in the graphical representation.

Soricut and Marcu [39] considered the problem of automatic sentence level discourse parsing using the RST-DT corpus. They introduced probabilistic models to segment a sentence into elementary discourse units (edu)s and to build a sentence level discourse parse tree. Their discourse segmentation model is a simple probabilistic model that estimates the segment probability $p(b|w,t)$, for each word w, where b is a Bernoulli random variable that indicates the probability that there is a segment boundary at the word w in the sentence with syntax tree t (a constituency-based parse tree that represents the syntactic structure of a sentence). To estimate this probability they used both lexical and syntactic features that capture the lexical and syntactic context around the word w. More specifically, for each word w, they picked the highest node N_w in the tree whose lexical head is w. They considered a feature composed of the siblings and

	T^-S^-	T^+S^-	T^-S^+	T^+S^+
Unlabeled	70.5	73.0	92.8	96.2
18 Labels	49.0	56.4	63.8	75.5
110 Labels	45.6	52.6	59.5	70.3

Table 2.1: Performance (F-score) of the sentence level discourse parser on the test set comprising 38 articles (991 sentences) with human-level accuracy for syntactic trees and discourse boundaries on RST-DT. T^+ and T^- denote gold-standard and automatic parses respectively. S^+ and S^- indicate gold-standard and automatic segmentation respectively. This table is reproduced from [39].

parent of N_w and estimated the probability of a segment boundary at w as the smoothed ratio of the number of times that feature occurred with a segment boundary at w over the total number of times it occurred. For discourse segmentation they attained 83.1% and 84.7% F-score using automatic and gold-standard parses, respectively. They built their discourse parser using a probabilistic model called the parsing model. That parsing model computes the conditional probability of a discourse candidate tree given a discourse-segmented lexicalized syntax tree (DS-LST). Their method extracts a dominance set from a given DS-LST and uses it as the feature in the probabilistic model. The dominance set consists of pair-wise dominance relations between the *edu*s. They reported their results, as in Table 2.1, using both gold-standard parses (T^+) and automatic parses(T^-), both gold-segmentation (S^+) and automatic segmentation (S^-) and considering all 110 rhetorical relations and a collapsed set of 18 rhetorical classes.

2.1.2 The Penn Discourse Treebank

The Penn Discourse Treebank (PDTB) [27] is the largest discourse-annotated corpus. It annotates the same Wall Street Journal articles as the Penn Treebank (PTB) and the PropBank, effectively aligning three different types of annotation (discourse, syntactic, and semantic, respectively). The PDTB takes a lexically-grounded and theory-neutral approach. It does not imply any particular structure for a complete discourse, instead, it annotates discourse relations only at a low level. At this level a discourse relation holds between two abstract objects, i.e. events, facts, states or propositions. Although the PDTB does not provide a tree or graph-like

discourse structure for a text, it may be possible to construct such a discourse structure from the low-level discourse relations. The PDTB annotates both explicit and implicit discourse relations. An explicit discourse relation is signaled by an explicit connective like *because, since, therefore,* etc. An implicit discourse relation, however, can exist without any such connective but still can be inferred by the reader. All discourse relations in the PDTB are binary. In an explicit relation, the connective takes exactly two arguments, ARG1 and ARG2. ARG2 appears in a clause which is syntactically bound to the connective and ARG1 is the other argument. The following is an example (from the PDTB) of an explicit discourse relation. Henceforth, in all examples the connective is underlined, ARG1 appears in italics and ARG2 is in bold.

> *The city's Campaign Finance Board has refused to pay Mr. Dinkins $95,142 in matching funds* <u>because</u> **his campaign records are incomplete**.

An explicit connective in the PDTB can be either a subordinating conjunction (e.g., *because*, *since*, *although*), a coordinating conjunction (e.g., *and*, *or*, *nor*) or a discourse adverbial (e.g., *however*, *otherwise*, *as a result*). There are 18,459 explicit discourse connective *tokens*[2] in the PDTB. Discourse connectives are often modified by adverbs such as *only*, *even*, *at least*, etc. The modified connectives like *only because*, *just when*, etc. are considered to belong to the same *type*[3] as that of their head (*because*, *when*, etc.). There are 100 unique connective *types* in the PDTB. Subordinating and coordinating discourse connectives take both arguments structurally, whereas discourse adverbials take ARG2 structurally but ARG1 can be anaphoric, i.e., it can occur anywhere in the discourse. In the following example, ARG1 for the discourse adverbial *nevertheless* is anaphoric.

> Mr. Robinson of Delta & Pine, the seed producer in Scott, Miss., said *Plant Genetic's success in creating genetically engineered male steriles doesn't automatically mean it would be simple to create hybrids in all crops*. That's because pollination, while easy in corn because the carrier is wind, is more complex and involves insects as carriers in crops such as cotton. "It's one thing to say you can sterilize, and another to then successfully pollinate the plant," he said. <u>Nevertheless</u>,

[2] A *token* is an occurrence of an explicit discourse connective.

[3] A connective *type* is an unmodified discourse connective such as *because* or *when*, i.e., a discourse connective in its base form.

> he said, **he is negotiating with Plant Genetic to acquire the technology to try breeding hybrid cotton**.

In most of the cases, the syntactic realization of an abstract object as the argument of a discourse connective is a clause, tensed or non-tensed as illustrated in the following examples.

> a) *A Chemical spokeswoman said the second-quarter charge was "not material"* **and that no personnel changes were made** as a result.
>
> b) Alan Smith, president of Marks & Spencer North America and Far East, says that Brooks Brothers' focus is to boost sales *by broadening its merchandise assortment* while **keeping its "traditional emphasis."**

The PDTB annotates implicit discourse relations to capture relations between abstract objects that are not explicitly realized in text. In the PDTB such implicit relations are annotated between adjacent sentences within the same paragraph. With each implicit relation, the connective that best expresses the inferred relations is also provided. The following shows an example of an implicit relation.

> Several leveraged funds don't want to cut the amount they borrow because it would slash the income they pay shareholders, fund officials said. But a few funds have taken other defensive steps. *Some have raised their cash positions to record levels*. [Implicit = because] **High cash positions help buffer a fund when the market falls**.

The PDTB also annotates the sense for both explicit and implicit connectives. The sense of a connective gives a semantic interpretation of the relation between its arguments. Multiple senses are given to connectives when appropriate. The tagset of senses form a hierarchy. At the top level, or *class level,* there are four sense tags representing the main semantic classes: "TEMPORAL", "CONTINGENCY", "COMPARISON" and "EXPANSION". For each *class level* sense a second level of *types* are defined to further refine the semantics of the class. For example, the "TEMPORAL" class has two *types* "Asynchronous" (when situations described in the arguments are temporally ordered) and "Synchronous" (when situations described in the arguments overlap). For each sense *type*, a third level of *subtypes* specifies the semantic

contribution of the arguments in the relation. For example, "Asynchronous" has two *subtypes* "precedence" (when the situation in ARG1 precedes the situation in ARG2) and "succession" (when the situation in ARG1 follows the situation described in ARG2). An example involving the "succession" subtype is shown below:

> No matter who owns PS of New Hampshire, after **it emerges from bankruptcy proceedings** *its rates will be among the highest in the nation*, he said. (TEMPORAL:Asynchronous:succession)

2.1.2.1 Discourse Parsing Using the PDTB

In this thesis our focus is on parsing explicit discourse relations in the style of the PDTB (low-level discourse relations). Parsing explicit discourse relations from text requires solutions to these three problems:

1. Identifying explicit discourse connectives
2. Identifying the arguments of the connective
3. Identifying the type or sense of each relation

In the following subsections we will go through each of these problems in detail.

2.1.2.2 Identifying Explicit Discourse Connectives

The challenge of identifying discourse connectives is to deal with ambiguity. There are 100 distinct discourse connective types in the PDTB. Only 11 of them appear as discourse connectives more than 90% of the time: *although, in turn, afterward, consequently, additionally, alternatively, whereas, on the contrary, if and when, lest* and *on one hand... on the other hand* [26]. So, most of the discourse connectives can appear as a non-discourse functional word and cause ambiguity in the identification process. The following examples illustrate discourse and non-discourse usage of the connective string *and*, respectively:

- It doesn't pay a dividend, *and* this trust needs income.
- My favorite colors are blue *and* green.

Features	Accuracy[4]	F-score
(1) Connective Only	85.86	75.33
(2) Syntax Only	92.25	88.19
(3) Connective + Syntax	95.04	92.28
(3) + Conn-Syn Interaction	95.99	93.63
(3) + Conn-Syn + Syn-Syn Interaction	96.26	94.19

Table 2.2: Results of discourse versus non-discourse usage reported in [26]. *Conn-Syn Interaction* denotes the pair-wise interaction features between the connective and all the syntactic features. Similarly, *Syn-Syn Interaction* indicates pair-wise interaction features between all the syntactic features. These results were obtained by doing a 10-fold cross-validation over the PDTB sections 2-22 using gold-standard parses.

Pitler and Nenkova [26] showed how syntactic features can be used to disambiguate discourse connectives. From the syntax tree their method finds the highest node that dominates all the words in the connective string and nothing else. This node is called the *self category*. The parent, and the left and right siblings of this *self category* node are then used as features along with the connective itself. They considered two properties of the right sibling node as features: whether the right sibling contains a VP (verb phrase) and/or a trace. They also experimented with feature combinations: pair-wise interaction features between connectives and each syntactic feature and interaction terms between pairs of syntactic features. They reported the best results, as in Table 2.2, on the task of disambiguating discourse versus non-discourse usage using a maximum entropy classifier by doing a 10-fold cross-validation over sections 2–22 of the PDTB.

Their results demonstrate that syntactic features can distinguish discourse versus non-discourse usage of connective strings quite well.

Wellner [42] reported two approaches to identify discourse connectives based on their syntactic context; one based on syntactic constituency structure and another on dependency structure. In the first approach he considered features derived from the constituent parse tree. These features include the connective feature, the path from the connective head word to the root,

[4]Accuracy is the fraction of instances (both positive and negative) which are correctly predicted.

	Accuracy	F-score
GS	97.34	95.76
Auto	96.02	93.62

Table 2.3: Results achieved for connective identification by [17]. *GS* and *Auto* indicates gold-standard and automatic parses respectively. These results were obtained by doing a 10-fold cross-validation over the PDTB sections 2-22.

some syntactic context features (whether an NP appears before a VP in the path, etc.) and some conjunctive features. In the dependency structure based approach he used features derived from the dependency representation. These features include the connective feature, contextual features (previous and next words and part-of-speech), features involving the parent and sibling of the connective head in the dependency tree and clause detection features (whether the parent and/or any sibling has a syntactic subject). He used sections 2–21 of the PDTB for training a binary maximum entropy classifier. Combining the two approaches he achieved precision[5], recall[6] and F-score[7] of 89.96%, 98.29% and 93.94% respectively on evaluation data consisting of PDTB sections 23–24.

Lin *et al.* [17] reproduced Pitler and Nenkova's work and improved over their reproduced result by adding some new features. These new features include the previous and next words and part-of-speech of the connective and their combinations. They also included as a feature the path from the connective to the syntactic root. They reported their results using both gold-standard (GS) and automatic parses by using 10-fold cross-validation over the PDTB sections 2–22. Table 2.3 shows the results they achieved with their enhanced feature set.

2.1.2.3 Identifying Arguments of Discourse Connectives

Each explicit discourse relation in the PDTB has exactly two arguments. ARG2 appears in a clause which is syntactically bound to the connective. Subordinating and coordinating conjunctions take ARG1 structurally but for discourse adverbials ARG1 can appear anaphorically. Identifying ARG1 is therefore more difficult than identifying ARG2. Wellner and Pustejovsky

[5]Precision is the fraction of the predictions that are accurate.

[6]Recall is the fraction of relevant instances that are predicted.

[7]F-score, or more precisely the F1-score is the equally weighted harmonic mean of precision and recall.

	ARG1	ARG2	Conn.
GS	76.4	95.4	74.2
Auto	69.8	90.8	64.6

Table 2.4: Results of argument identification reported in [43]. *GS* and *Auto* indicates gold-standard and automatic parses respectively. The *ARG1* and *ARG2* columns present the accuracy of identifying ARG1 and ARG2 respectively. The notion of accuracy used here is defined as the fraction of instances (only positive) that are correctly predicted [43]. The *Conn.* column shows the accuracy of identifying both arguments correctly.

[43] considered the problem of identifying the arguments of discourse connectives in text. Instead of identifying the full extent of the arguments they focused on identifying the heads of the arguments. They, therefore, reformulated the problem which now resembles that of predicate-argument identification, where predicates are discourse connectives and arguments are head words representing discourse segments. Since the PDTB annotation does not annotate the head of an argument they used a slight variation of the head finding rules in [8] to determine the heads. They trained two ranking models, one for identifying ARG2 and one for ARG1. To build those models they used a rich set of features derived from constituent parse trees, dependency trees, properties and lexico-syntactic context of connectives. Then they built a re-ranker which can identify both of the arguments simultaneously by utilizing features which capture the inter-dependence between the arguments. They reported accuracy, i.e., the percentage of arguments correctly identified, using both gold-standard (GS) and automatic (BLLIP re-ranking) parses on PDTB sections 23–24. Their reported accuracy using the re-ranking model is presented in Table 2.4. The *Conn.* column shows connective accuracy, which is the percentage of connectives for which both arguments were correctly identified.

Elwell and Baldridge [11] improved on Wellner and Pustejovsky's work by using models tuned to specific connectives and connective types (subordinating conjunctions, coordinating conjunctions and discourse adverbials). They also added new features leveraging morphological properties of connectives and their arguments, additional syntactic configurations and a wider context of preceding and following connectives. They observed that using specialized models improved performance on this task. However, since the general model has more data

	ARG1	ARG2	Conn.
GS	82.0	93.7	77.8
Auto (Bikel parses)	80.0	90.2	73.6

Table 2.5: Argument identification accuracy reported in [11]. *GS* and *Auto* indicates gold-standard and automatic parses respectively. The *ARG1* and *ARG2* columns present the accuracy of identifying ARG1 and ARG2 respectively. Accuracy is defined as the fraction of instances (only positive) that are correctly predicted. The *Conn.* column shows the accuracy of identifying both arguments correctly.

to be trained on, they used an interpolation of the general and specialized models to utilize the strength of both strategies. Following [43] they used sections 2–21 for training, sections 0 and 1 for development and sections 23–24 for testing. Interpolating connective specific, connective type specific and general models they achieved their best result which is shown in Table 2.5.

Their result shows that using specialized models they gained a 3.6%-age point improvement over Wellner and Pustejovsky's reranking model. The improvement is even more significant on auto-parsed data (9%-age points). A careful comparison with Table 2.4 reveals that performance improved specifically in identifying ARG1. This is due to the fact that identifying ARG1 becomes difficult mainly due to the unrestricted presence of ARG1s of discourse adverbials. Building a specialized model for them helps to capture their characteristics more accurately.

2.1.2.4 Identifying Senses of Discourse Connectives

Although most of the discourse connectives tend to have a single sense, there are a few that can be quite ambiguous in that they frequently occur with different senses. For example, *since* often signifies a Temporal relation, but also often indicates Contingency. Pitler and Nenkova [26] experimented with sense classification using almost the same set of features they used for connective identification. They did not, however, consider the finer-grained senses, i.e., *types* or *subtypes*, instead they performed classification among only the top four sense classes. In the PDTB some explicit relations are given a single sense while the rest are given two senses. In their work, both senses were considered to be correct. They reported that the connective itself is a good feature for identifying the sense class. In fact, they achieved an accuracy of

Coarse	Semi-coarse	Fine
92.40	84.18	76.59

Table 2.6: Results (F-scores) for sense classification reported in [42]. The column *Coarse* shows the F-score obtained on classification between the top four class level senses in the PDTB along with a fifth class OTHER (for EntRel[10]and NoRel[11]). The F-score obtained on classification between the fine-grained 42 categories (including OTHER) is shown in the *Fine* column. The result shown in the *Semi-coarse* column was obtained on a classification using a set of 13 semi-coarse-grained level senses which result from collapsing some of the fine-grained categories that occur quite infrequently.

93.67% by using only the connective phrase as the feature. After adding syntactic features and pair-wise interaction features between the connective and each syntactic feature the accuracy was raised to 94.15%. They, however, mentioned that assuming only the first sense to be correct would degrade their performance by about 1%-age point. It is worth mentioning that the inter-annotator agreement on sense class is 94% [27].

Wellner [42] reported a more elaborate experiment with sense classification. He conducted his experiments on sense classification at different levels of granularity that include coarse-grained level (top four classes in the hierarchy), fine-grained level (all senses at the bottom of the hierarchy) and a semi-coarse-grained level (collapsing some infrequent fine-grained senses). For each of these levels he considered the problem of assigning senses to both explicit and implicit connectives jointly as well as separately. To model the dependencies between adjacent discourse relations he modeled the problem with a linear-chain first-order conditional random field, where each discourse relation is an element in the sequence and the sequence is delimited by paragraph boundaries. Table 2.6 shows the F-scores he obtained for identifying the sense of explicit relations at each level of sense categories.

[11] An EntRel annotation is used in the PDTB for an adjacent sentence-pair where no discourse relation can be inferred and where the second sentence only serves to provide some further description of an entity in the first sentence

[11] A NoRel annotation is used in the PDTB for an adjacent sentence-pair where neither a discourse relation nor entity-based coherence can be inferred between the adjacent sentences

2.1.3 The Biomedical Discourse RelationBank

The Biomedical Discourse RelationBank (BioDRB) annotates a subset of the full text biomedical articles of the Genia corpus with discourse relations. It adopts the annotation guidelines of the PDTB and annotates both explicit and implicit relations. However, there are some differences between the annotation process of the BioDRB and the PDTB. There are many discourse connectives in the BioDRB which are not present in the PDTB. The annotators of the BioDRB were given the connective list of the PDTB but were encouraged to annotate new discourse connectives [28]. While the PDTB has 273 distinct explicit discourse connectives (including modified connectives), there are 178 unique explicit discourse connectives in the BioDRB. It was reported that only 44% of the discourse connectives in the BioDRB also occur in the PDTB [32]. The remaining 56% connectives include many frequent cue phrases in the biomedical domain - like "by", "due to", "in order to", etc., which may indicate domain-specific characteristics of the BioDRB. The BioDRB also differs from the PDTB in the way senses are grouped into a hierarchy. These senses in the BioDRB are organized into two levels, with the second *subtype* level refining the *types* at the top level. Examples of discourse relations annotated in the BioDRB are shown below:

> *IL-10 has been shown to block the antigen-specific T-cell cytkine response* by **inhibiting the CD28 signaling pathway**. (Purpse.Enablement)
>
> *The phosphorylation of signal transducer and activator of transcription 3 was sustained in both blood and synovial tissue CD4+ T cells of RA* , but **it was not augmented by the presence of 1 ng/ml IL-10**. (Contrast)

Ramesh and Yu [32] considered the problem of identifying discourse connectives from biomedical text. They built three classifiers: *pdtb* was trained and tested on the PDTB, *pdtb-biodrb* was trained on the PDTB and tested on the BioDRB, *biodrb* was trained and tested on the BioDRB. All of these classifiers were based on linear-chain first-order Conditional Random Field (CRF) models. A CRF is a probabilistic undirected graphical model used to label sequential data [40]. They trained their models using ABNAR [37], a biomedical named entity recognizer, with its default feature set, which includes standard bag-of-words, orthographic and n-gram features. They performed a 10-fold cross-validation to evaluate *pdtb*. To evaluate on BioDRB they used

	pdtb	*pdtb-biodrb*	*biodrb*
Precision	88±2	79±0.3	79±5
Recall	81±2	42±0.6	63±8
F1-score	84±1	55±0.5	69±5

Table 2.7: Performance of different models for identifying discourse connectives reported in [32]. The results in *pdtb* show the performance of a classifier which was trained and tested on the PDTB. *pdtb-biodrb* presents the results obtained by training a classifier on the PDTB and testing it on the BioDRB. Finally, the results obtained by doing a 12-fold cross-validation over the BioDRB are presented in the *biodrb* column.

		pdtb-biodrb	*biodrb*
BioDRB ∩ PDTB	44%	94.3	94.9
BioDRB ∉ PDTB	56%	89.6	92.7

Table 2.8: Accuracies of PDTB-BioDRB and BioDRB. This table is reproduced from [32]. BioDRB ∩ PDTB denotes connectives which are present in both the PDTB and BioDRB corpora, whereas BioDRB ∉ PDTB indicates the connectives which are present only in the BioDRB corpus. These results show that *biodrb* edged out *pdtb-biodrb* on recognizing discourse connectives that are present ony in the BioDRB.

12-fold cross-validation. Table 2.7 shows their result.

They also reported that while the accuracies for *pdtb-biodrb* and *biodrb* are the same on the 44% connectives that are common to both the PDTB and the BioDRB, on the remaining 56% of the connectives, *biodrb* gave higher accuracy as shown in Table 2.8:

Prasad *et al.* [28] reported a simple baseline for classifying the senses of explicit discourse connectives in the BioDRB. In their experiment they used the (case-insensitive) connective text string as the only feature and obtained an accuracy of 90.9% on identifying the first sense.

2.2 Biomedical Relation Extraction

Understanding relations between biomedical entities is important for biomedical knowledge discovery. Knowledge of biomedical relations can be used for discovering regulatory pathways, signal cascades, metabolic processes, disease models, etc. [13]. The goal of relation extraction is to find occurrences of relationships between biomedical entities. While the type of entity is usually very specific (gene, protein or drug), the relationship type can be very general (any biomedical association), or very specific (e.g., a regulatory relation) [7]. Several approaches to this task have been reported in the literature. Manually generated template-based methods use patterns generated by domain experts to look for relationships in text. Automatically generated template-based methods synthesize patterns from known relations and exploit them to extract relations from unseen text. Co-occurrence based methods infer relations based on the assumption that two entities are related if they frequently co-occur. Statistical methods build statistical probabilistic models from known data sets by capturing statistical properties of different aspects of biomedical relations of interest. Finally, rule-based methods apply manually curated rules that depend on various linguistic analyses to extract relations from text.

RelEx [13] is a rule-based biomedical relationship extraction system. For any given sentence RelEx generates a dependency parse tree and produces candidate relations by finding paths connecting pairs of genes or proteins. These candidate paths are chosen according to the following three rules which reflect the usual constructs in English that are used to describe relations.

