

EVENT COREFERENCE RESOLUTION

CHEN BIN

Bachelor of Computing (Hons.), NUS

A THESIS SUBMITTED

**FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN
COMPUTER SCIENCE**

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2012

Acknowledgment

Obtaining this Ph.D. degree happens to be the largest achievement in my life. It is a tough challenge academically, physically and mentally. At the very last moment before I complete it, I would like to thank all the people around me that give help and support.

First of all, I would like to direct my most sincere appreciation to my Ph.D. supervisors Professor TAN Chew Lim from National University of Singapore and Dr. SU Jian from Institute for Infocomm Research. Without their invaluable guidance and inspirations, it will be a mission impossible for me to complete this Ph.D. study.

I would like to thank Professor Massimo Poesio from University of Essex for the initial discussion on the topic. I would also like to thank Dr. Sinno Pan Jialin and (Dr. to be) Mr. Zhang Wei from Institute for Infocomm Research for discussion on technical details.

Besides academic helps I received, I would also like to thank my lovely and precious wife Ms. WANG Meizhi for her continuous support both physically and mentally. I also wish to thank my parents for their love and encouragement.

To all the people I mentioned and not yet mentioned, I thank you all for your support and help. I wouldn't be making it this far without you.

Table of Contents

Summary	i
List of Tables	ii
List of Figures	iii
List of Examples	iv
1.Introduction	1
1.1.Backgrounds	1
1.2.Motivations	3
1.3.Thesis Organization	5
2.Literature Survey	6
2.1.Event Definition	6
2.2.Event Coreference Definition	8
2.3.Event Coreference Taxonomy	8
2.4.Related Works	12
2.4.1. <i>Object Coreference Resolution</i>	12
2.4.2. <i>Event Coreference Resolution</i>	14
2.4.3. <i>Other Related Works</i>	16
2.5.Chapter Summary	17
3.Resolution Framework	18
3.1.Mention-Pair Models	19
3.1.1. <i>Instance Generation</i>	19
3.1.2. <i>Learning Models</i>	21
3.2.Chain Formation Models	21
3.2.1. <i>Best-Link Method</i>	21
3.2.2. <i>Graph Partitioning Method</i>	21
3.3.Chapter Summary	22
4.Event Mention Extraction	24
4.1.Heuristic Based Extraction	25
4.2.WordNet Based Extraction	25

4.3.Topic Based Extraction	27
4.3.1. <i>LDA-Based Topic Modelling</i>	28
4.3.2. <i>Combined versus Separated Topic Models</i>	29
4.4.Chapter Summary	29
5.Mention-Pair Resolvers	30
5.1.Seven Distinct Mention-Pair Resolvers	31
5.2.Flat Features	32
5.2.1. <i>Failures of Conventional Features</i>	32
5.2.2. <i>Complications with Event Coreference Resolution</i>	33
5.2.3. <i>Features for Event Coreference Resolution</i>	34
5.3.Structural Information	46
5.3.1. <i>Minimum-Expansion Tree</i>	47
5.3.2. <i>Simple-Expansion Tree</i>	48
5.3.3. <i>Full-Expansion Tree</i>	49
5.3.4. <i>Incorporate Structural Knowledge through Convolution Tree Kernel</i> ...	51
5.4.Utilizing Competing Classifiers' Results	52
5.5.Better Instance Selection Strategy	53
5.6.Chapter Summary	55
6.Chain Formation using Spectral Graph Partitioning	56
6.1.Brief Introduction on Spectral Graph Partitioning	57
6.1.1. <i>Applying Spectral Graph Partitioning to Event Coreference Resolution</i> ...	57
6.1.2. <i>Incorporating Pronoun Coreference Information</i>	59
6.2.Pruning of Inappropriate Edge	60
6.2.1. <i>Eliminating Semantic Incompatibility</i>	60
6.2.2. <i>Propagating the Negative Edges</i>	61
6.3.Seed Clusters Creation	62
6.3.1. <i>Knowledge Guided Seed Clusters</i>	63
6.3.2. <i>Proximity Guided Seed Clusters</i>	64
6.4.Ordering of Eigen-Decomposed Points	65
6.5.Chapter Summary	66

7.Chain Formation through Random Walks	67
7.1.Brief Introduction on Random Walk	68
7.2.Random Walk Model for Event Coreference Resolution	69
7.2.1. <i>Random Walk Through Stationary Transition Probability</i>	69
7.2.2. <i>Random Walks Through Sampling Method</i>	71
7.2.3. <i>Incorporating Corpus Knowledge through Terminating Criteria and Terminating Probability</i>	73
7.2.4. <i>Incorporating Mention Knowledge through Starting Points Selection</i> ...	74
7.3.Incorporating Linguistics Knowledge in a Dynamic Way	74
7.3.1. <i>Dynamic Chain Consistency Enforcement in Random Walk</i>	75
7.3.2. <i>Mention Preference Knowledge through Dynamic Probability Updating</i>	77
7.4.Dynamic Chains Pruning using Object Mentions	78
7.5.Chapter Summary	79
8.Experiments Results and Discussion	81
8.1.Introduction to OntoNotes 4.0 Corpus	81
8.1.1. <i>Event Coreference Annotation</i>	81
8.1.2. <i>Corpus Statistics</i>	82
8.2.Performance Metrics	83
8.2.1. <i>Event Mention Extraction Metric</i>	83
8.2.2. <i>Mention-Pair Resolution Metric</i>	83
8.2.3. <i>Event Chain Resolution Metric</i>	84
8.3.Experiment Settings	85
8.4.Experiment Results	86
8.4.1. <i>Event Mention Extraction Performances</i>	86
8.4.2. <i>Mention-Pair Resolution Performances</i>	89
8.4.3. <i>Event Chain Formation Performances using Spectral Graph Partitioning</i>	96
8.4.4. <i>Event Chain Formation Performances using Random Walk</i>	99
8.4.5. <i>Comparing Spectral Graph Partitioning versus Random Walk</i>	102
8.4.6. <i>Randomly Selected Error Analysis</i>	105
8.5.Chapter Summary	108