1. effector-relation-effectee (e.g., ‘A activates B’)

2. relation-of-effectee-by-effector (e.g., ‘Activation of A by B’)

3. relation-between-effector-and-effectee (e.g., ‘Interaction between A and B’)

These rules are actually applied to a chunk dependency parse tree. This chunk dependency parse tree is produced from the dependency parse tree by collapsing all the nodes corresponding to a noun phrase chunk into a single node in the dependency tree. An example of a dependency tree and its corresponding chunk dependency tree is shown in Figure 2.2.

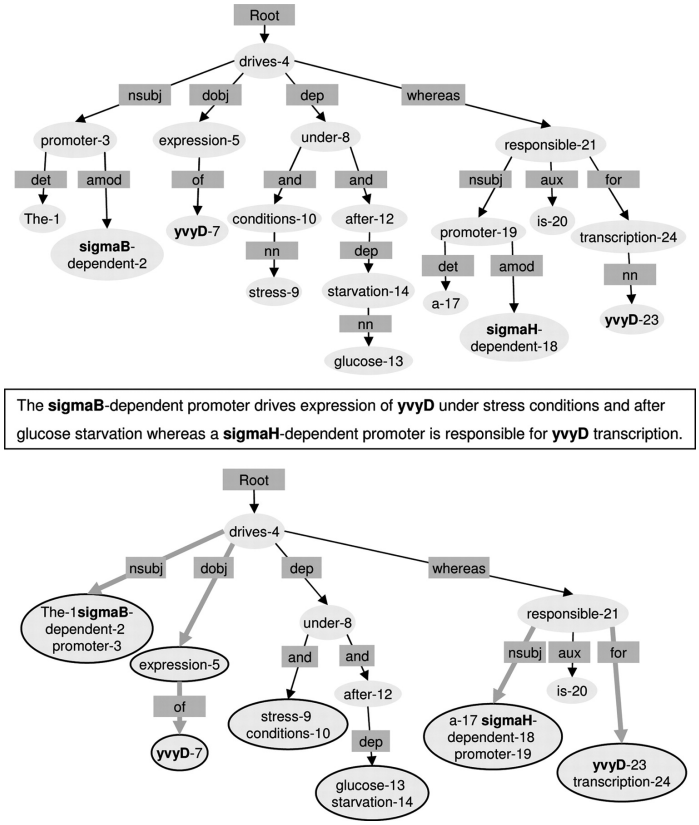

Figure 2.2: A dependency tree and its corresponding chunk dependency tree. Reproduced from [13]. Words marked in bold indicate gene/protein names, thick gray edges indicate paths that are extracted by Rule 1.

Rule 1 extracts candidate paths starting from noun phrase chunks and ending chunks containing names of genes or proteins. Chunks with an incoming subject-dependency edge are chosen as potential candidates. If there are no subject dependencies then all gene/protein name containing chunks are considered as potential start/end points. Not all candidate paths found by this rule contain a valid relation. Therefore, these candidate paths undergo a filtering stage where some filtering rules are applied to cancel out the spurious candidate paths. Rule 2 is divided into two sub-rules. Rule 2a extracts the longest paths from the dependency tree containing only noun phrase chunks and dependencies of the types of *of, by, to, on, for, in, through, will*. Only the paths where there is a dependency between two protein containing names are retained. Rule 2b is directly applied to chunk sentences. It extracts the longest sequences of chunks connected by either *of, by, to, or, for, in, through* or *with*. The sequences which contain

at least two of these terms and one of which occurs between two chunks each containing at least one protein are retained. Rule 3b extracts paths where two noun phrase chunks are connected by the dependency *between* and the second chunk is connected to another noun phrase chunk via an *and* dependency.

Fundel *et al.* evaluated RelEx on the LLL data set[12] [23], which was created by extracting Medline abstracts on Bacillus subtilis, where they achieved precision, recall and F-score of 68%, 83% and 75%, respectively, on the basic test set, which is better than the performance showed by the best official submission on the LLL-challenge (precision: 50%, recall: 53.8%, F-score: 54.3%) [23].

Katrenko and Adriaans [14] proposed a representation for learning relations based on dependency trees. They used information found in the predefined levels of the dependency tree, such as the local dependency context of the involved entities and the tree's roots. They divided all of their features into two groups: local and global context. A global context consists of a least common subsumer (LCS) and the root of the tree. The LCS of two nodes A and B in a dependency tree T is a node L, such that L is the ancestor for both, A and B, and there exists no other node N being an ancestor for A and B, such that L is an ancestor of N. They showed that the least common subsumer often includes words, such as *bind*, *interact*, *inhibit*, etc., which are important for relation learning in the biomedical domain. The local context consisted of a parent of a given node and its two children. The features of a parent and a child were lemmata of the corresponding words and the syntactic function between the node in question and a parent.

They used different machine learning classifiers (as shown in Table 2.9) and used two data sets to evaluate their feature set. One of the data sets was AIMed [3] and the other was LLL[23]. The AIMed data set was compiled from 225 Medline abstracts and contains manual annotation of protein-protein interactions. On the LLL data set they achieved an F-score of 58.5% by doing a 5-fold cross-validation with a BayesNet classifier. Table 2.9 shows the results they obtained on the AIMed dataset by doing 10-fold cross-validation with different machine learning classifiers.

Bui *et al.* [2] proposed a hybrid approach consisting of two phases to extract protein-protein

[12] http://genome.jouy.inra.fr/texte/LLLchallenge/

Method	Precision	Recall	F-score
Naive Bayes	71.5	57.6	63.8
BayesNet	68.9	64.5	66.6
IB3	81.3	51.6	63.1
IB1	77.4	66.3	71.4
Stacking	69.4	76.2	72.7
Bagging	68.2	63.7	65.8
AdaBoost	67	68.7	67.8

Table 2.9: Results on the AIMed dataset reported in [14]. These results were obtained by doing a 10-fold cross-validation over the AIMed dataset using different machine learning classifiers.

interactions (PPI) from biomedical literature. In the first phase they automatically divided the data into five groups depending on its semantic properties and extracted candidate PPI pairs from those groups. In the next phase they applied support vector machine (SVM) classifier with a default RBF[13] kernel to classify candidate PPI pairs using features specific for each class. The division of data into groups was done by applying a set of 5 rules. Those rules were manually crafted to capture the common patterns that normally occur in literature to present protein-protein interactions. Each rule is based on an abstract pattern involving a relation cue word (*bind*, *interact*, *inhibit*, etc.) along with two protein names. They created a list of relation cue words by combining the relation lists used in previous work by [6, 13, 24]. The following list shows the 5 abstract patterns.

- Form1: $PRO_i\ word^*$ REL (verb) $word^*\ PRO_j$
  - Example: PRO1 interacts with PRO2
- Form2: $PRO_i\ word^*$ REL (noun/verb) $word^*\ PRO_j$
  - Example: PRO1 has a weak interaction with PRO2.
- Form3: REL (noun) $word^*\ PRO_i\ word^*\ PRO_j$

[13] Radial Basis Function

  - Example: interaction between PRO1 and PRO2.

- Form4: PRO_i/PRO_{i+1} or $PRO_i PRO_{i+1}$ or PRO_i-PRO_{i+1}

  - Example: PRO1/PRO2 binding, PRO1-PRO2 compound.

- Form5: PRO_i $word^*$ PRO_j $word^*$ REL

  - PRO1, PRO2 interact; in PRO1, PRO2 complex.

Here, REL is a relation cue word and can be a noun or a verb, $word^*$ are tokens between PRO^* and REL. PRO_i and PRO_j are any protein pair with $j > i$. Based on these basic forms, they mapped the semantic relations of these forms into parse trees. Those mapped relations were then used to find candidate PPI pairs which were placed into 5 different groups based on the rules used to find them.

For the second phase they used support vector machine classifiers and a set of features which are combinations of some features that had been previously proposed by [6, 24, 22]. The feature set includes REL, the count of protein names and relation cue words in the sentence, distance (number of words/tokens) between PRO1-REL and REL-PRO2 (or between REL-PRO1 and PRO1-PRO2), the distance from the joint node connecting PRO1 and PRO2 in the parse tree, the paths connecting the joint node with PRO1 and PRO2 and two lexical features containing the list of tokens between PRO1, PRO2 and REL.

They used all of the five well known corpora for PPI which were converted into a unified format and are provided by Pyysalo *et al.* [29]: AIMed, BioInfer, HPRD50, IEPA and LLL. Two types of evaluation were performed, a single corpus test and a cross-corpora test. In the single corpus test they evaluated the performance by 10-fold abstract-wise cross-validation (CV), and use the one-answer-per-occurrence criterion [22]. Table 2.10 shows the performance they obtained on the five corpora.

[15] Where all occurrences of the same interaction pair in a document have to be identified.

Corpus	Precision	Recall	F-score
AIMed	55.3	68.5	61.2
BioInfer	61.7	57.5	60.0
HPRD50	70.2	77.9	73.8
IEPA	67.4	83.9	74.7
LLL	84.1	84.1	84.1

Table 2.10: Results of the 10-fold abstract-wise CV on five PPI corpora reported in [2]. The one-answer-per-occurrence criterion[15]was used for this evaluation.

2.3 Natural Language Processing Tools

As mentioned at the onset of this chapter, this thesis develops an algorithm for extracting higher order relations from text. The algorithm involves discourse relation parsing and biomedical relation extraction. We developed machine learning based systems relying on rich linguistic processing for solving these two problems. In our implementation we extensively used some existing tools and libraries for applying machine learning methods and doing linguistic processing. In this section we will introduce these tools.

Mallet is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning text applications [19]. It includes sophisticated tools for classification, sequence tagging, topic modelling and numerical optimization. Mallet is open source software. In this thesis, it is extensively used for developing all of our machine learning models. The advantage of using MALLET is that it is especially tuned for machine learning in the natural language processing domain. We used version 2.0.7.

BLLIP Reranking Parser is a statistial parser developed by Charniak and Johnson [5]. It is the Charniak parser [4] followed by a reranker stage. The reranker receives N (typically 50) best parses from the Charniak parser and reranks them to find the best parse. The default models provided with this parser were trained on newswire articles. McClosky and Charniak [20] produced biomedical models for this parser to use for parsing biomedical

text. The BLLIP reranking parser is also known as Charniak-Johnson Max-Ent reranking parser.

Stanford Parser is a statistical parser that generates the Stanford dependency representation as well as phrase structure trees. The Stanford Dependency representation was designed to provide a simple description of the grammatical relationships in a sentence [9]. Unlike the phrase structure representation, it represents all sentence relationships uniformly as typed dependency relations, that is, as triples of a relation between pairs of words, such as "the subject of *went* is *John*" in the sentence "John went to school.". We used this parser to produce the dependency representation of a sentence.

Chapter 3

Identifying Explicit Discourse Connectives

Parsing of an explicit discourse relation starts with identifying the discourse connective. Accurately identifying the discourse connectives is therefore very important for discourse relation parsing. In Chapter 2 we introduced some previous works [26, 42, 17] that considered the problem of discourse connective identification. We achieved improvements to the state-of-the-art for identifying discourse connectives using the Penn Discourse Treebank (PDTB). The improvements were achieved by drawing on and combining ideas from existing works as well as introducing some novel features.

There are 100 explicit discourse connective types in the PDTB. As mentioned in Section 2.1.2.2, most of these discourse connectives can assume a non-discourse usage in a sentence. Identifying discourse connectives using PDTB thus becomes a problem of disambiguating discourse versus non-discourse usage of the occurrences of the connective phrases. The following two sentences both contain an occurrence of the discourse connective *once*. However, *once* assumes a discourse usage only in the first sentence. In the later one it only acts as an adverb.

> Asbestos is harmful *once* it enters the lungs.
> Asbestos was *once* used in cigarette filters.

Like others [26, 42, 17], we have treated the problem of identifying discourse connectives as a machine learning (ML) problem. With an expanded set of features, we trained a discourse connective classifier using machine learning methods. The classifier decides whether an occurrence of a connective phrase is either a discourse usage or a non-discourse usage.

3.1 Features

In this section we will discuss the feature set we used for ML classification. The features in our feature set are a combination of some features that were previously proposed by [26, 42, 17] and some novel ones. We have grouped the features into two classes: syntactic features and surface level features.

3.1.1 Syntactic features

Features derived from a syntax tree can be used to disambiguate a discourse connective. In a syntax tree the path connecting the root of the tree to the connective node carries some of the most useful information for connective identification. For example, in the PDTB sections 2-22, the sections used to train the classifier, there are 2442 discourse usages of *and*. In 2395 of these cases the grand-parent of *and* in the syntax tree is a clausal node (S, SBAR, SBARQ, SINV, SQ).

Figure 3.1 shows a syntax tree for a sentence containing three occurrences of the connective phrase *and*. The second *and* (henceforth and_s) is a discourse connective but the first (or the third) *and* (henceforth and_f) is not. A careful look at this syntax tree reveals that and_f is inside a noun phrase (NP), which is a strong indication that it is not a discourse connective but rather a noun phrase coordinator instead. On the other hand, and_s is inside a clausal node (S). Moreover, both of its left and right siblings include clausal nodes implying that and_s is coordinating two clauses and therefore acting as a discourse connective.

Wellner [42] derived syntactic/constituent features solely from the path connecting the root to the connective. More specifically, he used the last few constituents of this path in different combinations and a collapsed version of the complete path, where adjacent constituents with identical labels are compressed into one. Pitler and Nenkova [26], on the other hand, did not use the complete path to derive features. Instead, they used syntactic context beyond this path through their *left sibling* and *right sibling* features. In our experiment we found that augmenting the syntactic features from [26] with features carrying information about the whole path is beneficial. However, it became evident that using all the constituents of that path individually is better than using the whole path as a single feature. We, therefore, used each individual con-

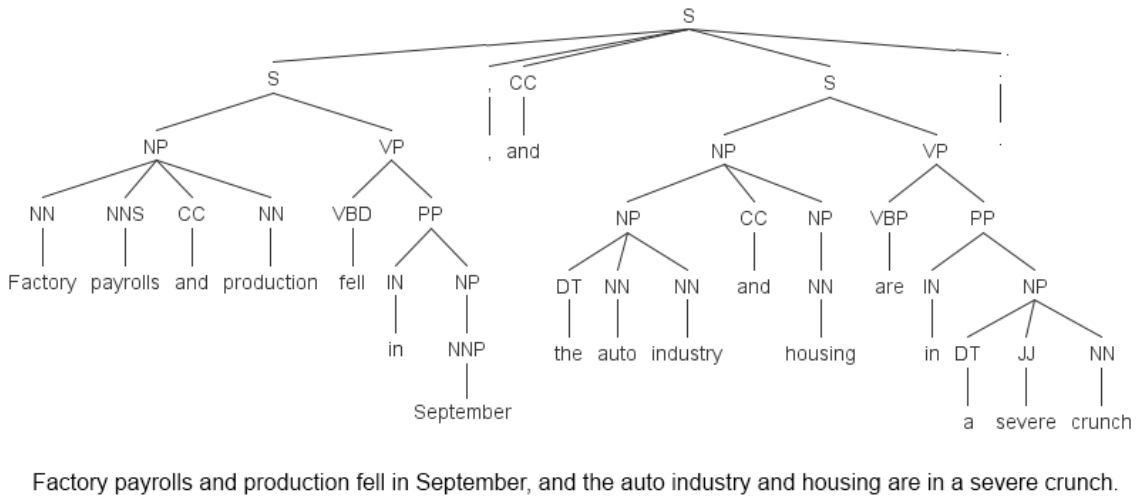

Figure 3.1: A syntax tree showing discourse and non-discourse usage of *and*. There are three occurrences of the connective string *and* in the sentence. Only the second *and* is a discourse connective.

stituent which is a predecessor to the *parent category* constituent, combined with its distance from the connective as a feature.

It was shown in [26] that including more features about the right sibling can improve the result. In our experiment on the development set we achieved an improved result by using the part-of-speech of the syntactic head of the *right sibling* as a feature. To find the syntactic head we used the head finding rules from Collins' PhD thesis [8].

Knott [15] did an extensive study of discourse connectives and their properties. He classified discourse connectives into the following categories based on their syntactic types: subordinating conjunctions, coordinating conjunctions, discourse adverbials, prepositional phrases and phrases taking sentence complements. Following [42] we used the category of a connective as a feature. We also collapsed the last three categories into one resulting in three categories instead of five. Moreover, we used the category of a connective in conjunction with the syntactic head of *the right sibling* as another feature.

The following is the list of all the syntactic features that we used:

Self Category The highest node in the syntax tree which is the ancestor of all the words in the

connective but no other words. For single word connectives this would be the POS tag of the word, but for multi-word connectives it would either be a phrasal or clausal category. For example, for the connective *in addition*, the *self category* would be a PP, though *in* is a preposition and *addition* is a noun.

Parent Category The immediate parent of the *self category*. For example, in Figure 3.1, for and_s this feature would be S, and for and_f it would be NP.

Left Sibling Category The syntactic category of the immediate left sibling of the *self category*. If the left sibling does not exist then this feature would be "None". For and_s this feature would be ",".

Right Sibling Category The syntactic category of the immediate right sibling of the *self category*. Again, for a non-existent right sibling this feature takes the value "None".

Right Sibling contains a VP Whether the right sibling contains a VP node underneath it. If a connective string has a discourse function then the right sibling will often be a clause (e.g., S, SBAR). In that case, there should be a verb phrase (VP) dominated by the right sibling. For example, for and_s and and_f this feature takes the value "true" and "false" respectively.

Ancestor-@*i* The syntactic category of the ancestor which is at a distance i from the parent category.

Head-POS POS tag of the head of the right sibling. For connective strings with discourse function the head of the right sibling will often be a verb.

Connective Category Category of a (potential) connective is either "Subordinator", "Coordinator" or "Adverbial".

Conjunctive Features Following [26], we used pair-wise interaction features between the first five syntactic features and pair-wise interaction features between the connective and each of the first five syntactic features. As mentioned above, we also used the combination of Head-POS and *connective category* as another feature.

3.1.2 Surface Level Features

Properties of the neighboring words of a discourse connective can sometimes signal its presence. Such surface level features were found to be useful for connective identification in [42, 17]. The features that they considered for a connective C with previous word *prev* and next word *next* include *prev*, *next*, part-of-speech (POS) of *prev*, POS of *next*, C + *prev*, C + *next*, C + *next* POS, C + *prev* POS, C POS + *prev* POS, C POS + *next* POS.

In our experiment we found that *prev* and *next* are good features especially when combined with the connective. For example, if a candidate connective string is followed by *of* or *to* (e.g. *as a result of*, *in addition to*) then it is not likely to be a discourse connective. Similarly when a candidate connective string *when* is preceded by text indicating either a year, month or week then *when* does not function as a discourse connective, rather it acts as a temporal modifier.

In our experiment we observed that the features derived from chunk tags are more useful than the POS features. Using the conjunction of the connectives and the chunk tags of the neighbouring words instead of the POS tags gave us a better result on the development set. For example, whenever *as* is immediately followed by a VP it is unlikely to be a discourse connective. However, *when* immediately followed by a VP may often be a discourse connective. Because approximating the chunk tags from the syntax tree gave us better results than using an automatic chunker (OpenNLP chunker), we approximated the chunk tag for a word by taking the label of its grand-parent in the constituent parse tree. We found that on PTB sections 0-2 there was 91% agreement between the chunker and our approximation.

The following is the list of all the surface level features we used:

- Connective phrase (C)
- *prev*
- *C + prev*
- *C + prev Chunk*
- *next*
- *C + next*

- *C + next Chunk*

3.2 Evaluation

According to the PDTB annotations, there are 18,459 tokens of the 100 explicit discourse connectives in the PDTB. The sentences containing these tokens constitute the set of positive training examples. The sentences containing occurrences of the connective strings in the text of the PDTB which are not annotated as discourse connectives are treated as negative training examples. Table 3.1 shows that there are twice as much negative examples as there are positive examples, globally and in each of the development (sections 0–1), training (sections 2–22) and testing sets (sections 23–24). We trained a binary maximum entropy (logistic regression) classifier using the features described above and the machine learning toolkit MALLET [19] to determine for each candidate phrase whether or not that candidate phrase is a discourse connective according to the PDTB annotation scheme. Logistic regression is a discriminative probabilistic classification model. A logistic regression classifier is also called a maximum entropy classifier because it is obtained by following the maximum entropy principle. In this thesis, we used the term *maximum entropy classifier* instead of the more accurate term *logistic regression classifier* because we have found that *maximum entropy classifier* is more commonly used in the NLP literature. The principle of maximum entropy states that the best probability model for the data is the one which maximizes entropy over the set of probability distributions that are consistent with the evidence [33]. By maximizing entropy, in other words maximizing uniformity, this model avoids any unnecessary assumptions and becomes the least committed model. Binary logistic regression models the conditional distribution of the output given the observation as follows:

$$p(y = 1|x) = \frac{1}{1 + exp(-\sum^{k} \lambda_i f_i(x, y))}$$

where x is an observation, y is the output (class), $f_i(x, y)$ is a binary valued feature function, the λ_is are the feature weights and k is the total number of features. The parameters of this model are computed by maximum likelihood estimations, i.e., finding the best fit to the training data.

Sections	Positive	Negative
0–1	1462	3404
2–22	15402	37327
23–24	1595	3246

Table 3.1: Distribution of positive and negative examples in the PDTB

We also experimented with a Naïve Bayes classifier and the AdaBoost ensemble method with decision tree classifiers. However, we found that the logistic regression classifier gave us the best result. Most of the features we discussed above are in fact feature templates and can generate a large number of binary features. To minimize data sparsity and to reduce the risk of overfitting we applied feature pruning to remove the features which occurred less than two times.

Pitler and Nenkova (henceforth P&N) [26] and Lin *et al.* [17] report their results by doing a 10-fold cross validation over PDTB sections 2-22. They use section 0 and 1 as their development data set. Wellner [42], however, uses sections 2-21 for training, sections 23-24 for testing and section 22 for developing his features. The PDTB official guideline recommends sections 2-21 for training, section 22 for development and section 23 for testing [27]. In this work, however, we have followed P&N to prepare the development and training/validation data sets. We have observed that PDTB section 24 contains many incorrect annotations. As Wellner mentioned in his thesis, coordinating connectives within VP-coordination appear frequently as discourse connectives in section 24, whereas according to the PDTB annotation guidelines, such connectives should not be so annotated. In section 24, 183 instances of *and* are annotated as discourse connectives. In 77 of these instances, *and* is immediately inside a VP. In comparison, among the 2442 positive instances of *and* in sections 2-22, only 26 are immediately dominated by a VP. As we wanted to compare our feature set with that of all the previous works on the same setting, a 10-fold cross validation on sections 2-22 seemed to be the better choice to us. An advantage of using 10-fold cross-validation is that we repeatedly use 90% of the data for training and the remaining 10% for testing. The resulting average accuracy, in most cases, is a reliable estimation of the true accuracy when the model is trained on all data and tested on unseen data [34]. Moreover, extensive tests on numerous datasets, with different learning

techniques, have shown that 10 is about the right number of folds to get the best estimate of error, and there is also some theoretical evidence that backs this up [45].

Using gold standard parses, P&N achieved an F-score of 94.19% (Table 2.2). We replicated their work by using both gold standard parses and automatic parses generated by the BLLIP reranking parser [5]. Using gold standard parses we got an F-score of 95.34%. F-score dropped to 93.58% when we used the automatic parses. Interestingly, our replicated result is somewhat better than that of the original work. The reasons behind this may include improvements in MALLET and/or differences in implementation. Lin *et al.* also replicated P&N's work and achieved an F-score of 92.75% and 91.00% with gold standard and automatic parses, respectively.

As we added the new surface level and syntactic features to P&N's features, the performance of the classifier improved. With the gold standard parses the F-score increased to 96.22%. We measured the statistical significance of this improvement by using Wilcoxon signed-rank test [44]. The test was applied to the differences between the error rates for all 10 folds. We used the Wilcoxon signed-rank test instead of the 10-fold cross-validated paired t-test because we found it difficult to reliably determine whether the distribution of the differences in error rates is normal only using a set of 10 samples. The null hypothesis for the test is that the differences are distributed symmetrically around zero, i.e., the error rates are drawn from the same distribution. The test showed that the improvements we obtained are statistically significant at significance level $\alpha = 0.001$ ($p = 0.0009766$).

Our results are better than that reported by Lin *et al.* for their enhanced feature set. They achieved an F-score of 95.76% with gold standard parses. As mentioned earlier Wellner reported an evaluation of his classifier on PDTB sections 23-24. To compare with our results we trained a classifier with his feature set (both constituent and dependency features) and evaluated it by doing a 10-fold cross validation over sections 2-22. Using gold standard parses and Wellner's feature set we got an F-score of 95.85%. Using a Wilcoxon signed-rank test, we found that our feature set gave us a statistically significant improvement over this replicated result at $\alpha = 0.001$ ($p = 0.0009766$).

Table 3.2 compares the results obtained with gold standard (GS) and automatic (AUTO) parses using three different feature sets, namely P&N, Wellner's feature set (WN) and our new

	P&N		WN		F&M	
	GS	AUTO	GS	AUTO	GS	AUTO
Accuracy	97.24	96.11	97.57	97.34	97.78	97.53
Precision	94.08	90.37	95.54	95.10	95.82	95.11
Recall	96.64	97.02	96.18	95.85	96.64	96.52
F-Score	95.34	93.58	95.86	95.47	96.22	95.81

Table 3.2: Comparison of results on the PDTB with different feature sets. *P&N* and *WN* columns present the results obtained by reproducing the work of Pitler and Nenkova [26] and Wellner [42] respectively. The column *F&M* shows the results we obtained using our new set of features. *GS* and *Auto* indicate that the results were obtained using gold-standard parses and automatic parses respectively. These results were obtained by doing a 10-fold cross-validation over the PDTB sections 2-22.

set of features (F&M).

We also evaluated our feature set using the BioDRB. Unlike the PDTB, where discourse annotations were aligned with both raw text and parse trees (the Penn TreeBank), the BioDRB annotations are only aligned with raw text. This means that for each attribute (e.g., connective phrase, arguments) of a discourse relation its annotation contains the offset address of the span of text in the full text article that corresponds to that attribute. To conduct our experiments on the BioDRB in the same way that we did on the PDTB, we had to preprocess it. The sentences in the 24 full text articles were segmented using an open-source natural language processing library named OpenNLP [12]. Parse trees for the segmented sentences were then produced by using the BLLIP re-ranking parser with the self-trained biomedical parsing model [21]. Then, we automatically aligned the discourse annotations with the parse trees and produced new annotations in the PDTB annotation format.

One of our syntactic features, namely the connective category features, depends on the syntactic category of the connective. As discussed above, the category of a connective is given in the classification provided by Knott [15]. However, we found many connectives (e.g., following, by, to) in BioDRB which are not classified in [15]. Therefore, we manually analyzed those connectives and categorized them into three classes: subordinating conjunctions, coordinating

	PDTB-BioDRB	BioDRB-CC	BioDRB-12F-CV
Accuracy	91.43	93.73	94.34
Precision	86.16	87.34	85.17
Recall	75.00	85.36	79.80
F-Score	80.19	86.28	82.36

Table 3.3: Results of the evaluation of the feature set on the BioDRB. The *PDTB-BioDRB* column presents the evaluation of the classifier which was trained on the PDTB and tested on BioDRB. The results of doing a 12-fold cross-validation over the BioDRB is presented in the *BioDRB-12F-CV* column. The second column, *BioDRB-CC*, shows the results obtained by doing a 10-fold cross-validation over the BioDRB considering only the connectives which are common to both the PDTB and BioDRB.

conjunctions and discourse adverbials. Our classification is provided in Appendix B.

We evaluated our feature set on the BioDRB in three different ways. Results of each of these tests are shown in Table 3.3. First, we used the binary maximum entropy classifier which was trained on the PDTB sections 2-22 and tested it on the BioDRB. The column *PDTB-BioDRB* shows the results of this test. Since not all connectives in the BioDRB are also present in the PDTB, for this test we only considered the connectives which are common to both the PDTB and the BioDRB. Second, we did a 10-fold cross-validation on the BioDRB considering only the common connectives. The column *BioDRB-CC* presents the results of this test. Finally, we did a 12-fold cross-validation on the BioDRB considering all of the 179 connective types in it. The results we obtained are shown in column *BioDRB-12F-CV*.

Using our feature set we achieved a significantly better result than what was reported by Ramesh and Yu [32]. By doing a 12-fold cross-validation on the BioDRB, they obtained precision, recall and F-score of 79%, 63% and 69% respectively. The reasons why we achieved better results include the fact that they used the feature set of a biomedical named entity recognizer, named ABNER[1] [38]. The features that ABNER uses are mainly orthographic features, i.e., surface level features [36]. We have found that, for discourse connective identification, syntactic features are more powerful than surface level features.

[1] http://pages.cs.wisc.edu/~bsettles/abner/

With our set of features above, we obtained inferior results for the BioDRB in comparison to the results obtained with the PDTB. There are a few reasons behind this. First, the number of discourse connective types annotated in the BioDRB is greater than that in the PDTB (179 versus 100). Second, there are a few connective types in the BioDRB (e.g., *to*, *by*) which are very frequently used as non-discourse function words. Since they very rarely act as discourse connectives, they are difficult to identify when they do so. In addition, because of the presence of such connectives, the number of negative training examples in the BioDRB is five times greater than the number of positive training examples. Third, since there isn't a gold standard parsed treebank for the BioDRB, we had to depend on an automatic parser which may produce incorrect parses resulting in incorrect syntactic features for both the training and testing phases. A typical sentence in a biomedical research article is usually structurally more complex than one in a Wall Street Journal article. Therefore, a parser is likely to make more errors when parsing a sentence from the BioDRB than parsing a sentence from the PDTB. The BLLIP reranking parser has an F-measure of 91.02% on section 23 of the PTB [5].

3.3 Discussion

Connective or connective category specific ML models have been proved to be useful for discourse argument identification [11]. Following that path, we experimented with connective category specific classifiers for discourse connective identification. However, we obtained results inferior to what we got using a general classifier. We also evaluated the feature sets on each connective category. Table 3.4 shows the connective category-wise results of 10-fold cross-validation over the PDTB sections 2–22 using the P&N and F&M feature sets. Using the new syntactic and surface features we were able to increase the F-score on the subordinating conjunction and discourse adverbial categories by 1.64%-age point and 1%-age point, respectively.

By analyzing classifier errors, we found that it is sometimes not possible to identify connectives based only on syntactic and surface level context, and it seems that some form of semantic understanding is required. For example, one problem with subordinating conjunctions like *when*, *after* and *before* is that they often occur as temporal modifiers and the syntactic

	Coordinating Conjunction		Subordinating Conjunction		Discourse Adverbial	
	P&N	F&M	P&N	F&M	P&N	F&M
Accuracy	98.81	98.89	96.69	97.62	93.92	95.08
Precision	96.95	97.39	91.86	94.67	93.69	95.15
Recall	98.05	97.94	96.17	96.53	95.09	95.73
F-Score	97.49	97.66	93.95	95.59	94.38	95.44

Table 3.4: Connective type-wise results on the PDTB. *P&N* indicates that the results were obtained by using the feature set proposed in [26]. *F&M* indicates that our proposed feature set was used for the evaluation.

context we consider does not indicate that. The following sentence shows one such scenario in which *when* is used as a temporal modifier.

(3.1) Notably, one of Mr. Krenz's few official visits overseas came a few months ago, *when* he visited China after the massacre in Beijing.

To overcome this problem we experimented with a feature created just for subordinating conjunctions. This feature was computed by first taking the parent of the SBAR[2] that dominates the connective and then checking whether it dominates a terminal node having a temporal sense. Words including the names of months, dates, etc. were considered to have a temporal sense. Earlier we have seen how such words combined with an immediately following connective like *when* can be good surface level features. However, here we are looking at a larger context than just the previous word. Inclusion of this feature slightly improved the performance of the classifier on connectives which are subordinating conjunctions. The F-score on this category improved from 95.59% to 95.77%. We believe that including more such semantic features will help to further improve the classifier's performance. The F&M results reported in Table 3.2 do not include this feature.

Adding Wellner's dependency features to our feature set slightly degraded the performance. In a way, this indicates that our feature set already covers all aspects that these dependency features are meant to capture.

[2]Clause introduced by a (possibly empty) subordinating conjunction

From Table 3.2 it can be observed that our proposed feature set is more robust than the P&N features. Using the F&M features, the F-score achieved with gold standard parses and automatic parses differs by only 0.4 percentage points, whereas using the P&N features the difference is about 1.7 percentage points, showing that our feature set is less dependent on the quality of the parse.

Chapter 4

Identifying the Arguments of Explicit Discourse Connectives

Based on some early work on discourse structure in [41], the Penn Discourse Treebank (PDTB) was annotated by treating a discourse connective as a discourse-level predicate which takes exactly two *abstract objects* such as events, states and propositions [1] as its arguments. Since there are no generally accepted abstract semantic categories for classifying the arguments of discourse connectives as have been suggested for verbs (e.g., agent, patient, theme, etc.), the two arguments of a discourse connective are simply labeled as ARG2, for the argument which appears in the clause that is syntactically bound to the connective, and ARG1 for the other argument which may lie in the same sentence as the connective, or anywhere in the prior discourse [27].

A discourse connective and its two arguments can appear in any linear order. For discourse connectives which are subordinating conjunctions, ARG2 corresponds to the subordinate clause since this clause is syntactically associated with the connective. Therefore, the linear order of the arguments can be ARG1–ARG2 or ARG2–ARG1 as shown in (4.1) and (4.2) respectively. In all examples that follow, the discourse connective is underlined, ARG1 appears in italics and ARG2 appears in bold.

(4.1) *The federal government suspended sales of U.S. savings bonds* <u>because</u> **Congress hasn't lifted the ceiling on government debt**.

(4.2) Because **it operates on a fiscal year**, *Bear Stearns's yearly filings are available much earlier than those of other firms*.

The order of arguments for connectives which are coordinating conjunctions or adverbials is generally ARG1–ARG2 because ARG1 usually appears in the prior discourse as shown in (4.3) and (4.4). However, as depicted in (4.5), ARG1 can also be embedded inside ARG2.

(4.3) *He believes in what he plays*, and **he plays superbly**.

(4.4) *Despite the economic slowdown, there are few clear signs that growth is coming to a halt*. As a result, **Fed officials may be divided over whether to ease credit**.

(4.5) As an indicator of the tight grain supply situation in the U.S., market analysts said **that late Tuesday the Chinese government**, *which often buys U.S. grains in quantity*, **turned** instead **to Britain to buy 500,000 metric tons of wheat**.

Generally a discourse connective appears at the beginning of ARG2, but adverbials can also appear medially or finally in ARG2 as shown below.

(4.6) *The chief culprits, he says, are big companies and business groups that buy huge amounts of land "not for their corporate use, but for resale at huge profit."* . . . **The Ministry of Finance**, as a result, **has proposed a series of measures that would restrict business investment in real estate** . . .

(4.7) *Polyvinyl chloride capacity "has overtaken demand* **and we are experiencing reduced profit margins** as a result", . . .

The discourse connectives that are subordinating or coordinating conjunctions are said to be structural [41] because they take their arguments structurally. Discourse adverbials, on the other hand, take one argument, the ARG2, structurally but ARG1 can be anaphoric, i.e., it can appear anywhere in the prior discourse.

In the PDTB there are no syntactic constraints on the structure of an argument. An argument, therefore, can correspond to any set of constituents in the syntax tree. In other words,

there is no direct alignment between a discourse argument span and its syntax. This representation makes the process of identifying arguments difficult because the space of candidate arguments is too large [42]. To overcome this problem Wellner and Pustejovsky [43] (henceforth W&P) proposed a head-based representation of the arguments in the PDTB and reformulated the problem of argument identification into argument head identification.

The head for a given argument extent is determined using two steps. In the first step, the least common ancestor (LCA) node of all the terminal nodes in the argument extent is found. Then a modified version of the head finding rules presented in [8] is used to find the head of the LCA. (4.8) shows a discourse relation where the head for each argument appears in small caps.

(4.8) After ADJUSTING **for inflation**, the Commerce Department said, *spending didn't* CHANGE *in September*.

W&P argued that identifying the heads of discourse arguments would be sufficient or even preferable in many applications involving discourse parsing. In this thesis, we used this head-based representation and built machine learning (ML) rankers to identify the heads of the arguments of explicit discourse connectives.

4.1 Candidate Argument Head Selection

The first step in applying a machine learning approach for this problem is to decide a way to select the candidate argument heads. Since an ARG2 lies in the same sentence as the connective, the choices for the candidate head of ARG2 are limited. But there are many more choices for the candidate head of ARG1 as it can be anywhere in the prior discourse. Following W&P we applied the following two rules to limit the number of candidate argument heads.

Part-of-speech A candidate should have an appropriate part-of-speech (POS) which includes all verb categories, common nouns and adjectives.

Distance A candidate should be within 10 steps from the connective where a single step is a link in the dependency tree or a sentence boundary.

For ARG2 we look for candidates only in the sentence where the connective resides. However, the search space of candidates for ARG1 includes a portion of the prior discourse. Since there are no links between the dependency trees of two consecutive sentences, we created a virtual SENT link connecting the heads of two consecutive sentences. A SENT link is counted as a step.

4.2 ML Models and Features

W&P proposed a maximum entropy ranker with a large set of features to find the best candidate for an argument. Maximum entropy models are widely used for such classification tasks in the field of Natural Language Processing because they are accurate, can be used with non-independent, overlapping features and they are reasonably fast to train [11]. For this particular task a ranker is more advantageous than a classifier as explained by Elwell and Baldridge in [11]. A ranker considers a set of candidates and selects the best one. On the other hand, a classifier would independently classify each candidate as the true head or not. Since a discourse argument has only one head, ranking is more suitable for this problem. We trained two binary maximum entropy ranking models, one for identifying the head of each argument.

The rich set of features proposed by W&P considers syntax, dependency and lexical semantics. The features that we used in our experiments were mostly derived from their feature set. We used two slightly different feature sets for ARG1 and ARG2 identification. For identifying the head of ARG1 we developed and used some new features that leverage knowledge about the head of ARG2. Following W&P we also group the features into classes as shown below. We will specify the features that are new later.

Surface Features This group of features include the connective phrase, candidate head word, position of the connective in the sentence (initial, medial or terminal), category of the connective (subordinating conjunction, coordinating conjunction or adverbial), the distance between the connective phrase and the candidate head word (in terms of number of tokens/words), and whether the candidate head precedes or follows the connective. It also includes a few conjunctive features: the combination of position and category of the connective, the combination of the connective and the candidate head and finally the combination of the connective category and the relative position of the connective

and candidate head (before or after). For ARG1 this group includes a few more features: whether the candidate lies in the same sentence, whether the candidate precedes or follows the ARG2 head, the relative position of ARG1 with respect to the connective and ARG2 (e.g., ARG1–CONN–ARG2, CONN–ARG2–ARG1) and the combination of the relative position with the connective.

Syntactic Features Syntax carries information that has been shown to be useful for identifying the argument. Especially for structural connectives the path between the connective and the argument head often follows a consistent pattern. For example, for the connective *and*, we found that this path very often (more than 60% cases) follow the pattern: `CC--S--S--VP--VB(D|Z|P)`. It includes the path connecting the candidate head with the connective in the syntax tree. We also used two variants of this path in which repeated nodes and part-of-speech nodes were removed. The features in this group are computed from the syntax tree. For a candidate head that lies in a different sentence this path is calculated by traversing a virtual SENT node that connects the roots of two consecutive sentences.

Dependency Features Dependency parses provide a compact and natural representation of the syntax of a sentence with less data sparseness [43]. The features in this group include the shortest dependency path between the connective and the candidate head. Two compressed versions of this path in which repeating links and coordination links are omitted were also considered. We used another feature which is the combination of the POS tag of the candidate head and the POS tag of its parent word in the dependency tree. We used the Stanford dependency parser [9] to compute these features. The dependency structure for sentence (4.8) is shown in Figure 4.1.

Table 4.1 shows all the features we used for this task. An asterix (*) indicates a feature used for ARG1 identification only and + indicates a feature that is new.

Surface Features	
A	Connective phrase
B	Lowercase connective phrase
C	Candidate head word
D	Position of the connective in the sentence (initial, medial or terminal)
E	Category of the connective
F+	Distance between the connective and the candidate
G	Relative position of the candidate and the connective (before or after)
H	Whether the candidate lies in the same sentence as the connective
I*+	Relative position of candidate and ARG2 head
J*+	Relative position of connective, ARG2 head and the candidate
K	A & C
L*+	A & J
M	D & E
N	E & G
Syntactic Features	
O	Path from candidate to the connective
P	Length of path
Q	Collapsed path without part-of-speech
R	Collapsed path omitting repetitions of node tags
S	A & O
Dependency Features	
T	Dependency path from candidate to connective
U	Collapsed path without coordination links
V	Collapsed path after removing repetitions of links
W	A & U
X	A & U
Y+	Combination of candidate and its parent's POS tags

Table 4.1: Feature set for discourse argument identification. * indicates a feature is used for ARG1 identification only. + indicates a feature is new.

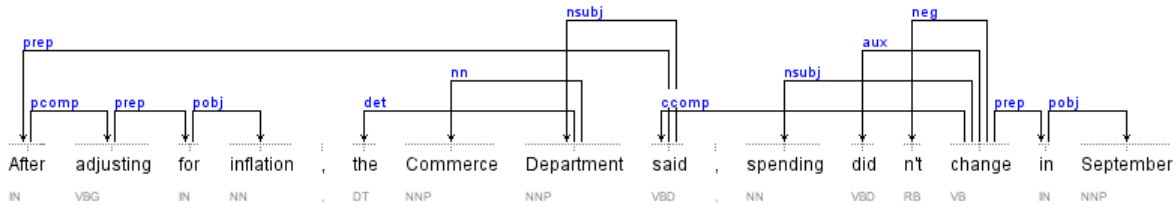

Figure 4.1: Dependency structure for the sentence (4.8). The definitions of the dependency relations are provided in Appendix E.

4.3 Evaluation

We evaluated our feature set using the PDTB and the Biomedical Discourse Relation Bank (BioDRB). For the PDTB we used both the gold standard parses and automatic parses produced by the BLLIP re-ranking parser [5]. For the BioDRB there are no gold standard parses. We produced automatic parses for the BioDRB by using the BLLIP re-ranking parser with the self-trained biomedical parsing model [21]. We also converted the BioDRB annotations to match exactly with the PDTB annotation format allowing us to use the two corpora in the same experimental setup.

We used the open-source machine learning toolkit MALLET [19] to train our models. We trained two independent binary maximum entropy rankers, one for ARG1 and the other for ARG2. Following W&P, to evaluate the performance of the rankers we use *accuracy*, which is the percentage of arguments correctly identified. An argument (head) is assumed to be correct if and only if it is the same argument head word as computed from the argument extent as annotated in the PDTB [43].

Following [43, 11] we used the PDTB sections 2–22 for training, sections 0–1 for development and sections 23–24 for testing. Table 4.2 compares our result (F&M) with that of W&P.

For the BioDRB we used 10-fold cross-validation to evaluate the rankers. For ARG1 and ARG2 we achieved accuracies of 75.24% and 92.44%, respectively. The accuracy was computed by taking the summation of the number of correct answers produced for each fold and dividing it by the total number of explicit discourse relations (2636). To our knowledge, we are the first to report results of identifying discourse arguments in the BioDRB.

	F&M		W&P	
	ARG1	ARG2	ARG1	ARG2
GS	79.25	94.79	76.4	95.4
AUTO	67.71	92.16	69.8	90.8

Table 4.2: Comparison of results for argument identification on the PDTB. Accuracies obtained by [43] for ARG1 and ARG2 identification are presented with the heading *W&P*. Our results are shown with the heading *F&M*. *GS* and *Auto* indicate that the results were obtained using gold-standard parses and automatic parses respectively.

4.4 Discussion

W&P used a re-ranking model to take into account the interdependency between the arguments. They argued that the interdependency between the arguments can not be modeled by using an independent ranker for each argument. By identifying both arguments simultaneously we can take into account global properties such as the pattern of the argument structure (e.g. Connective-ARG2-ARG1 vs. ARG1-Connective- ARG2) or properties of compatibility between the arguments (e.g. agreement in tense). They used independent rankers to produce a set of candidate argument head pairs. These candidate pairs were then ranked by the re-ranker which considers the interdependence features to find the best pair. Using gold standard parses they achieved an accuracy of 95.4% for identifying ARG2 which is slightly better than what we got (94.79%). However, our result using the automatic parses is slightly better than what they achieved (90.8%). On ARG1 identification using the gold standard parses we achieved an increase of 2.85%-age points in accuracy by incorporating the features derived from the ARG2 head. In contrast, there was about 2%-age points decrease using the automatic parses.