9.Conclusion and Future Work	109
9.1.Conclusion	109
9.2.Future Work	111
9.2.1. <i>Employing Ensemble Models</i>	111
9.2.2. <i>Incorporating more Semantic Knowledge</i>	111
9.2.3. <i>Knowledge Deep Parsing</i>	111
Bibliography	113
 Appendix A Model Design Details	 118
Appendix A1 <i>How to Identify Mentions Heads from Parse Tree</i>	118
Appendix A2 <i>WordNet Hypernym Lists for Event and Object</i>	120
Appendix A3 <i>Common Phrases</i>	121
Appendix A4 <i>Event Argument Extraction and Matching</i>	122
Appendix A5 <i>Fixed Pairings of Words</i>	123
Appendix A6 <i>Event Semantic Compatibility/Incompatibility</i>	124
Appendix A7 <i>Semantic Incompatibility Pruning Rules</i>	125
Appendix A8 <i>Semantic Compatibility Preference Rules</i>	126
Appendix A9 <i>Spectral Graph Partitioning</i>	127
Appendix A10 <i>Random Walk Graph Partitioning</i>	129
 Appendix B Empirical Model Settings	 131
Appendix B1 <i>How to Tune Parameters with Training Data</i>	131
Appendix B2 <i>20 Runs of Experiments through Random Sampling of Training and Testing Data</i>	132
Appendix B3 <i>Student's paired t-Test for Statistical Significances</i>	133
 Appendix C Experimental Results	 135
Appendix C1 <i>20 Sets of Experimental Results</i>	135
Appendix C2 <i>List of p-Values for Student's paired t-Test</i>	142

Summary

Event coreference is an important task in event extraction and other natural language processing tasks. Despite its importance, it was merely discussed in previous studies. In this thesis, we are first in the literature to provide a systematic and computation-oriented study on this challenging task. We present a global coreference resolution system dedicated to various sophisticated event coreference phenomena. First of all, seven resolvers are utilized to resolve different event and object coreference mention pairs with a new instance selection strategy and new linguistic features. Competing classifiers and topic related event detection are further imposed to enhance mention-pair resolvers. Secondly, two global solutions, spectral graph partitioning and modified random walk model, are employed for the chain formation. Spectral graph partitioning is equipped with heuristic-guidance and model specific manipulations to produce better coreference chain results. Being the first attempt to apply random walk model for coreference resolution, the modified model utilizes a sampling method, termination criterion and stopping probability to greatly improve the effectiveness of random walk model for event coreference resolution. The new random walk model facilitates a convenient way to incorporate sophisticated linguistic constraints and preferences, the related object mention graph as well as pronoun coreference information not used in previous studies for effective chain formation. Collectively, all the above techniques impose significant B^3 F-score improvement over the baseline system on the OntoNotes 4.0 Corpus.

List of Tables

Table 2.1 :Event Mentions after Restriction	12
Table 4.1 : Hypernymy List for Event v.s. Object NPs	26
Table 5.1 : Features for Conventional NP Resolution	33
Table 5.2 : Positional Features	35
Table 5.3 : Feature List	36
Table 5.4 : String-Matching Features	37
Table 5.5 : Grammatical Role Features	38
Table 5.6 : Mention Characteristic Features	38
Table 5.7 : Better Instance Selection Strategy	54
Table 8.1 : Corpus Distribution	82
Table 8.2 : Two Mention-Pair Evaluations	84
Table 8.3 : Event Mention Extraction using Heuristics and WordNet	86
Table 8.4 : Event Mention Extraction using Topic-related Keywords	87
Table 8.5 : Event Detection Effect on Resolution System	88
Table 8.6 : Flat Feature Effectiveness	90
Table 8.7 : Contribution from Single Knowledge Source	91
Table 8.8 : Different Combinations of Syntactic Structural Knowledge	91
Table 8.9 : Mention-Pair Performances	93
Table 8.10: Performance using Competing Classifiers' Results	94
Table 8.11: Performance using New Instance Selection	96
Table 8.12: Performance using Pronoun Coreference Information	97
Table 8.13: Pruning of Inappropriate Edges	97
Table 8.14: Forming Seed Clusters	98
Table 8.15: Ordering of Decomposed Points	99
Table 8.16: Modified vs. Conventional Random Walk Model	100
Table 8.17: Incorporate Pronoun Coreference into Random Walk	101
Table 8.18: Enforcing Constraints and Preferences	101
Table 8.19: Performance using Object Graph Information	102
Table 8.20: Comparison between Spectral Graph Partitioning & Random Walk	103

List of Figures

Figure 1.1	: Event Extraction Results	4
Figure 3.1	: Two-Step Framework	19
Figure 3.2	: (Ng&Cardie,2002a) Training Instance Selection Illustration ...	20
Figure 3.3	: Relation among Chapters	23
Figure 4.1	: Event Extraction Overview	24
Figure 4.2	: WordNet Hypernymy Filters	27
Figure 5.1	: Mention-Pair Resolvers Overview	30
Figure 5.2	: Illustration for Synonymy List Feature	41
Figure 5.3	: Minimum-Expansion Tree	48
Figure 5.4	: Simple-Expansion Tree	49
Figure 5.5	: Full-Expansion Tree	50
Figure 5.6	: Competing Classifiers' Results	53
Figure 6.1	: Overview of Spectral Graph Partitioning	56
Figure 6.2	: Spectral Graph Partitioning Process	57
Figure 6.3	: Algorithm for Our Spectral Graph Partitioning	59
Figure 6.4	: Negative Edge Propagation	62
Figure 6.5	: Results Before and After Applying Seed Clusters	64
Figure 6.6	: Ordering of Points	65
Figure 7.1	: Overview of Random Walk Model	68
Figure 7.2	: Spectral Graph Partitioning vs. Random Walk Model	75
Figure 7.3	: Object Nodes Pruning Situation	79

List of Examples

Example 1.1	1
Example 1.2	2
Example 1.3	3
Example 2.1	7
Example 2.2	9
Example 2.3	9
Example 2.4	9
Example 2.5	9
Example 2.6	10
Example 2.7	10
Example 2.8	10
Example 2.9	10
Example 2.10	11
Example 5.1	44
Example 5.2	47
Example 5.3	53
Example 5.4	54

Chapter 1: Introduction

1.1 Background

The last decade has seen an explosive growth in the amount of textual information in mass media. Such an enormous amount of information is infeasible for manual processing and understanding. Thus there is a need for an effective and efficient text mining system to gather and utilize the knowledge encoded in the texts. An intelligent text mining system should be able to perform various natural language processing (NLP) tasks such as discourse analysis.