Elwell and Baldridge [11] (henceforth E&B) used connective- and connective-category-specific ranking models for argument identification. The category-specific models allowed them to model the specific distribution of a connective or connective category more closely. Since a general model has more data to be trained on, they used an interpolation of the general, connective- and connective-category-specific models. They reported state-of-the art accuracy (82.0%) on ARG1 head identification. However, on ARG2 head identification they only achieved

93.7% accuracy using the gold standard parses. E&B used a slightly modified version of W&P's feature set along with some new features leveraging morphological properties of the connectives and their arguments, additional syntactic configurations and a wider context of preceding and following connectives. Following E&B we also experimented with connective-category-specific models with our feature set. However, the results from this experiment are slightly inferior to the results of our general model.

Since the feature values used in our models are produced by imperfect methods, we are always faced with noise in the information used by our trained models. The two rules used for candidate argument head selection as discussed in Section 4.1 limit the search space of candidate words. We found that for some arguments the head found by the head finding rule does not have an appropriate part-of-speech. (4.9) shows one such case. The head as returned by the head finding rule for ARG1 is *by*, which does not have an appropriate part-of-speech. Similarly, in many cases the true head of (ARG2) cannot be reached from the connective within the 10 step limit. We considered such cases as incorrect predictions.

(4.9) At those levels , stocks are set up to be hammered by index arbitragers , who lock in profits *by buying futures* when **futures prices fall** , ...

We used features derived from ARG2 for identifying ARG1. As discussed above W&P used re-ranking to take into account the interdependency between the arguments. By using ARG2 head to compute feature values when identifying ARG1 head we tried to capture a limited form of interdependency, i.e., the linear order of ARG1, the connective and ARG2. The rational behind this is that for certain connectives the position of ARG2 imposes a restriction on the placement of ARG1. For example, for *and* ARG1 always precede ARG2. However, using ARG2 as feature has an important implication. Any error made in identifying ARG2 will propagate to the process of identifying ARG1. We found that if we do not use the gold standard for ARG2 and use the prediction of the ranker tarined on sections 2-22, the accuracy of identifying ARG1 using gold standard parses drops from 79.25% to 75.61%. This indicates an error propagation of more than 3%-age points.

Chapter 5

Identifying the Senses of Explicit Discourse Connectives

The Penn Discourse Treebank (PDTB) contains annotations of senses for each explicit and implicit discourse connective in the form of sense tags. Sense tags provide a semantic description of the relations between the arguments of connectives[27]. There can be more than one semantic interpretation of a discourse relation depending on the context and the content of the arguments. In the case of multiple simultaneous interpretations, the PDTB provides multiple sense tags.

The set of sense tags in the PDTB is organized into a hierarchy as shown in Figure 5.1. At the top level, or *class level*, there are four tags representing the four major semantic classes: TEMPORAL, CONTINGENCY, COMPARISON and EXPANSION. Each class is further refined by a second level of types. For example, CONTINGENCY has two *types* "Cause" (when the arguments are related via a direct cause-effect relationship) and "Condition" (when one argument poses a hypothetical scenario and the other shows its (possible) consequence). To indicate the semantic contribution of each argument there is a third level of *subtypes*. For CONTINGENCY "Cause" has two subtypes: "reason" (indicating that the situation specified in argument 2 (ARG2) is interpreted as the cause of the situation specified in argument 1 (ARG1), as often with the connective *because*), and "result" (indicating that the situation specified in ARG2 is interpreted as the result of the situation specified in ARG1, as often with the connective *as a result*).

The following is a brief description of the four *class level* tags used in PDTB. For each tag

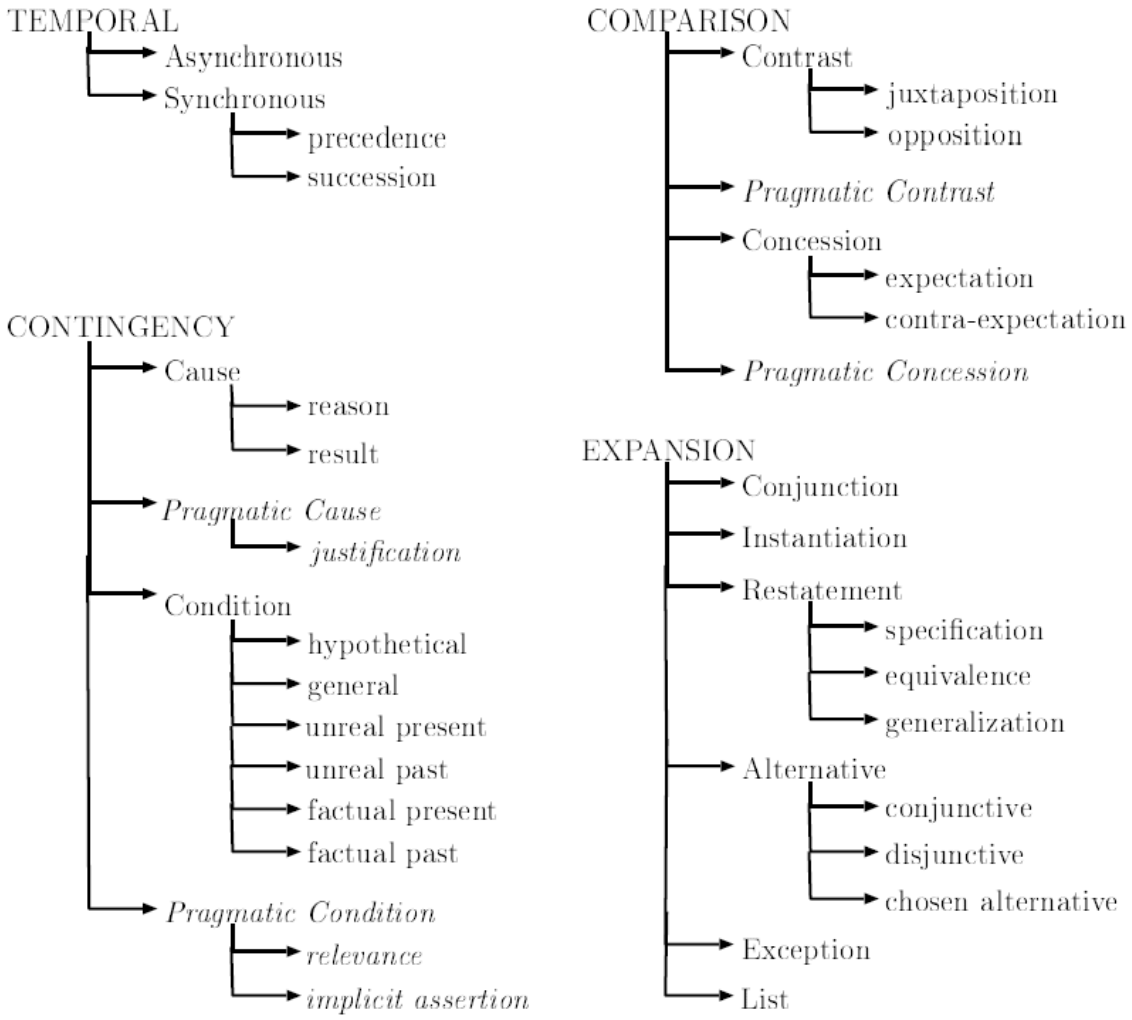

Figure 5.1: Hierarchy of the sense tags in the PDTB. This figure is reproduced from [27].

we also provide an example in which ARG1 and ARG2 are presented in italics and bold face, respectively.

TEMPORAL This sense tag is used when the discourse connective indicates that its arguments are temporally ordered.

> But a Soviet bank here would be crippled unless Moscow found a way to settle the $188 million debt, *which was lent to the country's short-lived democratic Kerensky government* before **the Communists seized power in 1917**. (TEMPORAL:Asynchronous:precedence)

CONTINGENCY This *class level* tag is used when the connective indicates that the situations described in ARG1 and ARG2 are causally related.

> *Use of dispersants was approved* when **a test on the third day showed some positive results**, officials said. (CONTINGENCY:Cause:reason)

COMPARISON This class tag applies when the connective indicates that the discourse relations is established to highlight the differences between the two situations described in the arguments.

> *Most bond prices fell on concerns about this week's new supply and disappointment that stock prices didn't stage a sharp decline*. **Junk bond prices moved higher**, however. (COMPARISON:Contrast:opposition)

EXPANSION This class is used for those discourse relations which expand the discourse and move the narrative forward.

> *Chairman Krebs says the California pension fund is getting a bargain price that wouldn't have been offered to others*. In other words: **The real estate has a higher value than the pending deal suggests**. (EXPANSION:Restatement: equivalence)

In the Biomedical Discourse Relation Bank (BioDRB) senses are annotated following the annotation guideline of the PDTB. However, there are some significant differences. Senses are also organized into a hierarchy in the BioDRB but there are only two levels in the hierarchy. Table 5.1 shows the complete BioDRB sense classification.

The senses at the top level (*types*) are further refined by the senses at the second level (*subtypes*). These refinements can be of two kinds. The first kind refines the semantics. For example, the three subtypes of the sense RESTATEMENT further refines the nature of the restatement. The second kind, on the other hand, specifies the directionality of the arguments. For example, the subtypes of CONCESSION specifies the directionality of the concession: "Contra-Expectation" indicates that ARG1 raises an expectation that ARG2 denies, whereas "Expectation" specifies that ARG2 raises the expectation that ARG1 denies [28].

The following is a brief description of the top level senses in BioDRB.

Type	Subtype	Type	Subtype
CAUSE	Reason	CONDITION	Hypothetical
	Result		Factual
	Claim		Non-Factual
	Justification		
PURPOSE	Goal	TEMPORAL	Synchronous
	Enablement		Precedence
			Succession
CONCESSION	Contra-Expectation	ALTERNATIVE	Chosen-Alternative
	Expectation		Conjunctive
			Disjunctive
CONTRAST		INSTANTIATION	
CONJUNCTION		EXCEPTION	
SIMILARITY		CONTINUATION	
CIRCUMSTANCE	Forward-Circumstance	BACKGROUND	Forward-Background
	Backward-Circumstance		Backward-Background
RESTATEMENT	Equivalence	REINFORCEMENT	
	Generalization		
	Specification		

Table 5.1: BioDRB sense classification. This table is reproduced from [28].

Cause The arguments are causally related and are not in a conditional relation.

Condition One argument is conditioned on the other one.

Purpose One argument presents an action which enables the situation described in the other argument.

Temporal The arguments are temporally related.

Concession One of the arguments raises an expectation that is denied by the other one.

Contrast The values for some shared property in the arguments are in opposition to each other.

Similarity There is a shared property between the arguments.

Alternative The arguments denote alternative situations.

Instantiation ARG1 evokes a set and ARG2 instantiates one or more elements of that set.

Restatement ARG2 restates the situation described in ARG1.

Conjunction The arguments are the members of a list defined in the prior discourse.

Exception ARG2 shows an exception to the generalization presented in ARG1.

Reinforcement ARG2 provides facts to support the claims of effects associated with ARG1.

Continuation ARG2 expands a discourse by saying something about an entity from ARG1.

Circumstance One argument presents a circumstance under which the situation in the other argument is obtained.

Background One argument presents the background necessary for interpreting the other argument.

We treated the problem of identifying the sense of a discourse connective as a machine learning classification problem. Following [26], for the PDTB we did the classification among the top four *sense classes*. For the BioDRB, classification was also done on the top sixteen sense *types*.

5.1 Features

Most of the discourse connectives in the PDTB usually take only one of the senses from the four *sense classes*. For example, *before* always takes a TEMPORAL sense. The connectives themselves are therefore very good features for sense classification. Pitler and Nenkova [26, 28] report better than 90% accuracy on connective sense classification using only the connectives as the features. Pitler and Nenkova [26] (henceforth P&N) used almost the same set of features

for sense classification as they proposed for connective identification. Following that work, we incorporated those features into our feature set. Those features include the connective phrase, syntactic features (e.g., self category, parent category), and pair-wise interaction features between the connective and each syntactic feature.

Wellner [42] experimented with connective sense classification by employing a large set of features. In his feature set he incorporated several features leveraging information involving the argument heads of the discourse connectives. The set includes the heads themselves along with features capturing the syntactic and semantic context between them. However, using features derived from the argument heads will cause error propagation. Though the ARG2 head can be identified with relatively high accuracy (95.4%) [43], the same is not true for the ARG1 head, for which the state-of-the-art accuracy is only 82.0% [11]. We, therefore, incorporated some features in our feature set which are derived from the ARG2 head only. We also used some surface level features, which are slight modifications of some of the features proposed in [42]. The following is a brief description of these two groups of features.

Surface Level Features This group of features includes the category of the connective (coordinating conjunction, subordinating conjunction and discourse adverbial), the combination of the connective phrase and the previous and next words, and the combination of the connective phrase and the previous and next chunk tags.

Head Features These features include the ARG2 head word, and the combination of the connective phrase and the part-of-speech of the ARG2 head word.

5.2 Evaluation

With the features discussed in the previous section, we trained a maximum entropy classifier using MALLET [19]. Following P&N, we evaluated the classifier by doing a 10-fold cross validation (CV) over the PDTB sections 2-22. The results (macro-average accuracies[1]) we obtained using both the gold standard (GS) parses and the automatic parses (AUTO) are shown in Table 5.2. As discussed above, using just the connectives as a feature results in quite a high accuracy

[1] $Accuracy = \frac{1}{n}\sum Accuracy(i)$, where n = 10 for 10-fold CV.

Features	GS	AUTO
Connective	94.61	94.59
Connective + P&N	95.55	94.72
(2) + Surface + Head	95.71	95.37

Table 5.2: Results (accuracies) of classification between the top four sense classes in the PDTB considering both senses to be correct. *GS* and *Auto* indicate that the results were obtained using gold-standard parses and automatic parses respectively.

Features	GS	AUTO
Connective	93.08	93.10
Connective + P&N	94.17	93.24
(2) + Surface + Head	94.25	93.91

Table 5.3: Results of sense classification on the PDTB considering only the first sense to be correct.

for identifying the top level *sense classes* in the PDTB. The inclusion of the new features improve the performance of the classifier on the automatic parses. Using Wilcoxon signed-rank test we found that the improvements are statistically significant at $\alpha = 0.005$ ($p = 0.004883$.).

Some discourse connectives in the PDTB are annotated with multiple senses. For example, in the following sentence the discourse connective *while* is given multiple senses.

(5.1) *The dollar finished mixed* , while **gold declined**. (Temporal.Synchrony or Comparison.Contrast.Juxtaposition)

The results presented in Table 5.2 were obtained by considering both senses of a connective to be correct. If we consider only the first sense to be correct then accuracies drop slightly, as shown in Table 5.3.

Considering both senses to be correct, P&N achieved an accuracy of 94.15% using gold standard parses. Using the same feature set gave us a 1%-age point higher accuracy, which may be due to the fact that we used an improved version of MALLET and there may be differences between our implementations. The human inter-annotator agreement on the top class level senses was 94% [26]. It indicates that further improvement may be difficult to achieve.

BioDRB Type-level Senses	PDTB Class-level Sense
Concession, Contrast	Comparison
Cause, Condition, Purpose	Contingency
Temporal	Temporal
Alternative, Background, Circumstance, Conjunction, Continuation, Exception, Instantiation, Reinforcement, Restatement, Similarity	Expansion

Table 5.4: Grouping of BioDRB sense types into PDTB generalized classes. This table is reproduced from [28]

Features	Only First Sense is Correct	Both Senses are Correct
Connective	89.65	90.23
Connective + P&N	90.60	91.21
(2) + Surface + Head	91.44	92.01

Table 5.5: Results of sense classification on the BioDRB. These results were obtained by doing a 10-fold cross-validation over the BioDRB.

Rashmi Prasad *et al.* [28] discussed a simple baseline for classifying connective senses in the BioDRB. To compare the BioDRB with the PDTB they grouped the BioDRB sense types into the four generalized *class senses* in the PDTB as shown in Table 5.4. Their baseline system considered the connectives as the only features. Considering only the first sense to be correct they achieved an accuracy of 90.9% on classification among the four generalized *class senses*.

We performed the same experiment with the feature set we discussed in Section 5.1. We mapped the BioDRB senses into the four class-level senses following Table 5.4. Considering only the first sense to be correct we obtained an accuracy of 95.36% by doing a 10-fold cross-validation over the BioDRB.

We also experimented with connective sense classification among the sixteen type-level senses in BioDRB. The results we obtained are presented in Table 5.5. We did a 10-fold cross validation over the BioDRB and we report here the macro-average accuracies.

Table 5.5 shows that the surface level and ARG2 head dependent features helped to improve

the classifier's performance by about 0.8%-age point. Using a Wilcoxon signed-rank test we found that the improvement is statistically significant at $\alpha = 0.05$.

5.3 Discussion

The confusion matrix for sense classification between the four class-level senses in the PDTB is shown in Table 5.6. It is evident from that table that most of the errors involve the pair CONTINGENCY and TEMPORAL. 230 discourse connectives having the CONTINGENCY sense were incorrectly predicted as having the TEMPORAL sense. We found that the connectives mainly responsible for such errors are *as* (100 errors) and *when* (95 errors). Interestingly, the number of errors made in the opposite direction, i.e., TEMPORAL predicted as CONTINGENCY, is significantly less, which may indicate that the classifier is more capable of identifying a TEMPORAL sense than a CONTINGENCY sense. We observed that the same connectives (*as* and *when*) are also responsible for the majority of the errors in the opposite direction. The following example shows the sense of *as* being incorrectly predicted.

(5.2) Early in the day , bond dealers said *trading volume was heavy* as **large institutional investors scrambled to buy long-term Treasury bonds on speculation that the stock market 's volatility would lead to a "flight-to-quality" rally** . Predicted: TEMPORAL Correct: CONTINGENCY

Similarly *but* and *while* are responsible for producing errors involving the next most confusing pair - EXPANSION and COMPARISON. However, in the opposite direction (COMPARISON predicted as EXPANSION) *but* contributes only a single error and there is no error for *while*. The following is an example of error involving *but*.

(5.3) *Profit from the calls goes to charity* , but **ABC Sports also uses the calls as a sales tool** : After thanking callers for voting , Frank Gifford offers a football videotape for $19.95 , and 5% of callers stay on the line to order it . Predicted: COMPARISON Correct: EXPANSION

Semantic features derived from the argument text spans would likely improve performance of the classifier in these cases. For example, when a discourse connective has a TEMPORAL sense, its arguments are likely to contain terms that indicate a sense of time. For COMPARISON,

	CONTINGENCY	EXPANSION	TEMPORAL	COMPARISON
CONTINGENCY	2342	49	**230**	11
EXPANSION	6	5025	46	**120**
TEMPORAL	54	20	2872	30
COMPARISON	31	48	38	4480

Table 5.6: Confusion matrix for sense classification on the PDTB. It shows that most of the errors involve the pairs CONTINGENCY-TEMPORAL and EXPANSION-COMPARISON.

there has to be some common words shared by both of the arguments. Consider the example we have shown earlier for describing the COMPARISON class. In that example both arguments talk about prices. ARG1 mentions a decline in stock prices while ARG2 reports an increase in junk bond prices. The presence of two terms with opposite polarity gives a sense of contrast. P&N suggested that incorporating features used for classifying implicit relations may also be useful [26]. Pitler et al. [25] considered the problem of identifying senses of implicit discourse relations. Since there is no discourse connective for an implicit discourse relation, the sense has to be predicted solely from the words in the argument spans. They used several semantic features derived from analyses of the words in the argument spans. These features include polarity feature (number of words in the argument spans with negated, non-negated positive, negative or neutral sentiment), verb class feature (number of words in the argument spans that are in the same Levin verb class [16]), etc. Inclusion of these features in our feature set will improve performance especially on the finer-grained sense types.

We also experimented with identifying the full senses (both *types* and *subtypes*) of the discourse connectives in the BioDRB. We used all the features (P&N, Surface and Head features) and considered both senses given to a connective in the BioDRB to be correct. By doing a 10-fold cross-validation over the BioDRB we obtained an accuracy of 86.84%. To our knowledge we are the first to experiment on identifying the full senses on BioDRB.

Chapter 6

Biomedical Relation Extraction

Understanding relations between biomedical entities is an essential element in biomedical knowledge discovery. Most biomedical relation information is obtained from the free text of scientific research articles. Extracting relationships from such free text scientific articles is, therefore, considered an important biomedical text mining problem. Several approaches to unfold this problem have been reported in the literature. The simplest among these is the co-occurrence based approach. It is based on the simple assumption that entities that frequently co-occur together are somehow related. Relations extracted using this approach usually show high recall but low precision [13]. Pattern-based approaches have been proposed to improve the precision. However, they tend to have low coverage which results in low recall. Other approaches depend on the analysis of the underlying sentence.

These approaches use relation extraction rules which depend on different natural language processing (NLP) techniques to analyze the syntactic and semantic properties of a sentence. There are two ways in which relation extraction rules can be generated from these analyses. In the first rule-based approach, a set of rules is manually curated. A rule can range from a simple regular expression to a complex logical expression involving syntactic or dependency structures. These rules are then used to find occurrence of relations from free text. The other approach depends on machine learning methods to learn implicit extraction rules (in the form of feature weights) automatically from a manually annotated corpus. In this thesis, we used both rule-based and machine learning based methods to extract biomedical relations from free text.

6.1 Protein-Protein Interaction (PPI) Corpora

The type of a relation between biomedical entities can be very general (any biomedical association), or very specific (e.g., a regulatory relation). In our work, we have focused on interaction relations between proteins (PPI). The reason behind this choice is the availability of standard corpora that annotated such relations. Because of their availability several researchers have also used them for experimentation and evaluation of their methodologies. This allows us to compare our work directly with these previous works.

Pyysalo *et al.* [29] did an extensive survey of the five well-known corpora for PPI: AIMed, BioInfer, HPRD50, IEPA, and LLL. The AIMed corpus was created for the comparison of PPI extraction methods [3]. It consists of 225 PubMed abstracts; 197 of them contain protein-protein interactions as identified by the Database of Interacting Proteins (DIP) [1]. The BioInfer corpus was created to support information extraction in the biomedical domain [30]. It consists of sentences from different PubMed abstracts that contain at least one pair of interacting proteins. Besides proteins, it also contains annotation for other entities including genes, RNA types and other related entity types such as biological process or properties. All interactions between these entities are annotated. The HPRD50 corpus was created to evaluate the RelEx system [13], a rule-based biomedical relation extractor. A set of 50 abstracts were collected which are referenced by the Human Protein Reference Database (HPRD). It contains manual annotation for direct physical relations, regulatory relations, as well as modifications. The IEPA corpus [10] was produced by annotating interactions between pairs of chemicals, mostly proteins, in sentences collected from PubMed abstracts. The LLL dataset was created by the Learning Language in Logic (LLL) 2005 challenge [23]. The challenge focused on information extraction of gene interactions in *Bacillus subtilis* where a gene interaction is defined as an agent/target pair, where an agent is a protein and a target is a gene.

There are significant differences between the ways entities and interactions are annotated in these corpora. For example, only AIMed and BioInfer provide an exhaustive annotation of the entities of types relevant to the corpus. The other corpora only provide a list of named entities or output from a named entity recognizer. Because these corpora were developed independently,

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

		AIMed	BioInfer	HPRD50	IEPA	LLL
	size	1955	1100	145	486	77
Entity	*scope*	human P/G	P/G/R and related	human P/G	Chemicals	P/G
	coverage	all occurrences	all occurrences	NER system	16 names	116 names
	types	no	111 types	no	no	P/G
PPI	*types*	no	68 types	no	no	3 types
	binding	no	yes	no	yes	no
	directed	no	yes	no	yes	yes
	complex	no	yes	no	no	no
	negative	no	yes	no	no	no
	certainty	no	no	yes	no	no

Legend:

Size: Number of sentences in the corpus

Entity scope: Types of the named entities identified in the corpus: (P)rotein, (G)ene, (R)NA

Entity coverage: Coverage of in-scope entity occurrences in each sentence

Entity types: Explicit identification of the type of the annotated interactions

PPI types: Explicit indication of the type of the annotated interactions

PPI binding: Identification of the specific text spans that entail the annotated interactions

PPI directed: Specification of the directionality of the interaction

(typically identification of agent vs. patient roles)

PPI complex: Annotation includes nested or n-ary (for $n > 2$) interactions

PPI negative: Annotation of negative interactions

PPI certainty: Annotation of the levels of certainty, or speculativeness, of interactions

Table 6.1: Summary of the five PPI corpora. This table is reproduced from [29].

the attributes of interactions in different corpora vary significantly. A summarization of the corpora was presented in [29] which is reproduced in Table 6.1.

Pyysalo *et al.* has proposed a unified format for these five corpora considering their greatest common properties [29]. In the unified format the interactions are undirected, untyped and there is no text binding of words specifying the interaction. Moreover, there is no complex structure, no negation and no interaction certainty. The unified format for the five corpora is

provided under an open-source license.

Bui *et al.* [2] has proposed a few text processing steps on the unified dataset to improve performance. They replaced all occurrences of protein names with a place holder, i.e, PROTEIN1, PROTEIN2, etc., to improve accuracy of the parser. They defined a set of rules to resolve cases in which one protein name is embedded into another and in which multiple protein names share a common prefix or suffix. These rules, however, do not change the number of protein names in a sentence. They simplified sentences with parenthesized expressions by removing these expressions if they do not include any protein mention. Moreover, sentences consisting of multiple clauses were split into clauses. In our work, we used the unified dataset after preprocessing it with the code provided by [2].

6.2 Rule-based Relation Extraction

6.2.1 Rules

A rule-based relation extraction system uses a set of manually created rules to find relations in free text. RelEx [13] is one very well-known rule-based relation extraction system. The rules incorporated in it depend mainly on the dependency representation of a sentence. Dependency representation represents all sentence relations uniformly as typed dependency relations. The relations are hierarchical: every word is linked to a word dominating it. Each relation is given a type depending on its grammatical role (Figure 6.1). This representation has been shown to be effective for information extraction from text [9]. The rules we used in our work are extensions of the rules used in RelEx. We also drew on ideas presented by Bui *et al.* in [2]. They built a hybrid system for relation extraction by combining a rule-based and a machine learning method. Their method first uses a set of rules to filter out spurious candidate PPI pairs. The remaining pairs are then used for classification using a machine learning method. They did not use a dependency representation to create their rules, instead they depended on a constituent parse tree representation. In our experiment, we used a set of 3 rules. The following gives a detailed description of each of these rules.

Rule 1 extracts relations between two entities (e.g., E1 and E2) expressed as E1-*relation*-

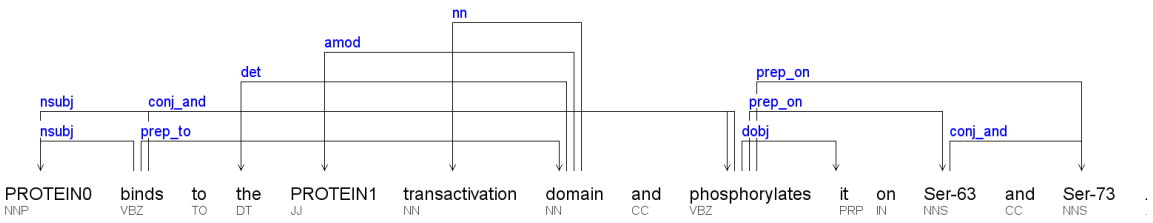

Figure 6.1: Dependency representation for sentence (6.1). The *relation term* - **binds** connects the *effector* PROTEIN0 with the *effectee* PROTEIN1. The dependency path between them is "PROTEIN0- *nsubj*-**binds**-*prep_to*-domain-*amod*-PROTEIN1".

E2, where *relation* is an expression that denotes interaction (e.g., binds, interacts with). For example, the sentence shown in (6.1) contains an interaction between two protein entities, in which the verb *bind* connects the *effector* PROTEIN0 with the *effectee* PROTEIN1. The corresponding dependency representation is shown in Figure 6.1.

(6.1) PROTEIN0 binds to the PROTEIN1 transactivation domain and phosphorylates it on Ser-63 and Ser-73.

This rule is used to extract such *subject-predicate-object* relationships from the dependency tree, where the *subject* and *object* are biomedical entities and the *predicate* contains at least one interaction denoting term (a *relation term*, such as, bind, interact). *nsubj*[2] and *nsubjpass* dependency relations are used as *seeds* to find such paths in the dependency tree. Starting from the governor[3] nodes of the *seeds* we traverse the dependency tree to form candidate paths. A candidate path must end at the head node of a noun phrase (*domain* is the head of the noun phrase *PROTEIN1 transactivation domain* in (6.1)) which contains an entity (E2). The tree traversal is restricted to follow only a fixed set of links. This restricted set can be represented

[2]Nominal subject relationship. The dependent of a relation is the syntactic subject of a clause. The governor is often a verb. Definitions of the dependency relations are provided in Appendix E

[3]The governor of a dependency relation is the word that dominates another word. To illustrate the direction of dependency, a directed edge is drawn from a governor towards its dependent. So, the governor is at the tail of the directed edge.

as a regular expression:
`dep|agent|*comp|*obj|advcl|(inf|part|rc)mod|prep*|abbrev|parataxis`.
We call such a regular expression a *path pattern*. We created such *path patterns* by manually analyzing dependency paths that connect interacting entities. To find the other entity (E1), we search through the tree starting at the nominal subject nodes, the dependents of the seeds. There are three possible cases:

- Entity E1 can be inside the noun phrase of which the nominal subject is the head. We only considered a small set of grammatical relation links to cover this case. Again, we can present this set as a regular expression: `nn|amod|abbrev`. It means that starting from the nominal subject node we only follow either *nn*, *amod* or *abbrev* links to find E1.

- E1 can be in a prepositional complement of a nominal subject as in "Activation of E1 affected E2". To cover such cases we considered the *path pattern*:
`prep_(of|from|like|including|in)|partmod|agent|dobj|dep`.

- E1 can also be inside a relative clause that modifies the nominal subject as in "Proteins which include E1, interact with..". We allowed Rule1 to traverse *rcmod*[4] relations to handle this case. However, we allowed it only when the dependent word of an *rcmod* relation satisfied the regular expression:
`similar|include(s|d)?|members?|identical|involve(s|d)?`.

For each *seed* there can be multiple candidate paths. Each candidate path goes through a filtering stage. If a candidate path does not contain any term which can indicate the occurrence of a relation in the biomedical domain, i.e., a *relation term*, then that path is removed from further consideration. We prepared a list of *relation terms* by combining relation lists used in previous works by [13, 2]. Appendix A shows the contents of this list. In a way, the domain knowledge needed for PPI is encoded in this list of *relation terms*. We also filtered out paths in which the governor of the *seed* is negated. We did this by checking whether that governor also dominates another node in the dependency tree through a *neg*[5] dependency link. A candidate

[4]Relative clause modifier.

[5]The negation modifier is a relation that links a negation word with the word that it modifies.

path may produce multiple interaction pairs.

Rule 2 finds relations in which the entities are connected by one or more prepositions as shown in (6.2), (6.3) and (6.4). This rule is a combination of three sub-rules. Rule 2a finds paths in the dependency tree that connect noun phrases with the prepositions: *of, by, to, on, for, in, through, with*. These paths are considered as candidate paths. If a path contains at least one *relation term* and there is more than one entity in the path, then those entities are used to form interaction pairs. The dependency structure for sentence (6.2) is shown in Figure 6.2. If we apply the *path pattern:* `prep_(of|by|to|on|for|in|through|with)` to that dependency tree, we get a dependency path: *binding-prep_by*-PROTEIN1. Since binding is a relation term, this is a candidate path. Now we look for entities in the noun phrases that are involved in the path. To find the noun phrase extent we apply the *path pattern: nn/amod* to each head of a noun phrase. By doing that we find PROTEIN0 starting from *binding*.

(6.2) Activation of PROTEIN0 by PROTEIN1 in NIH 3T3 cells and in vitro.

(6.3) PROTEIN0 binding by PROTEIN1 is blocked by MAb.

(6.4) A direct interaction between PROTEIN0 subunits and the PROTEIN1.

Sometimes Rule 2a may fail to find a relation even when a clear pattern is observable at the surface level. Figure 6.3 shows one such scenario. From the dependency tree we can find three paths using the *path pattern* for Rule 2a but none of them connects both PROTEIN0 and PROTEIN1. To circumvent this problem we use the second sub-rule Rule 2b. This rule is a regular expression (regex) that is directly applied to the raw text. For each pair of entities (e.g., E1 and E2) in a sentence we generate a sentence pattern and see whether it matches with the regex: `(PREP|REL|N)+ (PREP)(REL|PREP|N)* E1 (REL|N|PREP|PROT)+ E2`. Here, PREP is any preposition, REL is any relation term, N is any noun, and PROT is any protein instance. If a match is found then the pair (E1, E2) is considered as an interaction pair.

Rule 2c is used to find relations between entities in which both entities can be reached from a relation term by following dependency links of types:
`agent|prep_(of|by|to|on|for|in|through|with|between)|nn|amod`. An example is shown in (6.4). To find such relations, for each *relation term* which is a noun, we find

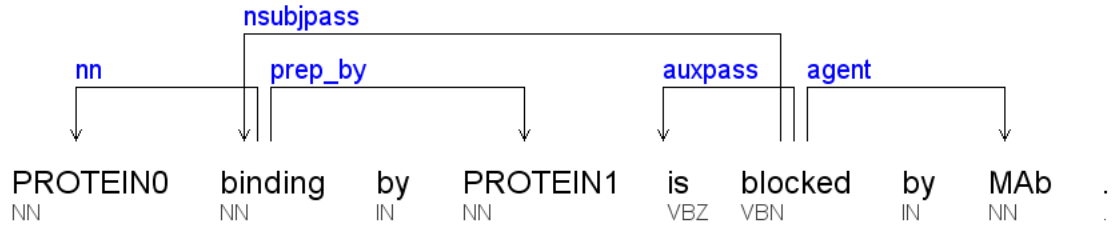

Figure 6.2: Dependency representation for sentence (6.3). The *relation term* **binding** connects PROTEIN0 with PROTEIN1. The dependency path between them is "PROTEIN0-*nn*-**binding**-*prep_by*-PROTEIN1".

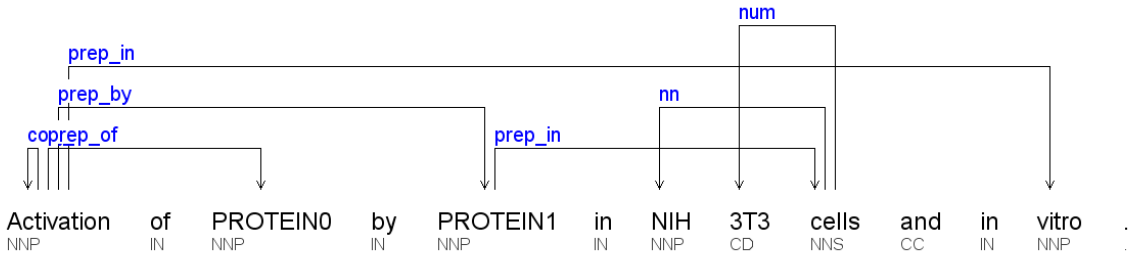

Figure 6.3: Dependency representation for sentence (6.2).

all the entities that can be reached by using this *path pattern*. Each pair of reachable entities is then considered as an interaction pair. There is a possibility of overlapping among these rules, i.e., some interaction pairs can be found by more than one rule.

The final rule, Rule 3, is a regular expression: `E1 [\/-]? E2 REL`, where E1, E2 are entities and REL is any *relation term*. This rule is used to find relations of the form: "E1/E2 binding" or "E1–E2 compound". Such constructs are mostly found in the BioInfer corpus.

Corpus	AIMed	BioInfer	HPRD50	IEPA	LLL
Positive Pairs	1000	2534	163	335	164
Negative Pairs	4834	7132	270	482	166

Table 6.2: Statistics of positive and negative instances in the PPI corpora.

Corpus	AIMed		BioInfer		HPRD50		IEPA		LLL	
	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
Rule1	389	514	608	427	91	29	160	66	113	21
Rule 2a	22	17	76	8	4	0	38	3	7	0
Rule 2b	37	21	76	11	10	1	17	2	8	0
Rule 2c	146	190	270	439	12	12	40	40	6	13
Rule 3	47	25	31	6	0	0	0	0	0	1
All	609	749	992	886	108	42	234	110	127	35
Precision	44.85		52.82		72.0		68.02		78.40	
Recall	60.9		39.15		66.26		69.85		77.44	
F-score	51.65		44.97		69.01		68.92		77.91	

Table 6.3: Results of PPI extraction on five corpora. *TP* and *FP* columns show the number of True Positives and False Positives respectively produced by the rules.

6.2.2 Evaluation

We evaluated our rule-based relation extraction system using the unified PPI corpora [29]. This unified dataset not only provides the positive interaction pairs but also the negative ones. Table 6.2 shows the number of positive and negative interaction pairs annotated for each PPI corpus.

For each sentence in the corpora we a produced parse tree using the BLLIP re-ranking parser [5]. To increase the accuracy of the parser on biomedical text we used the self-trained biomedical re-ranking model [21]. The dependency representation was generated from each parse tree using the Stanford Lexicalized Parser[6]. We used precision, recall and F-score as evaluation metrics. The results we obtained are presented in Table 6.3. The table also provides the individual contribution of each rule in our system.

[6]http://nlp.stanford.edu/software/lex-parser.shtml

Corpus	AIMed	BioInfer	HPRD50	IEPA	LLL
Precision	40	39	76	74	82
Recall	50	45	64	61	72
F-score	44	41	69	67	77

Table 6.4: Performance of RelEx reproduction on five corpora as reported in [29].

Corpus	AIMed	BioInfer	HPRD50	IEPA	LLL
Precision	37.2	51.7	62.2	62.9	81.9
Recall	81.7	67.0	84.7	88.1	85.4
F-score	51.1	58.4	71.7	73.4	83.6

Table 6.5: Performance of the PPI extraction algorithm reported in [2].

6.2.3 Discussion

Pyysalo *et al.* implemented (reproduced) RelEx and evaluated it on the unified corpora [29]. The results they obtained are shown in Table 6.4. However, since this is a reproduction, the performance of the system may diverge slightly from that of the original implementation. Comparing Table 6.3 with Table 6.4, we observe that our system achieved F-score results that were at least as good as those reported by Pyysalo *et al.* with the results especially on the larger corpora (AIMed and BioInfer) being somewhat better.

As mentioned earlier, Bui *et al.* also developed a rule-based relation extraction algorithm as part of their hybrid PPI extraction system [2]. Using constituent-parse-tree-based rules they achieved significantly better results especially on BioInfer. Table 6.5 shows the results they reported on the five PPI corpora.

They used a set of complex rules which are mainly based on some syntactic patterns. On the other hand, we used a set of rules that rely mostly on the dependency representation. The dependency representation gives a simple description of the grammatical relationships in a sentence. It is produced from the phrase structure through a set of complex rules. By using the dependency representation we can exploit the grammatical relationships it provides between the lexical items in a sentence. Our rules are extension of the rules presented in RelEx which take into account the common constructs used in English to express relations. Our rules try to

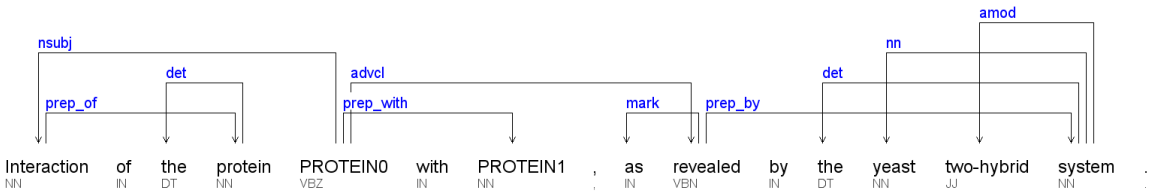

Figure 6.4: Error in dependency representation. The sentence shown here is the title of an abstract. The syntax parser incorrectly interpreted PROTEIN0 as a verb. As a result, the dependency parser made mistakes. For example, the *prep_with* dependency relation should hold between 'Interaction' and 'PROTEIN1'.

capture the same common constructs, however, our implementation varies from that of RelEx. We developed our rules, specifically the implementation details, by analyzing the relations from the LLL corpus. In other words, we used the LLL corpus as our development dataset and the rest of the corpora for testing.

Since the dependency representation is produced from the syntactic structure, there is a possibility of multi-level error propagation. This scenario is illustrated by the Figure 6.4. The parser made an error, PROTEIN0 was incorrectly interpreted as a verb. Due to this error, the dependency tree produced an incorrect dependency tree. We observed that parse quality has a significant influence on the performance of the system. Using parses produced by the Stanford Lexicalized parser resulted in a 1%–8%-age point drop in F-score. The BLLIP parser produced better parses because of the biomedical model which was trained on the biomedical domain.

We found that many of the errors occurred because of failure of the dependency parser. Especially for sentences that have complex structure, the dependency parser often produces incorrect dependency representations. For example, for the sentence shown in (6.5), the corresponding dependency representation contains a *prep_with* grammatical relation between *particular* and PROTEIN2. However, a *prep_with* relation should exist between *interaction* and PROTEIN2. Because of this error, the dependency path between PROTEIN0 and PROTEIN2 becomes (PROTEIN0)-*nsubj*-(confer)-*prep_in*-(particular)-*prep_with*-(PROTEIN2). This de-

pendency path is filtered out by Rule 1 because it does not contain any *relation term*.

(6.5) These results suggest that PROTEIN0 can confer transcriptional regulation and possibly cell cycle control and tumor suppression through an interaction with PROTEIN1 , in particular with PROTEIN2.

Another source of error are the relations involving co-reference. In (6.6), PROTEIN3 is related to both PROTEIN5 and PROTEIN6 through a co-reference. Although there is a dependency path between PROTEIN3 and PROTEIN5, it contains a *conj_but* relation. We do not traverse through a *conj_but* (or a *conj_and*) dependency link because it may lead to extraction of spurious relations in the absence of co-reference.

(6.6) PROTEIN3 also interacts with PROTEIN4 , but it interacts more strongly with PROTEIN5 and PROTEIN6 .

Some of the errors were due to the fact that the relation list we used is not exhaustive. It includes only the terms that were found to appear frequently to express biomedical relations. Hence, it may restrict us from extracting a valid relation only because its vocabulary is limited. An example is shown in (6.7). Since the word *substrate* is not in the relation list, the relation between PROTEIN0 and PROTEIN2 is not recognized.

(6.7) PROTEIN0 (PROTEIN1) is a major substrate of the PROTEIN2 and has been implicated in PROTEIN3 signaling.

6.3 Machine Learning-based Relation Extraction

In this approach relation extraction is treated as a binary classification problem. For the PPI task this requires a representation for a PPI pair and a suitable machine learning method. A protein pair is represented using a set of features which are derived from the sentence or its constituents or its dependency representation. A classifier is then trained on such positive and negative pairs to learn to distinguish between them.

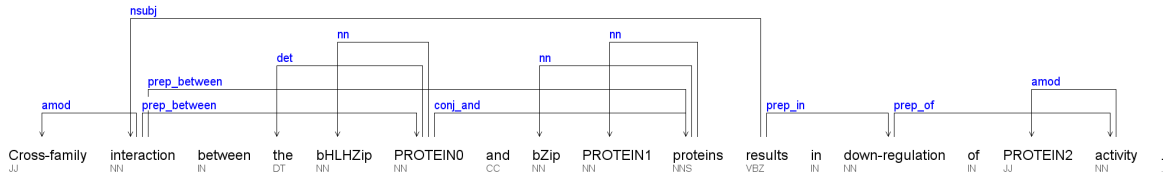

Figure 6.5: Dependency representation for a sentence from the AIMed corpus.

6.3.1 Features

The features we used are mostly derived from the dependency representation of the sentence. We also used some syntactic and surface level features. We grouped our features into three classes.