For a successful discourse analysis, a text mining system should have the capability of understanding the referential relations among different expressions in texts. Hence, coreference resolution, the task of resolving a given text expression to its referred expression in prior texts, is important for an intelligent text processing system.

In linguistics, an expression that points back to a previously mentioned expression is called an anaphor, and the expression being referred to by the anaphor is called its antecedent. The mentions denoting the same object/event within the article together form a coreference chain. Most previous work on coreference resolution aims at object coreference in which the coreferent expressions are referring to the real world objects such as persons, places and organizations. For example, in the following sentence,

*"I bought **a new house** yesterday. **It** was in the sub-urban area."*

(Example 1.1)

We can find an anaphor "*it*" and its antecedent is "*a new house*". Both the antecedent and the anaphor are referring to the house which is a real world entity.

While object coreference is well studied, its counterpart, event coreference, lacks exploration. In this thesis, we will conduct a systematic literature survey and analytical and experimental study on event coreference. In the event coreference, the coreferent expressions are referring to an event which is a much more abstract concept compared to the real world object such as the “house” in Example 1.1.

Consider the following sentences,

*“This was an all-white, all-Christian community that all the sudden was taken over -- not taken over, that's a very bad choice of words, but **invaded** by, perhaps different groups. **It** began when a Hasidic Jewish family bought one of the town's two meat-packing plants 13 years ago.”*

(Example 1.2)

We can find the anaphor “*it*” and its antecedent “*invaded*” are referring an event in which an original white and Christian community is diluted by other ethnic groups. Compared to the real world object “house” in Example 1.1, we have an event in Example 1.2 which is more complicated to describe and resolve.

As we can see from these two examples, event coreference is a more complicated linguistic phenomenon than general object coreference. The difficulties come from various aspects. One of the major causes is that the definition for event is more complicated than that of the object (this will be discussed in detail in Chapter 2). Another cause is that event coreference resolution requires more world knowledge than what the object coreference resolution requires.

1.2 Motivation

Event coreference resolution is an important task in natural language processing research. Despite the lack of attention in the previous studies, we found there are two motivating factors for a focused study on the topic of event coreference resolution.

The first important reason is because of its significant existence in text collections. According to our corpus investigation (Section 8.1.2), 69% of the articles in the OntoNotes 4.0 corpus¹ contain at least one event coreference chain while 16% of all the coreference chains are event chains. Such a large contribution makes event coreference resolution an essential and critical task for an intelligent text mining system.

In addition to significant proportion, event coreference resolution helps an event extraction system to acquire more important details related to events.

Consider the following example,

*“Israel has **fired** missiles on the offices of the Palestinian Authority. ... **It** has caused seven deaths with many injuries. ... Israel helicopter gunships **fired** across the Gaza Strip for more than two hours. ... **The attack** in Gaza has been said to cause more violence in Gaza and West Bank and terminate the current round of Middle East peace talk in an unexpected way.”*

(Example 1.3)

The four mentions here, “**fired**”, “**it**”, “**fired**” and “**the attack**” are referring to the same event (an Israel attack in Gaza Strip on Palestinian Authority). Establishing

¹ OntoNotes 4.0 corpus is annotated by multiple research institutions including BBN Technologies, University of Pennsylvania and etc. It consists of more than 2000 documents from mixed genres including new article, new wire, broadcasting news. The detail of OntoNotes 4.0 corpus can be found in Section 8.1.

this event chain will provide us with all necessary details about the “*air strike*” event mentioned in different sentences, such as “Israel / Israel helicopter gunships” being the actuator, “offices of Palestinian Authority” being the target, “seven deaths and many injuries” being the consequence, “Gaza Strip” being the location and “more than two hours” being the duration. Current event extraction systems are mostly working at the sentence level where an event is bound to a single sentence. Without successful event coreference resolution, such separated pieces of information cannot be assembled correctly to facilitate a higher level of information extraction or understanding. Figure 1.1a&b demonstrate the impact of event coreference resolution on event extraction output.

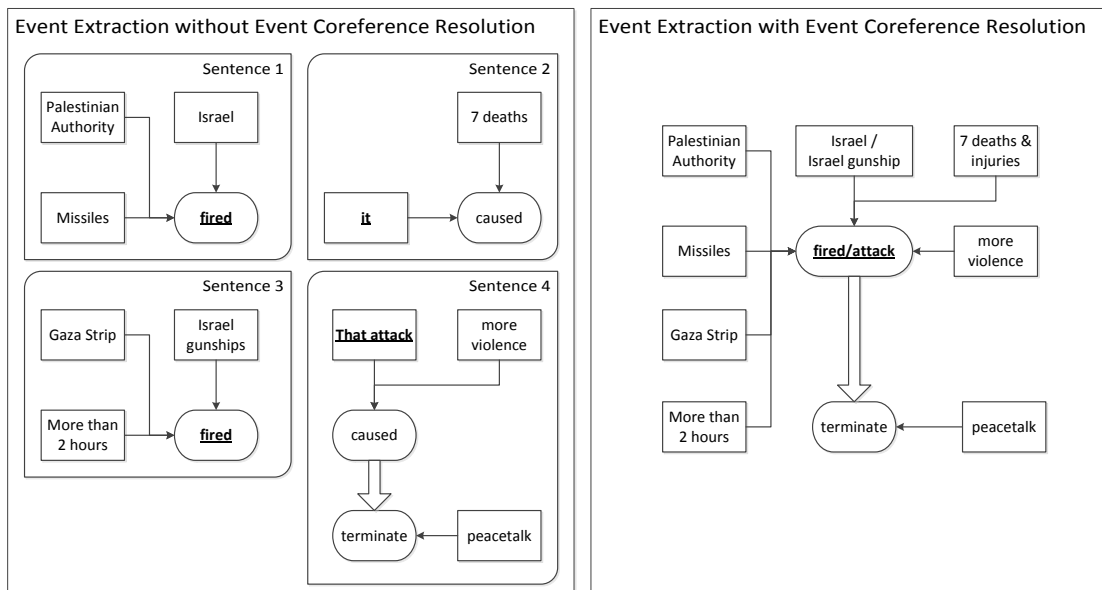


Figure 1.1: Event extraction results (a) without event coreference resolution; (b) with event coreference resolution.