Dependency Features The path in the dependency representation between two entities contains a great deal of information. In rule-based relation extraction, as we have seen in the previous section, an interaction pair is identified mostly based on the information contained in this path. Consider the dependency representation shown in Figure 6.5. There is an interaction between PROTEIN0 and PROTEIN1 in that sentence. The dependency path between these two entities is the following: PROTEIN0-*prep_between*-interaction-*prep_between*-PROTEIN1. From such a dependency path, we picked the *relations terms* and the dependency relations they govern. For example, from the dependency path just mentioned, we would pick the *relation term* - *interaction* and the grammatical relation *prep_between*. We used the *relation term* and a stemmed version of it as features. We also used another feature that takes into consideration the relative position of the *relation term*, i.e., whether it occurs before the first entity, occurs after the right entity, or occurs between the two entities. The *relation term* combined with the dependency relation (e.g., *prep_between*) was used as another feature which takes into account the grammatical role that the *relation term* playes in the dependency path. We used this feature also in combination with the relative position of the *relation term* and the two entities. In previous section we have seen that there are different ways in which a relation can be

expressed in a sentence. The presence of a *relation term* can be a signal for an interaction and its grammatical role and position can indicate the type of construct used to express that interaction. There can be multiple *relation terms* around the two protein mentions. We tried to find the *key relation term* that best describes the interaction. To find the *key term*, we search for any *relation term* that occurs between the entities and dominates them both in the dependency representation. If a *relation term* is found this way it is considered as the *key term*. If no such *key term* is found in this step we find a word that appears between the entities, has a child which is a *relation term* and dominates the two entities. That child is considered as the *key term*. If we fail to find the *key term* between the entities, we search for it on the left of the first entity and on the right of the second entity in the sentence using the same method. We used the *key term* and its combination with its relative position as features. Another feature in this group is a collapsed version of the dependency path. We replaced all occurrences of *nsubj/nsubjpass* with *subj*, *rcmod/partmod* with *mod*, *prep_x* with *x* and everything else with *O*. For example, the collapsed path between PROTEIN0 and PROTEIN1 in Figure 6.5 would be *prep_between:prep_between:O*. We also considered another collapsed version where we kept only the *prep_** dependency relations. We considered another feature which checks whether there is any node in the path between the entities which governs a *neg* dependency relation. This feature was used to detect negative constructs like "PROTEIN0 does *not* bind to PROTEIN1". The last feature in this group is a boolean feature which checks whether there are two consecutive *prep_between* links in the dependency path.

Syntactic features To compute the features in this group we first identified the least common ancestor (LCA) node of the two entities in a pair in the syntax tree. For the first feature, we used Collins' head finding rule to find the head of that LCA node. If the head word is a *relation term* then this feature takes a stemmed version of the head word as its value, otherwise it takes a *NULL* value. The label of each the constituents in the path between the LCA and each entity combined with its distance from the LCA node was considered as a feature. This feature is inspired by the *POS* feature used in [2].

Surface features The features in this group are derived directly from the raw text. We consid-

ered as features the *relation terms* that occur between the entities in a pair, or are within a short distance (4 token distance) from either entity. The feature values were composed of the *relation terms* and their relative position (i.e., *left*, *middle* or *right*).

6.3.2 Evaluation

We trained a binary maximum entropy classifier using the features described above. We treated all the features as binary features. To avoid data sparsity and overfitting problems we applied feature selection. We used the feature selection code provided in MALLET which ranks the features by information gain and selects a specified number of top ranked features. We performed two types of evaluation. 10-fold cross-validation (CV) and 10-fold abstract-wise cross-validation. The problem with normal 10-fold cross-validation for relation extraction is that pairs from the same sentence can be used for both training and testing for a single fold. This happens because each fold is created by randomly choosing 10% of all the candidate pairs in the corpus. Since the neighboring pairs in the same sentence can have identical features, this can result in an up to 18% over-estimation of the F-score performance compared to a more realistic setting [35, 31]. To circumvent this problem, 10-fold abstract-wise cross-validation was proposed where all data from a single abstract are kept together to avoid using them for both training and testing [31].

In a single abstract the same relation can appear multiple times. There are two approaches to deal with such identical interaction pairs. One approach is called *one-answer-per-occurrence* which requires each mention to be extracted. The other approach, known as *one-answer-per-relation*, demands only each unique pair of interacting entities to be recognized from each document[31]. In this work, we followed the *one-answer-per-occurrence* criterion. Table 6.6 shows the results we obtained on the five PPI corpora.

6.3.3 Discussion

Bui *et al.* [2] used a hybrid method for PPI extraction. Their method was composed of a rule-based algorithm and a machine learning classifier. The rule-based algorithm extracted PPI pairs and divided them into five groups depending on their semantic properties. Then a

	10-fold CV			10-fold abstract-wise CV		
Corpus	Precision	Recall	F-score	Precision	Recall	F-score
AIMed	73.69	59.61	65.85	67.26	49.50	57.03
BioInfer	82.35	73.90	77.86	71.86	55.33	62.52
HPRD50	79.33	77.34	77.52	74.66	66.87	70.55
IEPA	79.39	75.75	77.27	74.48	75.82	75.15
LLL	89.0	87.66	88.12	87.43	89.02	88.22

Table 6.6: Performance of the binary maximum entropy classifier on the PPI corpora.

Corpus	Precision	Recall	F-score
AIMed	55.3	68.5	61.2
BioInfer	61.7	57.5	60.0
HPRD50	70.2	77.9	73.8
IEPA	67.4	83.9	74.7
LLL	84.1	84.1	84.1

Table 6.7: Results of the hybrid PPI extraction method reported in [2].

support-vector-machine (SVM) with a default RBF kernel was used to classify these candidate PPI pairs using features specific for each class. They evaluated their method using 10-fold abstract-wise cross validation on the five PPI corpora. They followed the one answer per occurrence criterion. The results they obtained are presented in Table 6.7.

Comparing Table 6.6 and Table 6.5, we observe that the results we have achieved are competitive with that reported in [2]. They achieved significantly better results only on AIMed. We have got a better precision on AIMed, however, our recall is much lower, which results in a (5.11%) lower F-score. They used a set of complex features which are mainly derived from the syntax tree. Our results show that we can also get good performance on relation extraction using features derived from the dependency representation.

Katrenko and Adriaans used a set of simple features (K&N) derived from the dependency representation. Using 10-fold cross-validation (not abstract-wise) on AIMed they achieved an

F-score of 72.7% using the stacking[7] ensemble method. Using a Naïve Bayes classifier gave them an F-score of 63.8%. However, reproducing their work using their features and a Naïve Bayes classifier we were only able to achieve an F-score of 53.58%. On the LLL corpus their best model could produce an F-score of only 58.5%.

[7] Stacking is a method of combining multiple classification models

Chapter 7

Extracting Higher Order Relations from Text

7.1 Higher Order Relations

In the previous chapter we have shown some examples of biomedical relations. We consider these relations as lower order relations. We use the term higher order relation to denote a relation that exists at a higher level than a typical biomedical relation. A higher order relation can exist on top of a biomedical relation in the sense that it can relate two biomedical relations. Since a higher order relation can take a biomedical relation as its argument, we say that such a relation conveys information that exists at a higher level in the semantics of a discourse. Consider, for example, the following sentence:

(7.1) Aspirin appeared to prevent VCAM-1 transcription, since it dose-dependently inhibited induction of VCAM-1 mRNA by TNF.

By analyzing this sentence we can find two biomedical relations involving Aspirin: Aspirin–*prevents*–VCAM-1 transcription and Aspirin–*inhibits*–induction of VCAM-1 mRNA. These two relations are connected by the word *since*. The connection conveys a causal sense, which indicates that the latter relation causes the earlier one. We call this connection between these two biomedical relations a higher order relation. Extracting higher order relations will give us more information about biomedical relations in a similar way that extracting biomedical

relations give us about biomedical entities. We can use our knowledge of biomedical relations to answer queries regarding biomedical entities. For example, from the previous example we can provide an answer to the query "What prevented VCAM-1 transcription?". In a similar manner, knowledge about higher order relations will enable us to answer to a query like "What evidence is there that Aspirin may prevent VCAM-1 transcription?".

Consider the pair of sentences shown in (7.2). From these two sentences we can find two biomedical relations between *profilin* and *actin* that contrast each other. This contrast relation is an example of a higher order relation.

(7.2) Acanthamoeba profilin affects the mechanical properties of nonfilamentous actin. In contrast, profilin had little effect on the rigidity and viscosity of actin filaments.

We wish to define higher order relations in a much broader sense than what we have just described. A higher order relation not only takes a biomedical relation as its argument, it may also take an *observation*. For example, consider the following sentence:

(7.3) PROTEIN1 does not bind to the PROTEIN2 receptor, but PROTEIN3 binds to both the PROTEIN4 and PROTEIN5.

We can find two biomedical relations here: PROTEIN3–*binds*–PROTEIN4 and PROTEIN3–*binds*–PROTEIN5. But there is another important piece of information which reveals the fact that PROTEIN1 does not bind to PROTEIN2. This actually indicates an absence of an interaction. We call such information, where there are no relational facts but rather some other kind of biomedical information, an *observation*. Here, the observation that PROTEIN1 does not bind to the PROTEIN2 receptor is in direct contrast with the fact that comes afterwards. The sense of contrast is brought by the discourse connective *but*.

Discourse relations can play an important role in finding higher order relations in text. We know that discourse relations make a text coherent by connecting discourse segments together. If the discourse segments that a discourse relation connects contains biological information such as facts, observations or relations, then we can find a relation involving this biological information. And we can interpret that relation by the sense of the discourse relation. For example, if we consider the hierarchical sense types of BioDRB, *but* demonstrates the sense *Contrast* in (7.3). Therefore, it can be said that the observation that PROTEIN1 does not

bind to PROTEIN2 has a *Contrast* relation with the biomedical relation between PROTEIN3 and PROTEIN4/PROTEIN5. This *Contrast* relation is again an example of a higher order relation. Similarly, it can be said from (7.1) that, the interaction between Aspirin and VCAM-1 mRNA has a *Causal* higher order relation with the interaction between Aspirin and VCAM-1 transcription.

We have shown one application of higher order relations: question answering. Another application of higher order relations involving biomedical relations would be knowledge discovery. For example, if we mine a large amount of biomedical articles and extract higher order relations from them, we would be able to form a rich higher order relation graph. From that graph we can infer complex relationships between biomedical entities.

Higher order relations that involve biomedical facts or observations can also be very useful. We have found that a higher order relation can often express a piece of an argument or reasoning. Understanding and combining such relations from a text can provide a way to represent an author's reasoning or argument structure. For example, consider the following abstract taken from the Genia corpus.

Title Aspirin inhibits nuclear factor-kappa B mobilization and monocyte adhesion in stimulated human endothelial cells.

Background The induction of vascular cell adhesion molecule-1 (VCAM-1) and E-selectin by tumor necrosis factor-alpha (TNF) is mediated by mobilization of the transcription factor nuclear factor-kappa B (NF-kappa B). Since **salicylates have been reported to inhibit NF-kappa B activation by preventing the degradation of its inhibitor I kappa B**, *we studied a potential inhibition of this pathway by acetylsalicylate (aspirin) in human umbilical vein endothelial cells (HUVECs).*

Methods and Results Gel-shift analyses demonstrated dose-dependent inhibition of TNF-induced NF-kappa B mobilization by aspirin at concentrations ranging from 1 to 10 mmol/L. *Induction of VCAM-1 and E-selectin surface expression by TNF was dose-dependently reduced by aspirin over the same range*, while **induction of intercellular adhesion molecule-1 (ICAM-1) was hardly affected**. *Aspirin appeared to prevent VCAM-1 transcription*, since **it dose-dependently inhibited induction of VCAM-1 mRNA by**

TNF. As a functional consequence, **adhesion of U937 monocytes to TNF-stimulated HUVECs was markedly reduced by aspirin due to suppression of VCAM-1 and E-selectin upregulation**. *These effects of aspirin were not related to the inhibition of cyclooxygenase activity*, since **indomethacin was ineffective**.

Conclusions Our data suggest that *aspirin inhibits NF-kappa B mobilization, induction of VCAM-1 and E-selectin, and subsequent monocyte adhesion in endothelial cells stimulated by TNF*, thereby **providing an additional mechanism for therapeutic effects of aspirin**.

Most of the sentences in this abstract contain a discourse relation. This shows the typically coherent nature of a scientific article, especially an abstract. A scientific article typically contains many arguments. These arguments are usually expressed with the help of explicit or implicit discourse relations. Therefore, understanding these relations will assist understanding the logical arguments in the text. However, understanding the meaning of arbitrary free text is far from being a solved problem. Yet, we can sometimes extract useful information from it in the form of biomedical relations or observations.

Formally, we define a higher order relation as a binary relation that relates one biomedical relation with another biomedical relation, an observation or a fact. In this thesis we propose a method for extracting such relations with the help of discourse relation parsing and biomedical relation extraction.

7.2 Extracting Higher Order Relations

There are two stages in our method for extracting higher order relations from text. In the first stage we use a discourse relation parser to extract the discourse relations from text. Discourse relations can be of two kinds, implicit and explicit. In this thesis, our focus is on explicit discourse relations. However, implicit discourse relations are also a source of higher order relations. In the second stage we analyze each extracted explicit discourse relation to determine whether it can produce a higher order relation. We use a biomedical relation extraction system in this process. For each argument of an explicit discourse relation we find all occurrences

of biomedical relations in it. If there are no biomedical relations in an argument, we look for the presence of any biomedical entity in it. In case there are any, we treat that argument as an *observation*. However, if an argument of a discourse relation does not contain any biomedical entity in it, we discard that relation from further analysis. Higher order relations are then constructed by pairing the biomedical relations or observations found in the discourse arguments. The sense of the explicit discourse relation is used to interpret all the higher order relations derived from it.

Parsing an explicit discourse relation involves three steps: identifying the explicit discourse connective, the arguments and the sense. We discussed each of these steps in the previous chapters. For identifying the arguments of discourse connectives we used the head-based representation proposed by Wellner and Pustejovsky [43]. In the head-based representation, instead of identifying the whole extent of an argument we identify its lexical head. It was claimed that this head-based representation is often sufficient or even preferred in many applications of discourse parsing [43]. We found that this head-based representation is very suitable for the task of extracting higher order relations. The head of an argument plays an important role in selecting a biomedical relation as an argument to a higher order relation. We will explain this with an example. In (7.1) the ARG2 for the discourse connective *since* consists of the text "it dose-dependently inhibited induction of VCAM-1 mRNA by TNF". There are in fact two biomedical relations in it: Aspirin–*inhibits*–induction of VCAM-1 mRNA and TNF–*induces*–VCAM-1 mRNA. The latter relation is actually not causally related to the relation appearing in the other argument of the connective *since*, i.e., Aspirin–*prevents*–VCAM-1 transcription. We can detect this using the heads of the arguments. The syntactic head of ARG2, i.e., *inhibited* is causally related to the other head, *appeared*, of ARG1. Therefore, any biomedical relation extracted from a discourse argument should be considered as an argument of a higher order relation only if that relation involves the head of the discourse argument. By involvement, we mean the relation depends on the head, that is, it is the subject of the head or the head plays a role in materializing the relationship.

This observation regarding the heads of the discourse arguments has another useful implication. Since the biomedical relations that we have to consider need to involve the argument head, we only have to extract the portion of the argument that is influenced or dominated by

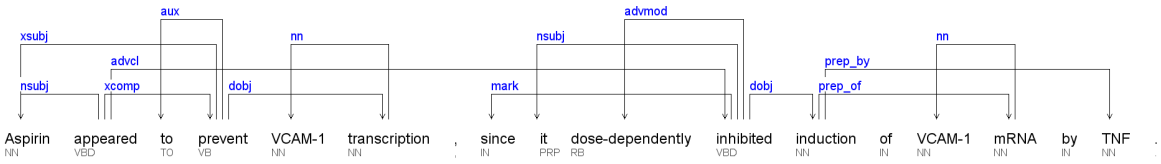

Figure 7.1: Dependency Representation for sentence (7.1).

the head. One simple way to do this is to consider the dependents of the head in the dependency representation. It was reported that finding the dependents of the syntactic head of an argument often gives a good approximation of the argument extent [42]. We found that this approximation gives us an argument extent which is sufficient for our task. Figure 7.1 shows the dependency representation for (7.1). If we follow the dependency links in it, we see that the whole argument extent is a dependent of the syntactic head. There is one dependency link that we must exclude from consideration: the *advcl* dependency link between the heads. In fact, we avoid any dependency link between the heads, otherwise one argument extent would merge with the other one.

So far, we have given an abstract definition of *observation*. More formally, we define an *observation* as a text segment that contains information about one or more biomedical entities, and if there are more than one, the entities are not connected with a biomedical relation. Observations can be further classified into more fine-grained categories (e.g., statement, property) depending on their semantic interpretations. We have left this classification for future work.

We interpret a higher order relation by the sense of the discourse connective. We used the sense types of BioDRB. A graphical representation of a higher order relation extracted from (7.1) can be shown as in Figure 7.1. The sense *Cause.Reason* denotes that the ARG2 is the cause and ARG1 the effect.

Here we propose an algorithm for extracting higher order relations from text. The algorithm is shown in Algorithm 1. It takes an explicit discourse relation as input and returns a set of higher order relations. Each higher order relation is represented as a triple. The first and last element in the triple are the arguments of the higher order relation. The middle element

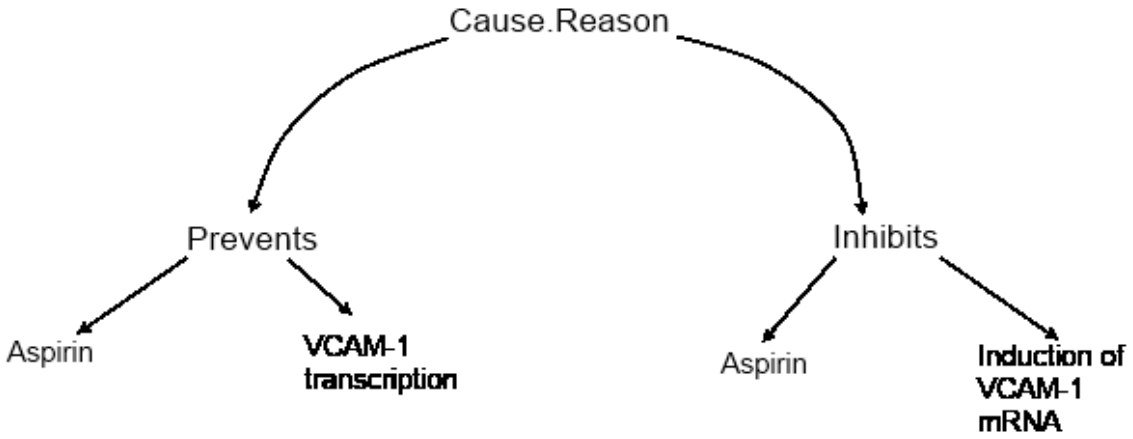

Figure 7.2: Graphical representation of a higher order relation.

denotes the sense or interpretation of the relations. An argument can be either an *observation* or a biomedical relation.

7.3 Evaluation

Our algorithm for extracting higher order relations depends on discourse parsing and biomedical relation extraction. We have discussed our implementation of these components and evaluated their performance in the previous chapters. We have evaluated the algorithm we present in this chapter in terms of how accurately it can use those components in order to find higher order relations. More specifically, we will measure how accurately it can determine the part of the full argument extent that contains the biomedical entities in it.

For this evaluation we used the AIMed corpus. This corpus contains an annotation for protein-protein interactions. From this corpus we collected 69 discourse relations. This selection was done using a semi-automatic process. We used our discourse parsing models to automatically find explicit discourse relations from the abstracts in AIMed. We then manually selected 69 of these extracted relations by verifying their correctness and whether they contained at least one biomedical relation.

For both ARG1 and ARG2 we performed two tests. We measured from the argument heads how many protein mentions occurring within the argument extent (the *True Positives*) are found

Algorithm 1 Algorithm for extracting higher order relations - Part 1

1: **procedure** EXTRACT-HOR(EDR) ▷ EDR is an explicit discourse relation
2: $H1 \leftarrow Head(Arg1(EDR))$ ▷ Get the head of ARG1
3: $H2 \leftarrow Head(Arg2(EDR))$ ▷ Get the head of ARG2
4: $sense \leftarrow Head(Sense(EDR))$ ▷ Get the sense of EDR

5: **if** $H1$ = *noun modifier* **or** *copular verb* **or** *auxilary verb* **then**
6: $H1 \leftarrow Parent(H1)$ ▷ In the dependency representation
7: **end if**
8: **if** $H2$ = *noun modifier* **or** *copular verb* **or** *auxilary verb* **then**
9: $H2 \leftarrow Parent(H2)$ ▷ In the dependency representation
10: **end if**
11: $span1 \leftarrow Dependents(H1)$ ▷ ARG1 extent
12: $span2 \leftarrow Dependents(H2)$ ▷ ARG2 extent
13: **if** $H1 \in span2$ **then**
14: $span2 \leftarrow span2 - span1$ ▷ Remove overlapping part
15: **else if** $H2 \in span1$ **then**
16: $span1 \leftarrow span1 - span2$ ▷ Remove overlapping part
17: **end if**

18: $relations1 \leftarrow \phi$ ▷ Set of relations in ARG1
19: $relations2 \leftarrow \phi$ ▷ Set of relations in ARG2

20: **for all** entity pair $(E1, E2)$ in $span1$ **do**
21: **if** $(E1, E2)$ forms a biomedical relation that involves $H1$ **then**
22: $R \leftarrow FormRelation(E1, E2)$
23: $relation1 \leftarrow relations1 \cup R$
24: **end if**
25: **end for**

Algorithm 1 Algorithm for extracting higher order relations - Part 2

```
26:     for all entity pair (E1, E2) in span2 do
27:         if (E1, E2) forms a biomedical relation that involves H2 then
28:             S ← FormRelation(E1, E2)
29:             relation2 ← relations2 ∪ S
30:         end if
31:     end for
32:     HOR ← φ                                   ▷ Set of Higher Order Relations
33:     if relations1 ≠ φ then
34:         if relations2 ≠ φ then
35:             for all relation R ∈ relations1 and S ∈ relations2 do
36:                 HOR ← HOR ∪ (R, sense, S)
37:             end for
38:         else if span2 contains a biomedical entity then
39:             for all relation R ∈ relations1 do
40:                 HOR ← HOR ∪ (R, sense, Observation(span2))
41:             end for
42:         end if
43:     else if relations2 ≠ φ and span1 contains a biomedical entity then
44:         for all relation S ∈ relations2 do
45:             HOR ← HOR ∪ (Observation(span1), sense, S)
46:         end for
47:     else if both span1 and span2 contain a biomedical entity then
48:         HOR ← HOR ∪ (Observation(span1), sense, Observation(span2))
49:     end if
        return HOR
50: end procedure
```

and how many protein mentions that lie beyond the argument extent (the *False Positives*) are found. For ARG1, we found that our algorithm missed only one protein mention and incorrectly

found three proteins from outside the argument extent. This indicates a precision of 98% and a recall of 99.32%. For ARG2, we obtained a 100% precision and a 99% recall. These results indicate that from a given argument head our algorithm can accurately approximate the argument extent which is used for extracting biomedical relations.