Without the proper event coreference resolution, the information about the “fire/attack” event is scattered in several sentences and each will form an individual event. In such an output, further NLP applications such as summarization system cannot utilize all the details about the same event. While incorporating event

coreference resolution, all the details of the same event are available for further NLP applications.

1.3 Thesis Organization

The rest of this thesis will be organized in the following way. The next chapter (Chapter 2) will provide a thorough literature review and a linguistic study of the event coreference resolution. The discussion on closely related work will also be given in Chapter 2. In Chapter 3, we will introduce the coreference resolution framework. After that, we will move on to our proposed framework in detail. Chapter 4 will describe the event detection process while Chapter 5 elaborates on mention-pair resolvers. Chapter 6 will present the chain formation process using spectral graph partitioning and Chapter 7 will discuss a more creative chain formation technique using the adapted random walk model. Following that, we will present the experimental results in Chapter 8 with discussion. The last chapter (Chapter 9) will conclude the thesis and give a discussion on future research directions.

Chapter 2: Literature Survey

In this chapter, we will present a thorough literature survey on event coreference. The survey will consist of the definition of event in Section 2.1. After that we will move on to define the event coreference and discuss the different types of event coreference phenomenon (Section 2.2 & 2.3). The last section (Section 2.4) will discuss the closely related works.

2.1 Event Definition

Before we can define the event coreferences as a set the textual mentions representing the same event, we need to precisely define what an event is. However, to precisely define an event is hard and complicated by itself. Unlike real world objects such as the “house” in Example 1.1, an event such as “invade” in Example 1.2 is hard to precisely define. A “house” can be uniquely defined by its location (or building name if it is a famous building such as “the White House”). An event will be as abstract as “a thing that happens or takes place, especially one of importance” in the Oxford Dictionary of English.

In this work we wish to take a deeper look into a more rigorous and formal definition. We will present two definitions. One is from philosophers Davidson and Quine. The other is from a computational linguist, Asher.

Davidson and Quine’s Definition

In the 60’s, the famous philosophers Donald Davidson and Willard Quine had proposed a theory to define event and the criteria to distinguish one event from another. The theory described an event as an abstract entity with spatio-temporal properties, and a set of causes and effects. Two events are the same if they have the same cause and effect as well as the same spatio-temporal properties.

Definition by Asher

In the 90's Nicholas Asher as a computational linguist formulated the event definition as follows: An event E should have a theme, necessary roles and possible optional roles. Considering the following example:

“John murdered Mary in their house last night because he thought she was cheating on him.”

(Example 2.1)

Following Quine's definition, this event entity has spatio-property as “John and Mary's house”, temporal property as “last night”, cause as “John thought Mary cheat on John” and effect “Mary was dead”.

Following Asher's definition, this event entity has a theme “murder”, necessary roles as murderer “John” and victim “Mary”, optional roles as location “John and Mary's house” and time “last night”.

Our Adapted Definition

For this study, we will combine the two definitions above to form the following one as:

“For an event E in this study, it has a theme T to describe this event entity; necessary roles R_n (such as “actor” and “patient”), spatiotemporal roles R_{st} (time and location) and optional roles R_o (such as “beneficiary”). Two events are considered the same event if they have the same theme, necessary roles and spatio-temporal roles.”

This definition combines the strength of both Quine's and Asher's. The spatio-temporal roles are added to distinguish events. Cause and effect are normally taken up

by other events and it required much complicated inference to derive. Thus we do not include them in our definition.

2.2 Event Coreference Definition

According to (Jurafsky and Martin, 2000), a natural language expression used to perform reference is called a referring expression, and the entity that is referred to is called a referent. Two expressions that are used to refer to the same entity are said to corefer. Between the two expressions, the prior one is called an antecedent while the latter one is called an anaphor. A collection of the coreferring expressions is a chain of coreference.

After formally defining event and coreference, we can derive the event coreferences as a collection of textual expressions that refers to the same event where the event is defined as in Section 2.1.

2.3 Event Coreference Taxonomy

Event coreferences can be categorized in two ways. The first taxonomy is categorized by the types of relations while the second one is done by the types of expressions. The taxonomy study provides us with a better understanding of the event coreference phenomenon. It also helps us to understand what knowledge is required to correctly resolve them.

Types of Relations

Although up to now we have considered the coreference relation to be an identity relation, there exist other relation types so we will discuss them in this subsection.

Identity Relation

This type of relation strictly follows our definition of event coreference. The anaphor and the antecedent are expressions of the same event, as in the following example.

*“I **ran** two miles yesterday. **The run** did me good.”*

(Example 2.2)

“The run” and “ran” are references to the same running event.

Event-Role Relation

In this type of relation, the anaphor participates as a role in the antecedent’s event.

Consider the following examples:

*“The spokesperson of the White House **announced** in press conference that **The statement** however indicates ...”*

(Example 2.3)

*“... **A huge explosion** happened outside the vessel’s side hull... **The serious damage** made the ship tilt towards one side. ...”*

(Example 2.4)

*“John was **murdered** last night. **The murderer** got away.”*

(Example 2.5)

In Example 2.3, “the statement” has the object role in the “announce” event. In Example 2.4, “the serious damage” is the consequence of the event “a huge explosion”. In Example 2.5, “the murderer” is the actor role in the “murder” event.

Although at the first glance, none of these three are related to our target, identity coreferences. In contrast, in Example 2.3, the object of the “announce” event is inseparable from the event itself. It means that an “announce” event will make no sense without whatever is announced. Thus the anaphor “statement” holds an identity relation to the “announce” event. In contrast, Examples 2.4 and 2.5 are not in the scope of this thesis as they are indeed non-identical relations.

Types of Expressions

In the book (Asher, 1993), an event is not only denoted by a single word or phrase as annotated in our example. It requires a precise segment of texts related to that event. Thus the following categorization on the types of event antecedent expression can give us some insights of the problem. The name for each type is quite self-explaining. Thus we will elaborate each type by an example.