We conducted another experiment, which is similar to the previous one except that now instead of counting only the protein mentions, we counted all the words that can be reached from an argument head. In other words, this experiment evaluates our algorithm in terms of how accurately it can identify the full argument extent (i.e., the words in it). For ARG1 and ARG2 we got an F-score of 91.98% and 92.98% respectively.

The accuracy of the overall system also depends on other components in the pipeline. The errors made in the discourse parsing stage or the relation extraction stage are propagated to the final stage of higher order relation extraction. For example, given incorrect argument heads the algorithm we presented here may find incorrect argument extents. Similarly failure of our relation extractor to find relations in an argument extent may fail to extract a potential higher order relation.

Some examples of higher order relations extracted by our system are shown in Appendix C.

7.4 Discussion

The relation extraction system we developed extracts relations in a simple form. It extracts a pair of interacting entities. However, this simple representation will not accurately represent many biomedical relations. Consider the following sentence:

(7.4) *Deletion of PROTEIN0 enhanced PROTEIN1 and PROTEIN2 silencing* , but **deletion of PROTEIN3 or PROTEIN4 did not affect silencing** , indicating that the mechanism of silencing differs from that at telomeres and silent mating loci.

There is a discourse relation signalled by the explicit discourse connective *but*. The ARG1 of this discourse relation contains biomedical relations between three protein mentions. Our relation extractor extracts two interaction pairs from this text segment: (PROTEIN0, PROTEIN1)

and (PROTEIN0, PROTEIN2). Such interaction pairs can only show whether two entities interact or not. But they do not show the nature of the interactions. Here we can extend an interaction pair by incorporating the relation term most likely to explain that relation. For these two interaction pairs we can form the following triples: (PROTEIN0–*enhanced*–PROTEIN1) and (PROTEIN0–*enhanced*–PROTEIN2). Yet, these extended triples do not cover all the aspects of the relations. A close observation reveals that PROTEIN0 does not directly enhance PROTEIN1 (or PROTEIN2). PROTEIN1 was enhanced by a function performed on PROTEIN0, namely *deletion*. Moreover, that function did not simply enhance PROTEIN1, it enhanced PROTEIN1 *silencing*. These are the details that need to be covered to fully understand the biomedical relation in ARG1. Capturing all aspects of a biomedical relation will result in a more robust higher order relation.

Extraction of many higher order relations is dependent on coreference resolution. For example, in (7.1), Aspirin is anaphorically referred to in ARG2. In our current implementation we lack coreference resolution. Therefore, augmenting a coreference resolution module in our pipeline would be an immediate improvement.

In our implementation, we used a simple but imperfect method to determine whether a biomedical relation involves the head of a discourse argument. We checked whether the head appears between the biomedical entities or within a short distance from either one in the sentence. However, this simple rule may produce spurious higher order relations. Consider the following sentence:

(7.5) *A low level of PROTEIN0 activated transcription of PROTEIN1 by PROTEIN2 RNA polymerase in vitro*, but **a higher level of PROTEIN3 repressed PROTEIN4 transcription**.

The head of ARG1 is *activated*. There are three relations in ARG1, (PROTEIN0, PROTEIN1), (PROTEIN0, PROTEIN2) and (PROTEIN1, PROTEIN2). The relation between PROTEIN1 and PROTEIN2 is expressed independently of the head — *activated*. Therefore, it should be removed from further consideration by the algorithm. However, our simple method would retain it. One way to improve this method would be to consider the rules we presented for rule-based biomedical relation extraction. Most of the rules give a dependency path corresponding to the relation they can extract. That path can then be analyzed to determine whether the

relation depends on the head.

In this thesis we have introduced the concept of higher order relations. We have claimed that these relations can be used for question answering, knowledge discovery or automatic understanding of reasoning presented in text. We have demonstrated that by combining discourse relation parsing and biomedical relation extraction we can extract some of these higher order relations from text. However, more work needs to be done to use their full potential. We need a rich semantic framework for representing biomedical relations and biomedical observations. Besides, we also need to develop a relation extractor that can extract not only the participating entities but also different attributes of the relations. It is required to fully interpret the higher order relations.

Chapter 8

Conclusions and Future Work

8.1 Contributions

In this thesis we have introduced the concept of higher order relations. We have suggested that these relations can be put to use in practical applications such as question answering, knowledge discovery and understanding reasoning in text. We have proposed a method of extracting higher order relations by employing both discourse relation parsing and biomedical relation extraction together with an algorithm that connects these underlying relations. The problem of parsing discourse relations is an important natural language processing (NLP) problem on its own right. There are two general motivations for pursuing this problem: 1) their use in natural language applications such as question answering, complex scenario-level event extraction, automatic summarization, natural language generation, etc. and 2) their use in facilitating better solutions to other semantic and pragmatic problems in NLP [42]. In our case, the use of discourse relation parsing is clearly driven by a motivation of the latter kind. However, we have treated this problem independently and so the discourse relation parsing system we developed is generic and is not focused on any single application.

In this thesis, we have considered parsing explicit discourse relations from text in the style of the Penn Discourse Treebank (PDTB). We have developed machine learning models for identifying the explicit connective, the arguments and the sense of an explicit discourse relation. We have achieved improvements to the state-of-the-art for identifying explicit discourse connectives on the PDTB. On the Biomedical Discourse Relation Bank (BioDRB) we also have

achieved state-of-the art results. To our knowledge we are the first to report results of argument identification on the BioDRB. We have attained promising results for sense identification on the PDTB. For identifying senses on the BioDRB we have obtained better results than what has been previously reported in the literature.

Biomedical relation extraction is also a very important problem for its application in biomedical knowledge discovery. Knowledge of biomedical relations can be used for discovering regulatory pathways, signal cascades, metabolic processes, disease models, etc. [13]. Most biomedical relation information is available in the free text of scientific research articles. Extracting relations from biomedical text is, therefore, considered an important biomedical text mining problem. We have implemented two approaches to extract biomedical relations, more specifically protein-protein interactions, from biomedical text and got promising results.

Extracting an explicit discourse relation gives us two text segments that are connected in a meaningful way. We exploited the idea that this connection can often connect two pieces of biomedical information (e.g., relations, observation) together. We used our biomedical relation extractor to look for biomedical relations in the text segments. We call the resulting relation a higher order relation.

To summarize, our contributions in this thesis are the following:

- We developed a system for parsing explicit discourse relations from text. Explicit discourse relation parsing involves three steps: identifying explicit discourse connectives, identifying their arguments and identifying their sense. We achieved state-of-the-art results on identifying discourse connectives on the Penn Discourse Treebank. More specifically, we got about 1%-age point increase in F-score which we found to be statistically significant. We achieved significantly better results (13.36%-age point increase in F-score) on identifying discourse connectives on the Biomedical Discourse Relation Bank than that reported in the literature so far. We also obtained promising results on argument and sense identification. On the BioDRB we achieved accuracies of 75.24% and 92.44% for identifying ARG1 and ARG2 respectively. We proposed some new features for identifying sense, which led to a 1%-age point increase in accuracy. We developed an explicit discourse relation parser for the biomedical domain leveraging the BioDRB. To our knowledge, we are the first to do this.

- We developed two systems for extracting biomedical relations, specifically protein-protein interactions (PPI). One is based on a rule-based approach and the other uses a machine learning method. We obtained promising results using our rule-based system on standard PPI corpora. We obtained competitive results in comparison to RelEx [13], a well known rule-based relation extraction system. On AIMed and BioInfer we obtained an increase in F-score of 7.65% and 3.97% respectively.

- We introduced the novel concept of a higher order relation. We suggested how they can be useful. We developed an algorithm that extracts higher order relations from text with the help of discourse and biomedical relations.

8.2 Future Work

There is scope for improvement in all areas of our work. There is, of course, the obvious need to enhance the performance of each of the underlying tools. The NLP community continues to work in this direction, and we intend to be part of that progress. I would now like to turn to other aspects of the broader research picture.

In this work, we have handled only those discourse relations that are signalled by explicit discourse connectives. Implicit discourse relations can also be a source of higher order relations. Consider the following example. There is an implicit RESTATEMENT relation between this pair of sentences. Researchers have begun to study this linguistic phenomenon [25].

(8.1) Interferon-gamma (IFN-gamma) is an important immunoregulatory protein produced predominantly by T cells and large granular lymphocytes (LGL) in response to different extracellular signals. [In particular] Two interleukins (ILs), IL-2 and IL-12, have been shown to be potent inducers of IFN-gamma gene expression in both T cells and LGL.

We have left the incorporation of implicit discourse connectives into our higher order relations for the future.

Our biomedical relation extraction system extracts relations in the form of an interaction pair. The interaction pair indicates whether the two entities in that pair interact or not. Though

for some applications such interaction pairs will be sufficient, they do not exhibit all aspects of biomedical relations. Consider the following sentence:

(8.2) SigmaF activity in the forespore regulates the proteolytic processing of SigmaE within the mother cell compartment.

The interaction pair (SigmaF, SigmaE) only expresses that there is some kind of interaction between SigmaF and SigmaE. It does not show other attributes of the relation that take place between them. From the sentence shown above, it can be observed that an interaction can have other important information besides the interacting entities. These pieces of information include the nature of the interaction, the location of the interaction, etc. Taking inspiration from linguistic phenomena and efforts to find computational models, we propose that these different aspects of a biomedical relation can be treated as its arguments, just as we treat different arguments of a verb (e.g., agent, patient, theme). Moreover, an interaction may mention which function of an entity triggers that interaction. To fully understand a biomedical relation we need to cover these aspects of it. For example, in the sentence shown above, it is noted that *activity of SigmaF* (a function of SigmaF) *in the forespore* (a location) *regulates* (nature of interaction) *the proteolytic processing of SigmaE* (a function of SigmaE) *within the mother cell compartment* (again a location). To represent an interaction in this manner we need to develop abstract semantic categories for classifying the arguments. We plan to work on this in the future.

In Chapter 7 we argued that higher order relations can help in understanding the logical arguments or reasoning presented in a text. We showed one biomedical abstract that consists of many discourse relations and biomedical relations. We will now analyze that abstract in more detail to give evidence supporting our argument.

The following is the title of the abstract.

(8.3) Aspirin inhibits nuclear factor-kappa B mobilization and monocyte adhesion in stimulated human endothelial cells.

From this title we can extract two biomedical relations. Since these relations appear in the title we can assume that they are central to understanding the whole abstract. The relations are the following:

(8.3a) Aspirin *inhibits* mobilization of NF-kappa B.

(8.3b) Aspirin *inhibits* monocyte adhesion (by NF-kappa B) in stimulated human endothelial cells.

The following two sentences compose the background of the abstract:

(8.4) The induction of vascular cell adhesion molecule-1 (VCAM-1) and E-selectin by tumor necrosis factor-alpha (TNF) is mediated by mobilization of the transcription factor nuclear factor-kappa B (NF-kappa B).

(8.5) Since **salicylates have been reported to inhibit NF-kappa B activation by preventing the degradation of its inhibitor I kappa B**, *we studied a potential inhibition of this pathway by acetylsalicylate (aspirin) in human umbilical vein endothelial cells (HUVECs).*

From (8.4) we can again extract two biomedical relations as follows:

(8.4a) TNF *induces* VCAM-1 and E-selectin.

(8.4b) Mobilization of NF-kappa B *mediates* induction of VCAM-1 and E-selectin.

There is a discourse relation in (8.5). Having the domain knowledge that Aspirin is a kind of salicylate we can find a relation from ARG2:

(8.5a) Aspirin *inhibits* NF-kappa B.

The higher order relation produced from the discourse relation in (8.5) relates (8.5a) with an observation. Understanding the observation would reveal that because of (8.5a) the author became interested in exploring an effect of Aspirin in human umbilical vein endothelial cells.

The next three sentences are excerpted from the methods and results section:

(8.6) *Induction of VCAM-1 and E-selectin surface expression by TNF was dose-dependently reduced by aspirin over the same range*, while **induction of intercellular adhesion molecule-1 (ICAM-1) was hardly affected**.

(8.7) *Aspirin appeared to prevent VCAM-1 transcription*, since **it dose-dependently inhibited induction of VCAM-1 mRNA by TNF**.

(8.8) As a functional consequence, **adhesion of U937 monocytes to TNF-stimulated HUVECs was markedly reduced by aspirin due to suppression of VCAM-1 and E-selectin upregulation**.

Each of these sentences contains a discourse relation which would lead to a higher order relation. (8.6) introduces the claim that Aspirin reduces the induction of VCAM-1 and E-selectin. The higher order relation in (8.7) uses this claim to explain another observation. Finally the consequence of the claimed fact is presented by the discourse relation in (8.8). The consequence is that Aspirin inhibits monocyte adhesion in stimulated human endothelial cells as expressed in the title. This type of reasoning is what motivated the research presented in this thesis. We will work on doing analysis of this kind automatically in the future.

Bibliography

[1] Nicholas Asher. *Reference to Abstract Objects in Discourse*. SLAP 50, Dordrecht, Kluwer, 1993.

[2] Quoc-Chinh Bui, Sophia Katrenko, and Peter M. A. Sloot. A hybrid approach to extract protein-protein interactions. *Bioinformatics*, 27:259–265, January 2011.

[3] Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. Ramani, and Yuk W. Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155, February 2005.

[4] Eugene Charniak. A maximum-entropy-inspired parser. In *ANLP*, pages 132–139, 2000.

[5] Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180, Ann Arbor, Michigan, 2005. Association for Computational Linguistics.

[6] Rajesh Chowdhary, Jinfeng Zhang, and Jun S. Liu. Bayesian inference of protein-protein interactions from biological literature. *Bioinformatics (Oxford, England)*, 25(12):1536–1542, June 2009.

[7] Aaron M. Cohen and William R. Hersh. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71, March 2005.

[8] M. Collins. *Head-driven statistical models for natural language parsing*. PhD thesis, University of Pennsylvania, 1999.

[9] Marie-Catherine de Marnee and Christopher D. Manning. Stanford typed dependencies manual. http://nlp.stanford.edu/software/stanford-dependencies.shtml, 2011.

[10] J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. Mining MEDLINE: abstracts, sentences, or phrases? *Pac Symp Biocomput*, pages 326–337, 2002.

[11] Robert Elwell and Jason Baldridge. Discourse Connective Argument Identification with Connective Specific Rankers. In *Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pages 198–205, Washington, DC, USA, 2008. IEEE Computer Society.

[12] Apache Software Foundation. Apache opennlp, 2012.

[13] Katrin Fundel, Robert Küffner, and Ralf Zimmer. RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, 23:365–371, January 2007.

[14] Sophia Katrenko and Pieter Adriaans. Learning relations from biomedical corpora using dependency trees. In *Proceedings of the 1st international conference on Knowledge discovery and emergent complexity in bioinformatics*, KDECB'06, pages 61–80, Berlin, Heidelberg, 2007. Springer-Verlag.

[15] Alistair Knott. *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. PhD thesis, University of Edingurg, Edinburgh, 1996.

[16] Beth Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University Of Chicago Press, September 1993.

[17] Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. A pdtb-styled end-to-end discourse parser. *CoRR*, abs/1011.0835, 2010.

[18] William C. Mann and Sandra A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[19] A. K. Mccallum. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu/, 2002.

[20] David Mcclosky and Eugene Charniak. Self-Training for Biomedical Parsing. In *Proceedings of ACL-08: HLT, Short Papers*, pages 101–104, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[21] David McClosky, Eugene Charniak, and Mark Johnson. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 28–36, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[22] Makoto Miwa, Rune Saetre, Yusuke Miyao, and Jun'ichi Tsujii. A rich feature vector for protein-protein interaction extraction from multiple corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 121–130, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[23] C. Nedellec. Learning language in logic - genic interaction extraction challenge. In *Proceedings of the Learning Language in Logic 2005 Workshop at the International Conference on Machine Learning*, 2005.

[24] Yun Niu, David Otasek, and Igor Jurisica. Evaluation of linguistic features useful in extraction of interactions from PubMed; Application to annotating known, high-throughput and predicted interactions in I2D. *Bioinformatics*, 26(1):111–119, January 2010.

[25] Emily Pitler, Annie Louis, and Ani Nenkova. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 683–691, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[26] Emily Pitler and Ani Nenkova. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 13–16, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[27] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. The Penn Discourse TreeBank 2.0. In *LREC*. European Language Resources Association, 2008.

[28] Rashmi Prasad, Susan McRoy, Nadya Frid, Aravind Joshi, and Hong Yu. The biomedical discourse relation bank. *BMC bioinformatics*, 12(1):188+, May 2011.

[29] Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Bjorne, Filip Ginter, and Tapio Salakoski. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(Suppl 3):S6+, 2008.

[30] Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Bjorne, Jorma Boberg, Jouni Jarvinen, and Tapio Salakoski. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1):50+, 2007.

[31] Sampo Pyysalo, Rune Sætre, Jun'ichi Tsujii, and Tapio Salakoski. Why Biomedical Relation Extraction Results are Incomparable and What to do about it. In Tapio Salakoski, Dietrich R. Schuhmann, and Sampo Pyysalo, editors, *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008), Turku, Finland*, pages 149–152. Turku Centre for Computer Science (TUCS), 2008.

[32] Balaji Polepalli P. Ramesh and Hong Yu. Identifying discourse connectives in biomedical text. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2010:657–661, 2010.

[33] A. Ratnaparkhi. *Maximum entropy models for natural language ambiguity resolution*. PhD thesis, ScholarlyCommons@Penn, January 1998.

[34] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-Validation. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 532–538. Springer US, 2009.

[35] Rune Sætre, Kenji Sagae, and Jun ichi Tsujii. Syntactic features for protein-protein interaction extraction. In Christopher J. O. Baker and Jian Su, editors, *LBM (Short Papers)*, volume 319 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[36] B. Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*, pages 104–107, 2004.

[37] B. Settles. ABNER: An open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, 21(14):3191–3192, 2005.

[38] Burr Settles. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics (Oxford, England)*, 21(14):3191–2, July 2005.

[39] Radu Soricut and Daniel Marcu. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 149–156, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[40] Charles Sutton and Andrew Mccallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[41] Bonnie Webber, Matthew Stone, Aravind Joshi, and Alistair Knott. Anaphora and discourse structure. *Computational Linguistics*, 16(1), 1994.

[42] Ben Wellner. *Sequence models and ranking methods for discourse parsing*. PhD thesis, Brandeis University, Waltham, MA, USA, 2009. AAI3339383.

[43] Ben Wellner and James Pustejovsky. Automatically identifying the arguments of discourse connectives. In *EMNLP-CoNLL*, pages 92–101. ACL, 2007.

[44] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[45] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, 1st edition, October 1999.