That Clause

*“John believed **that Mary was sick**. The teacher believed **it** too.”* (Example 2.6)

Infinitival Phrase

*“Fred wanted **to go to the movies**. But his mother wouldn’t allow **it**.”* (Example 2.7)

Gerund Phrase

*“**John’s hitting Fred** got everyone in trouble, for **it** led to a brawl.”* (Example 2.8)

Noun Phrase

*“**The claim that Susan got a C on the test** was surprising. John didn’t believe **it**.”*

(Example 2.9)

Verb Phrase

“Fred **hit a home run**, and then Sally did **it**.” (Example 2.10)

For each mention, its boundary is difficult to determine automatically. Therefore, we decide to define an event coreference mention as a sufficient minimal text expression that is capable for a computational discourse model. Thus in this thesis, only pronouns, noun phrases and action verbs are taken as the event coreference mentions. Several rationales embrace this simplification.

Firstly, determining the event mention boundary is, in general, performed and studied in the event extraction task. Both event coreference resolution and event extraction are individual modules in the whole text processing system. One should not overtake other’s functionalities. By resolving the event coreferences to the anchor² of the event, the event extraction module will be able to provide the relevant text boundary around the anchor.

Secondly, this restriction on thesis scope is in line with the annotation practices in representative annotated corpora such as OntoNotes4.0 which is used in this thesis.

Last but not least, such simplification will reduce the types of mentions to the most fundamental ones which are easier to comprehend. This is helpful for exploring the new and challenging task of event coreference resolution. Table 2.1 shows how this restriction will affect the mention types.

² The anchor of the event is represented by the action verb, minimal text-span of noun phrase or the event pronoun.

Original Type	Original Texts	Simplified Type	Simplified Texts
That Clause	John believed that Mary was sick . The teacher believed it too.	Verb	John believed that Mary was³ sick. The teacher believed it too.
Infinitive Phrase	Fred wanted to go to the movies . But his mother wouldn't allow it .	Verb	Fred wanted to go to the movies. But his mother wouldn't allow it .
Gerund Phrase	John's hitting Fred got everyone in trouble, for it led to a brawl.	NP	John's hitting Fred got everyone in trouble, for it led to a brawl.
Noun Phrase	The claim that Susan got a C on the test was surprising. John didn't believe it .	NP	The claim that Susan got a C on the test was surprising. John didn't believe it ."
Verb Phrase	Fred hit a home run , and then Sally did it .	Verb	Fred hit a home run, and then Sally did it .

Table 2.1: Event Mentions after Restriction

2.4 Related Work

The related work consists of three major parts. The first collection of works focuses on the conventional object coreference resolution. We review them as a closely related task. Certain findings in these works are proven helpful in our study as well. The second part is about event coreference itself. Although we are the first one to give a systematic and in-depth study on this topic, there are a few previous works on some of the sub-problems in our task. The last part summarizes the representative work on the machine learning models used in this thesis. They are the toolkit for event coreference resolution.

2.4.1 Object Coreference Resolution

In this section, we will present four closely related and representative works on object coreference resolution. There are many other works on object coreference resolution (apart from the four works we are going to discuss in this thesis). These four representative works

³ The predicate verb of "be" is considered as anchor for event expressions describing a situation and status.

are selected from a large collection of papers on object coreference resolution. We select these four because we either adapted their frameworks and features or inspired by their ideas and methods. (Soon et al., 2001) presented a general machine learning framework which was followed by many other researchers including us. (Ng and Cardie, 2002a) proposed an instance selection strategy for the general object coreference resolution machine framework. In this work, we proposed a revised training instance selection strategy dedicated for event coreference resolution. (Yang et al., 2006) proposed a way to utilize the structural knowledge embedded in syntax parse tree which inspired us to utilize the same structural knowledge. The last work we presented here is (Nicolae and Nicolae, 2006). Their work proposed a min-cut variation to perform graph partitioning for object coreference resolution. In this work, we have introduced two other graph partitioning approaches which perform well for event coreference resolution. Besides the four works above, readers can always extend their reading for object coreference resolution by referring to the reference section of these four papers.

(Soon et al., 2001) introduced a machine learning framework for mention-pair classifications. Their work proposed generic training and testing procedures to train and apply a machine learning algorithm such as support vector machine and decision tree. Parts of our work here follow these procedures as well.

(Ng and Cardie, 2002a) introduced an instance selection strategy to improve rule-learning based coreference resolution. In this work, we reexamine the scenario for event coreferences and propose a more dedicated strategy for event coreference resolution.

(Yang et al., 2006) introduced the syntactic tree kernel for object pronoun resolution. Inspired by them, we borrow the kernel method into our study and find positive results in several mention-pair classifiers. (Yang et al., 2008) presented a twin-candidate model which is a pair-wise ranking method for object coreference resolution. Such method has been proven helpful in our study as well.

(Nicolae and Nicolae, 2006) introduced a deviation of the Min-Cut graph partitioning algorithm to object coreference resolution. In this thesis, we have examined more graph partitioning approaches such as spectral clustering and random walk partitioning. We have also proposed dedicated techniques to those graph partitioning approaches to boost the performance of event coreference resolution.

2.4.2 Event Coreference Resolution

In this section, we introduce the related works on event coreference resolution. Event coreference resolution is a complicated task which can be further divided into several sub-problems such as event pronoun resolution attempted by (Donna, 2002) and (Müller, 2007); event verb resolution attempted by (Bejan and Harabagiu, 2010) and the untouched event noun phrase resolution. In addition, (Pradhan et al., 2007) is the paper promoting the OntoNotes 4.0 corpus which is the first large corpus that involves event coreference annotation. (Bejan and Harabagiu, 2010) proposed a resolution system on a different coreference corpus. However, based on our observation, their study belonged to a different task (we refer it as cross-document event verb resolution) from the event coreference definition adopted in this thesis. All the previous works on event coreference resolution only focused on one of the sub-problems in a big picture. In comparison to these previous works, our approach is the first systematic study on this topic in the literature. Our work here will be the first

attempt to draw the big picture for event coreference resolution. The details of the above mentioned previous work are presented below.

(Donna K. Byron, 2002) proposed semantic filtering as a complement to salience calculations to resolve event pronouns targeted by us. This knowledge deep approach only works for much focused domains like trains spoken dialogue addressed in the paper with handcrafted knowledge of suitable events for only the ten plus verbs involved. Clearly this approach is not suitable for general event pronoun resolution in news articles for example. Besides, there is also no performance report on event pronoun resolution, thus it is not clear how effective their approach is.

(Müller, 2007) proposed a pronoun resolution system using a set of hand-crafted constraints such as “argumenthood” and “right-frontier condition” together with a logistic regression model based on corpus counts. Their system targeted only three pronouns namely, “*it*”, “*this*” and “*that*”. The event anaphoric pronouns are resolved together with object referential pronouns. This preliminary explorative work only produced 11.94% F-score for event pronoun resolution which demonstrated the difficulties for event anaphora resolution.

(Pradhan, et al., 2007) applied a conventional coreference resolution system to the OntoNotes 1.0 corpus using the same set of features for object noun phrase anaphora resolution. There is no specific performance reported on event anaphora resolution. We think the event anaphors are not correctly resolved in general as the majority of these features are inappropriate for event anaphora resolution according to our investigation.

(Bejan and Harabagiu, 2010) proposed an unsupervised Bayesian model at event coreference resolution. The corpus statistics gathered on their Event Coref Bank V1.0 corpus⁴ shows a more focused corpus on cross-document verb coreference resolution. Only 20.9% (272 out of 1302) of intra-document chains have more than one mention. The intra-document event coreferences appear not well captured due to its corpus design. At the same time, 89.7% (1564 out of 1744)⁵ of the event mentions are verb mentions and none of the mentions annotated is pronoun. These observations are fundamentally different from the OntoNotes 4.0 corpus where only intra-document coreferences are annotated and all the NPs, pronouns and verbs are annotated. All of these factors made the findings in (Bejan and Harabagiu, 2010) more suitable for cross-document verb coreference resolution than the intra-document event coreferences according to (Asher, 1999)’s definition.

2.4.3 Other Related Works

In this section, we briefly introduce the relevant works on machine learning models and frameworks. The selected models are used in this thesis including Support Vector Machine, Latent Dirichlet Allocation, Spectral Graph Partitioning and Random Walk Graph Partitioning. Since this thesis will focus on the event coreference resolution rather than the machine learning. We will only list a few representative works on the above mentioned models. Readers can always extend their readings by referring to the reference section in the following publications.

(Joachims, 1999; 2001) presented the Support Vector Machine (SVM) model for the text classification task. Because of SVM’s robustness and efficiency, we employ

⁴ ECB V1.0 as in (Bejan and Harabagiu, 2010) is available at <http://www.hlt.utdallas.edu/~ady>

⁵ We use WordNet 3.0 first sense to identify verb mentions automatically.

SVM as our main algorithm for all the mention-pair coreference classifiers. Further to that, (Moschitti, 2006) presented a tool to incorporate tree structures into SVM learning which we have borrowed for some of our mention-pair classifiers.

(Luxburg, 2006; Shi and Malik, 2000; Shamir and Sharan, 2002) presented related studies on spectral graph partitioning (a.k.a. spectral clustering) on various tasks as image segmentation and gene clustering. It also comes with an in-depth discussion on the nature of spectral clustering. We have borrowed and adapted this method for our chain formation process with novel techniques to improve clustering results.

(Yeh et al., 2009; Ramage et al., 2009; Huges and Ramage, 2007; Hassan and Radev, 2010) presented works using random walk partitioning for NLP tasks such as semantic similarity, text polarity and semantic relatedness in WordNet and Wikipedia. We are inspired by their approach and adopt random walk partitioning to coreference resolution with necessary modifications and novel enhancements.

(Blei et al., 2003) presented a topic detection model using Latent Dirichlet Allocation (LDA). We adopt LDA in an unsupervised way and make it a very contributive factor to our event mention detection process.

2.5 Chapter Summary

In this chapter, we have derived the definitions for event and event coreference. In addition, we have introduced the different types of relations and expressions of the event coreference. Last but not least, we have presented the related works section to briefly introduce the selected related works. In the next chapter, we will move on to the adaptation of the conventional two-step resolution framework.

Chapter 3: Resolution Framework

Before we introduce our proposed system for event coreference, we would like to revisit the widely used two-step resolution framework for a deeper understanding. Most of the previous coreference resolution systems employ a two-step approach as in (Soon et al., 2001; Nicolae and Nicolae, 2006) and many others. The first step identifies all the pairs of coreferent mentions. The second step forms coreference chains using the coreferent pairs identified in the first step.

Although a handful of single-step frameworks were proposed recently such as (Cai and Strube, 2010), the two-step framework is still widely in use because it has been well-studied. Conceptually, the two-step framework adopts a divide-and-conquer strategy which in turn allows us to focus on different sub-problems at different stages. The mention-pair detection step allows us to employ many features associated with strong linguistic intuitions which have been proven useful in the previous linguistic studies. The later chain formation step allows us to leverage on efficient and robust graph partitioning algorithms, such as the random walk method, used in this thesis. In practice, the two-step framework is also more mature for practical use and has been implemented in a number of standard coreference resolution toolkits widely available such as RECONCILE (Stoyanov et al., 2010) and BART (Versley et al., 2008). Performance-wise, two-step approaches also show comparable performance to single step approaches on some benchmark datasets⁶.

In this paper, we are exploiting a new type of coreference phenomenon with

⁶ (Stoyanov et al., 2010) reported the RECONCILE (two-steps) achieved 74.25% B³ F-score on ACE 2005. (Haghighi and Klein, 2010) using a single-step approach reported 75.10% B³ F-score on the same dataset with the same train/test-splitting. According to our experiences, such a 0.95% difference is not statistically significant. Other single-step works such as (Rahman and Ng, 2009) and (Poon and Domingo, 2008) reported clearly lower B³ F-scores than RECONCILE using the same datasets but different train/test-splitting.

only a few previous attempts. Therefore, we employed the more matured two-step framework with innovative extensions to accommodate the complicated event coreference phenomena. Such a divide-and-conquer strategy will give us more insight for further advancements as well. Figure 3.1 gives an overview of the two-step coreference resolution system.