Appendix A

Relation Terms

A.1 Relation Terms for Protein-Protein Interaction

abolish, abolished, abolishes, abolishing, abrogat, acceler, accelerat, acceptor, accompanied, accompanies, accompany, accompanying, accumul, accumulation, acetylat, acetylate, acetylated, acetylates, acetylating, acetylation, acquir, act, acting, action, activ, activat, activate, activated, activates, activating, activation, activator, acts, adapt, add, addit, adhe, adher, affect, affects, affinities, affinity, aggregat, agoni, agonist, alter, altered, altering, amplif, antagoni, apparat, assembl, assist, associat, associate, associated, associates, associating, association, associations, attach, attached, attaches, attaching, attachment, attack, attacked, attacking, attacks, attenuat, attenuate, attenuated, attenuates, attenuating, augment, augmented, augmenting, augments, autophosphorylat, autoregulat, bind, binding, binds, block, blockage, blocked, blocking, blocks, bound, carbamoylated, carbamoylation, carboxyl, carboxylate, carboxylates, carboxylation, cataly, cause, caused, causes, causing, change, changed, changes, changing, characterization, characterized, cleav, cleavage, cleave, cleaved, cleaves, cleaving, clone, cloning, cluster, co-expression, co-immunoprecipitate, co-immunoprecipitated, co-immunoprecipitates, co-immunoprecipitating, co-immunoprecipitation, co-immunoprecipitations, co-localization, co-localized, co-localizing, co-operat, co-precipit, co-precipitate, co-precipitated, co-precipitates, co-precipitating, co-precipitation, co-precipitations, co-purifi, co-stimulate, co-stimulated, co-stimulating, coactivat, coactivator, coassociation, coexist, coexpres, coexpression, coimmunoprecipitate, coimmunoprecipitated, coimmunoprecipitates, coimmunoprecipitating, coimmuno-

precipitation, coimmunoprecipitations, colocaliz, colocalization, colocalized, compet, compete, competed, competes, competing, complex, complexation, complexed, complexes, complexing, component, compris, concentration, conjugat, conjugate, conjugated, conjugates, conjugating, conjugation, conserved, consisted, consisting, consists, contact, contacted, contacting, contacts, contain, contained, containing, contains, contribute, contributed, contributes, contributing, control, controled, controling, controlled, controlling, controls, convers, convert, converted, converting, converts, cooperat, cooperate, cooperated, cooperates, cooperating, cooperative, coprecipit, coprecipitate, coprecipitated, coprecipitates, coprecipitating, copurifi, correlat, correlate, correlated, correlating, correlation, costimulate, costimulated, costimulating, counteract, counterreceptor, coupl, cripple, crippled, cripples, crippling, cross-link, cross-linked, cross-linking, cross-links, cross-react, cross-reacted, cross-reacting, cross-reacts, cross-talk, crosslink, crosslinker, crosslinking, crosstalk, deacetylat, deacetylate, deacetylated, deacetylates, deacetylating, deacetylation, deaminated, deamination, decarboxylated, decarboxylates, decarboxylation, declin, decreas, decrease, decreased, decreases, decreasing, degrad, degrade, degraded, degrades, degrading, dehydrated, dehydrogenated, dehydrogenation, depend, depended, dependent, depending, depends, dephosphorylat, dephosphorylate, dephosphorylated, dephosphorylates, dephosphorylating, dephosphorylation, deplet, deposi, depress, depressed, depresses, depressing, deriv, destruct, determine, determined, determines, determining, dimer, diminish, diminished, diminishes, diminishing, direct, directed, directing, directs, disrupt, disrupted, disrupting, disruption, disrupts, dissociat, dissociate, dissociated, dissociating, dissociation, distribute, distributed, distributes, distribution, dock, docked, docking, docks, down-regulat, down-regulate, down-regulated, down-regulates, down-regulating, down-regulation, downregulat, downregulate, downregulated, downregulates, downregulating, down-regulation, drive, driven, drives, driving, effect, effected, effecting, effects, elavating, elevat, elevate, elevated, elevates, elevating, eliminate, eliminated, eliminates, eliminating, encod, encode, encoded, encodes, encoding, engage, engaged, engages, engaging, enhanc, enhance, enhanced, enhances, enhancing, enrich, evoke, evoked, exert, exhibit, expos, express, expressed, expresses, expressing, expression, facilitate, facilitates, facilitating, faciliteted, follow, followed, following, follows, form, formation, formed, forms, formylated, functio, function, functioned, functions, fuse, fused, fuses, fusing, generat, generate, generated, generates, generating,

glucosyl, glycosyl, glycosylated, glycosylates, glycosylation, govern, governed, governing, governs, heterodimer, heterodimerization, heterodimerize, heterodimerized, heterodimerizes, heterodimerizing, heterodimers, homodimer, homodimerization, homodimerize, homodimerized, homodimerizes, homodimers, homologous, homologue, hydrol, hydrolyse, hydrolysed, hydrolyses, hydrolysing, hydrolysis, hyperexpr, identified, imitat, immuno-precipit, immunoprecipit, immunoprecipitate, immunoprecipitated, immunoprecipitates, immunoprecipitating, impact, impacted, impacting, impacts, impair, impaired, impairing, impairs, implicate, implicated, import, improv, inactivat, inactivate, inactivated, inactivates, inactivating, inactivation, inactive, includ, incorporate, incorporated, incorporates, incorporation, increas, increase, increased, increases, increasing, increment, induc, induce, induced, induces, inducing, induction, influenc, influence, influenced, influences, influencing, inhibit, inhibited, inhibiting, inhibition, inhibitor, inhibits, initiat, initiate, initiated, initiates, initiating, interact, interacted, interacting, interaction, interactions, interacts, interfer, interrupt, involve, involved, involvement, involves, involving, isomerization, isomerize, isomerized, isomerizes, isomerizing, lead, leading, leads, led, ligand, ligate, ligated, ligates, ligating, ligation, limit, limited, limiting, limits, link, linked, linking, links, localization, mediat, mediate, mediated, mediates, mediating, methylate, methylated, methylates, methylating, methylation, migrat, mobili, mobilisation, mobilise, mobilised, mobilises, mobilising, mobilization, mobilize, mobilized, mobilizes, mobilizing, moderat, modif, modified, modifies, modify, modifying, modulat, modulate, modulated, modulates, modulating, neutrali, neutralise, neutralised, neutralises, neutralising, neutralize, neutralized, neutralizes, neutralizing, obstruct, operat, oppos, overexpress, overproduc, oxidis, oxidiz, oxidization, oxidize, oxidized, oxidizes, oxidizing, pair, paired, pairing, pairs, peroxidizing, perturb, perturbed, perturbing, perturbs, phosphoryates, phosphorylat, phosphorylate, phosphorylated, phosphorylates, phosphorylating, phosphorylation, potentiat, potentiate, potentiated, potentiates, potentiating, prducing, precede, preceded, precedes, preceding, prevent, prevented, preventing, prevents, process, produc, produce, produced, produces, producing, prohibit, promot, promote, promoted, promotes, promoting, raise, raised, raises, raising, react, reactivate, reactivated, reactivates, reactivating, recogni, recognise, recognised, recognises, recognising, recognize, recognized, recognizes, recognizing, recruit, recruited, recruiting, recruitment, recruits, reduc, reduce, reduced, reduces, reducing, reduction, regulat, regulate, regulated, regu-

lates, regulating, regulation, regulator, relate, related, releas, remov, replac, repress, repressed, represses, repressing, requir, require, required, requires, requiring, respond, responded, responding, responds, respons, response, responses, responsible, result, resulted, resulting, results, reversed, secret, sequester, sequestered, sequestering, sequesters, sever, signal, signaled, signaling, signals, splice, stabili, stabilization, stabilized, stimulat, stimulate, stimulated, stimulates, stimulating, stimulation, subunit, suppress, suppressed, suppresses, suppressing, suspend, synergise, synergised, synergises, synergising, synergize, synergized, synergizes, synergizing, synthesis, target, targeted, targeting, targets, terminate, terminated, terminates, terminating, tether, tethered, tethering, tethers, trans-activate, trans-activated, trans-activates, trans-activating, transactivat, transactivate, transactivated, transactivates, transactivating, transamination, transcri, transcribe, transcribed, transcribes, transcribing, transduc, transform, transformed, transforming, transforms, translat, translocat, transport, transregulat, trigger, triggered, triggering, triggers, ubiquitinate, ubiquitinated, ubiquitinates, ubiquitinating, ubiquitination, up-regulat, up-regulate, up-regulated, up-regulates, up-regulating, up-regulation, upregulat, up-regulate, upregulated, upregulates, upregulating, upregulation, use, utilis, utiliz, yield

Appendix B

Connective Category

B.1 Category for Discourse Connectives in BioDRB

1 hr. after	Subordinator
10 min before	Subordinator
180 seconds after	Subordinator
2 hours following	Conj-adverb
20 min later	Conj-adverb
30 minutes prior to	Conj-adverb
812 weeks after	Subordinator
a direct consequence of	Conj-adverb
accordingly	Conj-adverb
additionally	Conj-adverb
after	Subordinator
after which	Subordinator
albeit	Subordinator
also	Subordinator
alternatively	Conj-adverb
although	Subordinator
although still	Conj-adverb

and	Coordinator
and/or	Coordinator
appears to be due at least in part to	Conj-adverb
as	Subordinator
as a consequence	Conj-adverb
as a consequence of	Conj-adverb
as a result	Conj-adverb
as a result of	Subordinator
as an example	Conj-adverb
as demonstrated by	Conj-adverb
as early as 24 h after	Subordinator
as inferred by	Conj-adverb
at some point after	Subordinator
based on	Conj-adverb
because	Subordinator
because of	Conj-adverb
before	Subordinator
besides	Conj-adverb
between 12 h after	Subordinator
both in response to	Subordinator
both upon	Conj-adverb
briefly	Sentential
but	Coordinator
by	Subordinator
by contrast	Conj-adverb
by means of	Subordinator
by the fact that	Subordinator
consequently	Conj-adverb
conversely	Conj-adverb
despite	Conj-adverb

despite the fact that	Subordinator
due mainly to	Conj-adverb
due to	Conj-adverb
during	Conj-adverb
e.g.	Conj-adverb
either or	Coordinator
even though	Conj-adverb
except	Conj-adverb
except for	Subordinator
except that	Subordinator
finally	Conj-adverb
followed 4 hours later by	Conj-adverb
followed by	Conj-adverb
following	Conj-adverb
for	Subordinator
for example	Conj-adverb
for instance	Conj-adverb
forty-eight hours after	Subordinator
four days after	Subordinator
four hours after	Subordinator
four hours later	Conj-adverb
further	Conj-adverb
furthermore	Conj-adverb
given	Conj-adverb
hereafter	Subordinator
however	Conj-adverb
i.e.	Conj-adverb
if	Subordinator
if then	Coordinator
immediately after	Subordinator

in	Conj-adverb
in addition	Conj-adverb
in addition to	Subordinator
in an effort to	Subordinator
in brief	Sentential
in comparison to	Conj-adverb
in comparison with	Conj-adverb
in conclusion	Conj-adverb
in consequence of	Conj-adverb
in contrast	Conj-adverb
in contrast to	Conj-adverb
in fact	Conj-adverb
in general	Sentential
in large part by	Conj-adverb
in large part for	Subordinator
in order for	Subordinator
in order to	Conj-adverb
in outline	Sentential
in part by	Conj-adverb
in part via	Conj-adverb
in particular	Conj-adverb
in particular because of	Conj-adverb
in response to	Conj-adverb
in short	Sentential
in summary	Sentential
in that	Subordinator
in turn	Conj-adverb
in view of the fact that	Subordinator
indeed	Conj-adverb
insofar as	Subordinator

instead	Conj-adverb
mainly by	Conj-adverb
meanwhile	Conj-adverb
moreover	Conj-adverb
namely	Sentential
nevertheless	Conj-adverb
next	Conj-adverb
none-the-less	Conj-adverb
nonetheless	Conj-adverb
nor	Coordinator
not due merely to	Conj-adverb
not due to	Conj-adverb
not only but also	Subordinator
notably	Sentential
now that	Subordinator
on	Subordinator
on the basis of	Conj-adverb
on the contrary	Conj-adverb
on the other hand	Conj-adverb
once	Subordinator
one day after	Subordinator
only if	Subordinator
only when	Subordinator
or	Coordinator
particularly if	Subordinator
particularly since	Subordinator
particularly through	Conj-adverb
particularly to	Conj-adverb
predominantly via	Conj-adverb
presumably as result of	Conj-adverb

presumably due to	Conj-adverb
primarily by	Conj-adverb
prior to	Subordinator
probably because of	Conj-adverb
probably due to	Conj-adverb
provided that	Subordinator
rather	Conj-adverb
regardless of	Conj-adverb
second	Conj-adverb
similarly	Conj-adverb
since	Subordinator
since then	Coordinator
so	Subordinator
specifically	Conj-adverb
specifically to	Conj-adverb
still	Conj-adverb
subsequently	Conj-adverb
such that	Subordinator
then	Coordinator
thereafter	Conj-adverb
thereby	Conj-adverb
therefore	Conj-adverb
third	Conj-adverb
though	Conj-adverb
three days after	Subordinator
three days before	Subordinator
through	Conj-adverb
thus	Conj-adverb
to	Conj-adverb
two hours after	Subordinator

two hours before	Subordinator
two years after	Subordinator
unless	Subordinator
until	Subordinator
upon	Subordinator
via	Conj-adverb
well before	Subordinator
when	Subordinator
whereas	Conj-adverb
while	Subordinator
whilst	Subordinator
with	Conj-adverb

Table B.1: Classification of the discourse connectives in Bio-DRB into categories.

Appendix C

Higher Order Relations

C.1 Examples of Higher Order Relations

This section shows some higher order relations extracted by our system from the PPI corpora.

Example 1 *Acanthamoeba profilin affects the mechanical properties of nonfilamentous actin*. In contrast, **profilin had little effect on the rigidity and viscosity of actin filaments**.

ARG1 head: affects

ARG2 head: had

Sense: Contrast

ARG1 interactions: profilin ↔ actin

ARG2 interactions: profilin ↔ actin

HOR relations: (profilin ↔ actin, Contrast, profilin ↔ actin)

Example 2 *Absence of alpha-syntrophin leads to structurally aberrant neuromuscular synapses deficient in utrophin*. Thus, **alpha-syntrophin has an important role in synapse formation and in the organization of utrophin , acetylcholine receptor , and acetylcholinesterase at the neuromuscular synapse**.

ARG1 head: leads

ARG2 head: has

Sense: Cause

ARG1 interactions: alpha-syntrophin ↔ utrophin

ARG2 interactions: alpha-syntrophin ↔ utrophin, alpha-syntrophin↔ acetylcholine receptor, alpha-syntrophin↔ acetylcholinesterase

HOR relations: (alpha-syntrophin↔ utrophin, Cause, alpha-syntrophin↔ utrophin), (alpha-syntrophin↔ utrophin, Cause, alpha-syntrophin↔ acetylcholine receptor), (alpha-syntrophin↔ utrophin, Cause, alpha-syntrophin↔ acetylcholinesterase)

Example 3 In contrast to **the model of actin binding proposed for fimbrin***, the utrophin actin-binding domain appears to associate with actin in an extended conformation.*

ARG1 head: appears

ARG2 head: model

Sense: Contrast

ARG1 interactions: utrophin ↔ actin

ARG2 interactions: actin ↔ fimbrin

HOR relations: (utrophin ↔ actin, Contrast, actin ↔ fimbrin)

Example 4 As observed with the homologous Drosophila proteins , *hTAFII20 interacts directly with TBP*; however, **additional interactions between hTAFII20 and hTAFII28 or hTAFII30 were detected**.

ARG1 head: interacts

ARG2 head: were

Sense: Concession

ARG1 interactions: hTAFII20 ↔ TBP

ARG2 interactions: hTAFII20 ↔ hTAFII28, hTAFII20 ↔ hTAFII30

HOR relations: (hTAFII20 ↔ TBP, Concession, hTAFII20 ↔ hTAFII28), (hTAFII20 ↔ TBP, Concession, hTAFII20 ↔ hTAFII30)

Example 5 *A bZip protein, Fra1, was found to efficiently interact with USF*. **USF** specifically **interacts with Fra1 but not with other closely related family members, c-Fos, Fra2,**

FosB, or with c-Jun.

ARG1 head: interacts

ARG2 head: were

Sense: Restatement

ARG1 interactions: Fra1 ↔ USF

ARG2 interactions: USF ↔ Fra1

HOR relations: (Fra1 ↔ USF, Restatement, USF ↔ Fra1)

Example 6 *Collectively , eotaxin-3 is yet another functional ligand for CCR3*. **The potency of eotaxin-3 as a CCR3 ligand seems**, however, **to be approximately 10-fold less than that of eotaxin**.

ARG1 head: is

ARG2 head: seems

Sense: Concession

ARG1 interactions: eotaxin-3 ↔ CCR3

ARG2 interactions: eotaxin-3 ↔ CCR3

HOR relations: (eotaxin-3 ↔ CCR3, Concession, eotaxin-3 ↔ CCR3)

Example 7 *Intracerebroventricular injections of 200 ng PROTEIN0 caused a significant rise not only of PROTEIN1 but also of glucagon and glucose levels*. In contrast, **nanogram amounts of PROTEIN0 administered ICV cause a rise of PROTEIN1 levels**.

ARG1 head: caused

ARG2 head: cause

Sense: Contrast

ARG1 interactions: PROTEIN0 ↔ PROTEIN1

ARG2 interactions: PROTEIN0 ↔ PROTEIN1

HOR relations: (PROTEIN0 ↔ PROTEIN1, Contrast, PROTEIN0 ↔ PROTEIN1)

Example 8 *A reduction of food intake imposed on control rats, aimed at mimicking PROTEIN0-*

induced hyperphagia , produced a marked decrease in the expression of muscle PROTEIN1, whereas **ICV infusion of PROTEIN2 prevented such a decrease in PROTEIN3**.

ARG1 head: produced

ARG2 head: prevented

Sense: Contrast

ARG1 interactions: PROTEIN0 ↔ PROTEIN1

ARG2 interactions: PROTEIN2 ↔ PROTEIN3

HOR relations: (PROTEIN0 ↔ PROTEIN1, Contrast, PROTEIN2 ↔ PROTEIN3)

Example 9 *A neurotoxic fragment of PROTEIN0, Abeta 25-35, incubated in the presence of endogenous Ca2+ , increased significantly the PROTEIN1 activity of normoxic brain.* Thus, **PROTEIN0 exerts a similar effect on the membrane-bound PROTEIN1 from normoxic brain or subjected to ischemia reperfusion injury**.

ARG1 head: increased

ARG2 head: exerts

Sense: Cause

ARG1 interactions: PROTEIN0 ↔ PROTEIN1

ARG2 interactions: PROTEIN0 ↔ PROTEIN1

HOR relations: (PROTEIN0 ↔ PROTEIN1, Cause, PROTEIN0 ↔ PROTEIN1)

Example 10 *A low level of PROTEIN0 activated transcription of PROTEIN1 by PROTEIN2 RNA polymerase in vitro*, but **a higher level of PROTEIN3 repressed PROTEIN4 transcription**.

ARG1 head: activated

ARG2 head: repressed

Sense: Concession

ARG1 interactions: PROTEIN0 ↔ PROTEIN1, PROTEIN0 ↔ PROTEIN2, PROTEIN1

↔ PROTEIN2

ARG2 interactions: PROTEIN3 ↔ PROTEIN4

HOR relations: (PROTEIN0 ↔ PROTEIN1, Concession, PROTEIN3 ↔ PROTEIN4), (PROTEIN0 ↔ PROTEIN2, Concession, PROTEIN3 ↔ PROTEIN4)

Appendix D

Abbreviations

D.1 List of Abbreviations

The following table describes the meaning of the abbreviations and acronyms used throughout this thesis.

Abbreviation	Meaning
ABNAR	A Biomedical Named Entity Recognizer
BiODRB	Biomedical Discourse Relation Bank
BLLIP	Brown Laboratory for Linguistic Information Processing
CC	Coordinating Conjunction
CRF	Conditional Random Field
CV	Cross-Validation
FP	False Positive
HPRD	Human Protein Reference Database
LCS	Least Common Subsumer
LLL	Learning Language in Logic
ML	Machine Learning
NLP	Natural Language Processing
NP	Noun Phrase
PDTB	Penn Discourse Treebank

POS	Part-of-Speech
PPI	Protein-protein Interaction
PTB	Penn Treebank
RBF	Radial Basis Function
S	Simple declarative clause
SBAR	Clause introduced by a (possibly empty) subordinating conjunction.
SBARQ	Direct question introduced by a wh-word or a wh-phrase
SINV	Inverted declarative sentence
SQ	Inverted yes/no question, or main clause of a wh-question, following the wh-phrase in SBARQ
SVM	Support Vector Machine
TP	True Positive
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
VP	Verb Phrase

Table D.1: List of abbreviations and acronyms used in the thesis.

Appendix E

Stanford Typed Dependencies

E.1 Stanford Typed Dependencies

The following table shows the definitions of the Stanford typed dependencies that we mentioned in this thesis. These definitions are taken from the Stanford Dependency Manual [9].

Typed dependency	Definition
advcl	Adverbial clause modifier. An adverbial clause modier of a VP or S is a clause modifying the verb.
abbrev	Abbreviation modier. An abbreviation modier of an NP is a parenthesized NP that serves to abbreviate the NP (or to dene an abbreviation).
acomp	Adjectival complement. An adjectival complement of a verb is an adjectival phrase which functions as the complement (like an object of the verb).
agent	An agent is the complement of a passive verb which is introduced by the preposition "by" and does the action.
amod	Adjective modifier. An adjectival modier of an NP is any adjectival phrase that serves to modify the meaning of the NP.

appos	Appositional modifier. An appositional modier of an NP is an NP immediately to the right of the first NP that serves to dene or modify that NP. It includes parenthesized examples.
aux	Auxiliary. An auxiliary of a clause is a non-main verb of the clause, e.g. modal auxiliary, “be” and “have” in a composed tense.
cc	Coordination. A coordination is the relation between an element of a conjunct and the coordinating conjunction word of the conjunct.
ccomp	Clausal complement. A clausal complement of a verb or adjective is a dependent clause with an internal subject which functions like an object of the verb, or adjective. Clausal complements for nouns are limited to complement clauses with a subset of nouns like “fact” or “report”.
cop	Copula. A copula is the relation between the complement of a copular verb and the copular verb.
dep	Dependent. A dependency is labeled as dep when the system is unable to determine a more precise dependency relation between two words. This may be because of a weird grammatical construction, a limitation in the Stanford Dependency conversion software, a parser error, or because of an unresolved long distance dependency .
dobj	Direct object. The direct object of a VP is the noun phrase which is the (accusative) object of the verb.
infmod	Infinitival modifier. An innitival modier of an NP is an innitive that serves to modify the meaning of the NP.
neg	Negation modifier. The negation modier is the relation between a negation word and the word it modies.
nn	Noun compound modifier. A noun compound modier of an NP is any noun that serves to modify the head noun.

nsubj	Nominal subject. A nominal subject is a noun phrase which is the syntactic subject of a clause. The governor of this relation might not always be a verb: when the verb is a copular verb, the root of the clause is the complement of the copular verb, which can be an adjective or noun.
nsubjpass	Passive nominal subject. A passive nominal subject is a noun phrase which is the syntactic subject of a passive clause.
parataxis	Parataxis. The parataxis relation (from Greek for "place side by side") is a relation between the main verb of a clause and other sentential elements, such as a sentential parenthetical, or a clause after a ":" or a ";".
partmod	Participial modifier. A participial modier of an NP or VP or sentence is a participial verb form that serves to modify the meaning of a noun phrase or sentence.
pcomp	Prepositional complement. This is used when the complement of a preposition is a clause or prepositional phrase (or occasionally, an adverbial phrase). The prepositional complement of a preposition is the head of a clause following the preposition, or the preposition head of the following PP.
pobj	The object of a preposition is the head of a noun phrase following the preposition, or the adverbs "here" and "there" .
prep	A prepositional modier of a verb, adjective, or noun is any prepositional phrase that serves to modify the meaning of the verb, adjective, noun, or even another prepositon. In the collapsed representation, this is used only for prepositions with NP complements.

rcmod	Relative clause modifier. A relative clause modier of an NP is a relative clause modifying the NP. The relation points from the head noun of the NP to the head of the relative clause, normally a verb.
xcomp	Open clausal complement. An open clausal complement (xcomp) of a VP or an ADJP is a clausal complement without its own subject, whose reference is determined by an external subject. These complements are always non-nite. The name xcomp is borrowed from Lexical-Functional Grammar.

Table E.1: Definitions of the Stanford typed dependencies mentioned in this thesis.

Curriculum Vitae

Name:	Mohammad Syeed Ibn Faiz
Post-Secondary	The University of Western Ontario
	London, Ontario, Canada
	Masters Student. Computer Science
Education and	University of Dhaka
	Dhaka, Bangladesh
	B.Sc. in Computer Science and Engineering, 2008
Distinction:	1st class 1st
Related Work	Teaching Assistant, Research Assistant
Experience:	The University of Western Ontario
	2010 - 2012
	Full time faculty member
	Dept. of computer science & eng., UAP, Bangladesh.
	March 2009 - August 2010