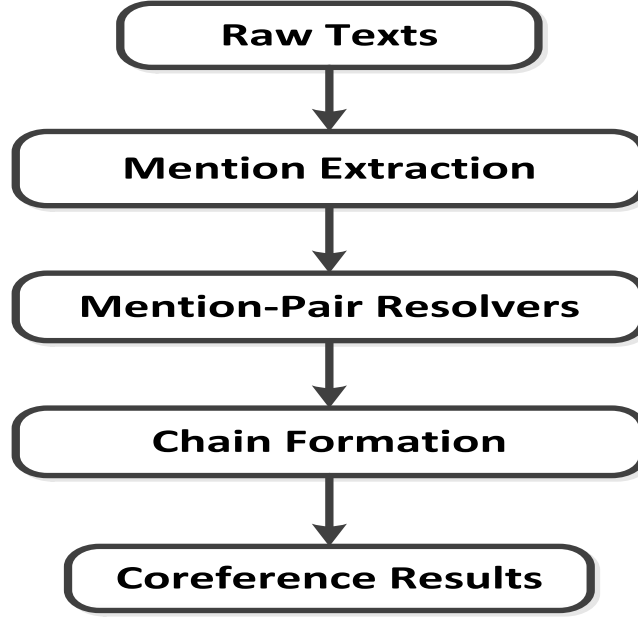


Figure 3.1: Two-Step Resolution Framework

3.1 Mention-Pair Models

Most the mention-pair models adopt the well-known machine learning framework for object coreference as proposed in (Soon et al., 2001) and (Ng and Cardie, 2002a).

3.1.1 Instance Generation

In this learning framework, a training or testing instance of the resolution system has the form of $fv(candi_i, ana)$ where $candi_i$ is the i^{th} candidate of the antecedent of anaphor ana . An instance is labelled as positive if $candi_i$ is the antecedent of ana , or negative if $candi_i$ is not the antecedent of ana .

An instance is associated with a feature vector which records different

properties and relations between *ana* and *candi_i*. The features used in our system will be discussed later in the paper.

During training, for each event anaphor, we consider the preceding event mentions as candidates for being an antecedent. The succeeding verbs are included to accommodate the cataphora phenomenon in which the antecedent occurs after the anaphor. A positive instance is formed by pairing the anaphor with its correct antecedent. At the same time, a set of negative instances is formed by pairing the anaphor with each of its candidates other than the antecedent, which follows the same negative instance selection strategy discussed in (Ng and Cardie, 2002a). (Ng and Cardie, 2002a)’s training instance selection strategy is illustrated in Figure 3.2. Given an anaphor NP₆ and its antecedent NP₃, (Ng and Cardie, 2002a) will generate one positive instance as NP₃---NP₆ and two negative instances as NP₄---NP₆ & NP₅---NP₆. The two NPs (NP₁ and NP₂) beyond the antecedent (NP₃) will not generate any instance.

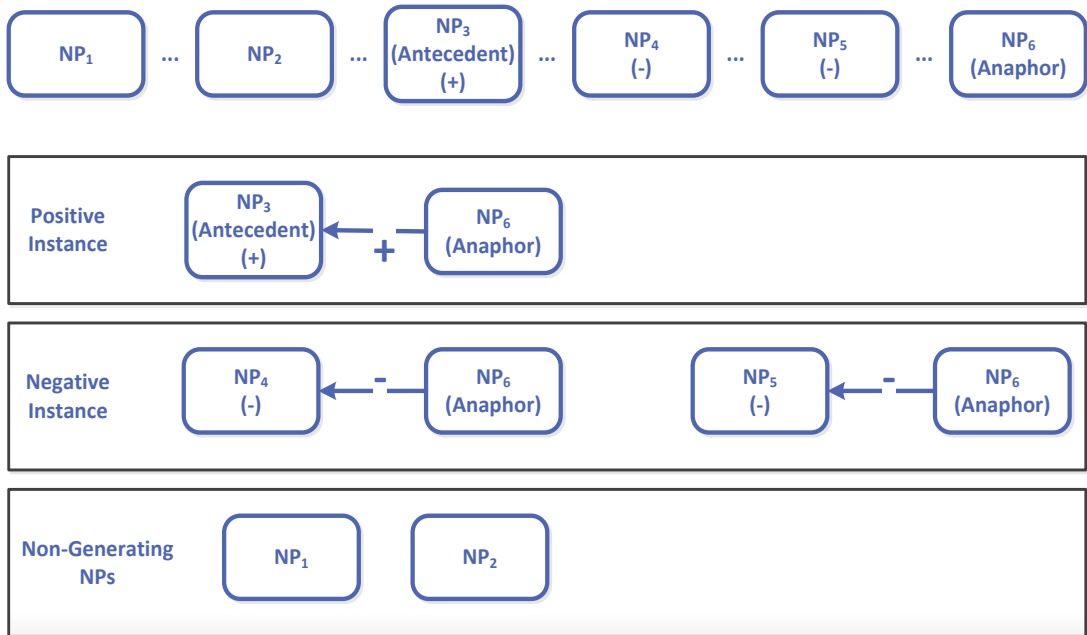


Figure 3.2: (Ng and Cardie, 2002a) Training Instance Selection Illustration

Testing instances are generated in the same manner except that all the preceding event mentions will be considered as candidates.

3.1.2 Learning Models

Based on these generated training instances, we can train a binary classifier using any discriminative learning algorithm. We will present our model in the next chapter.

3.2 Chain Formation Models

After the coreferent mention pairs are identified, coreference chains are formed based on those coreferent pairs. There are two major ways to form coreference chains in the literature: best-link heuristic and graph partitioning.

3.2.1 Best-Link Method

The best-link heuristic selects the candidate antecedent with the highest confidence for each anaphor and forms a “best-link” between them. After that, it simply joins all the mentions connected by “best-links” into the same coreference chain. The best-link heuristic approach is widely used as in (Yang et al., 2006) because of its simplicity and reasonably good performance.

The major criticism of the best-link heuristic falls on its lack of global consideration when forming the coreference chains. Global optimal solution cannot be guaranteed. The mentions are only joined through locally selected “best-links”. Thus chain consistency is not enforced.

3.2.2 Graph Partitioning Method

Graph partitioning approaches are proposed by various researchers to form coreference chains with global consideration. Here we take Best-Cut proposed in (Nicolae and Nicolae, 2006) as a representative of graph partitioning approaches such as hypergraph (Cai and Strube, 2010; Cai et al., 2011) and multigraph (Martschat et

al., 2012). Best-Cut is a variant of the well-known minimum-cut algorithm. A graph is formed using all the mentions as vertices. An edge is added between two mentions if there is a positive output from the mention-pair model. Then the set of edges are iteratively cut to form the coreference chains.

According to (Nicolae and Nicolae, 2006), their approach does not utilize coreferent pairs involving pronouns. However, event coreference chains contain a significant proportion of pronouns (18.8% of the event coreference mentions in the OntoNotes4.0 corpus). Leaving them untouched is obviously not a preferable choice. In Chapters 6 and 7, we will propose an alternative chain formation method to incorporate coreferent pronouns into the graph partitioning process to accommodate its intensive occurrences in event chains.

3.3 Chapter Summary

In this chapter, we have briefly reviewed the conventional two-step framework for object coreference resolution. A bird's view analysis is also given on the conventional chain formation method. The criticisms lead us to our proposed models and techniques to improve the conventional framework.

Figure 3.3 shows the relations between Chapter 3 and Chapters 4, 5, 6, 7. Chapter 3 establishes the general resolution frame work while the other four chapters propose various improvements at their corresponding steps.

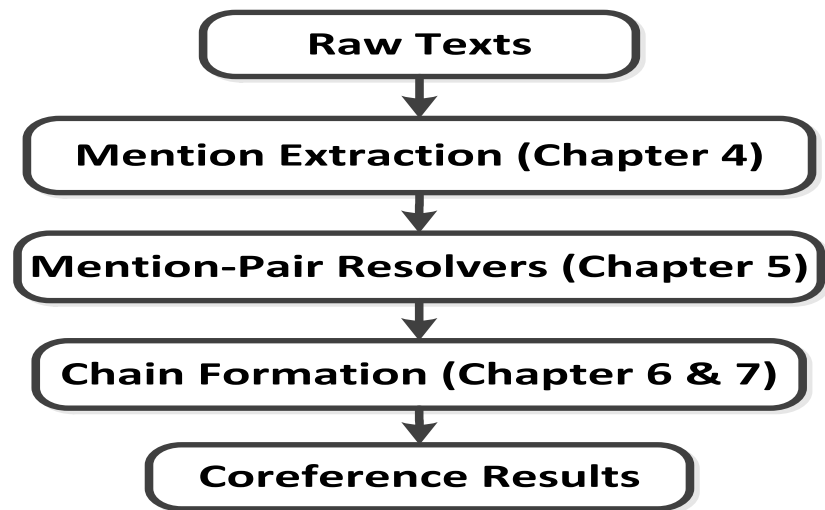


Figure 3.3: Relation among Chapters

Chapter 4: Event Mention Extraction

The very first task of our problem is to extract the event mentions from the texts. The major challenge lies in distinguishing the event mentions from the object mentions. We use the system mentions generated from the syntactic parse tree. Different sets of rules and settings are applied to different syntactic categories such as noun phrase (NP), verb and pronoun. There are different sources of knowledge used for the event mention extraction task. Figure 4.1 below shows an overview of our proposed methods. Recalling from Figure 3.1 which is the overview of the two-step framework, the techniques proposed in this chapter works on the “Event Mention Extraction” step.

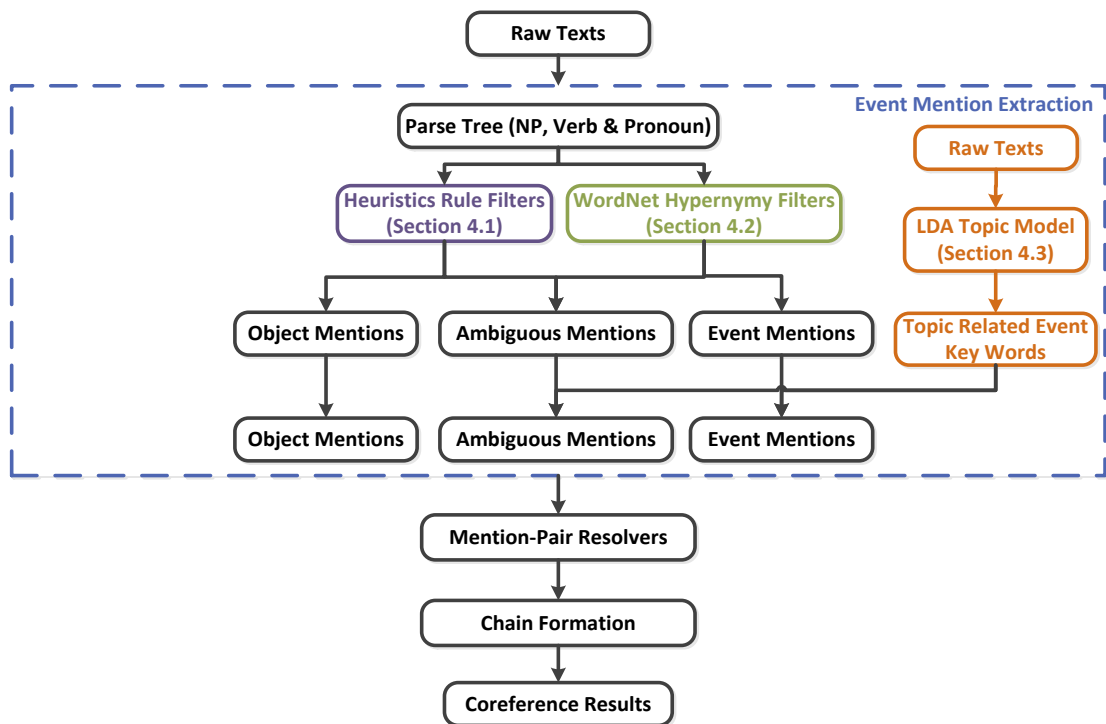


Figure 4.1: Event Mention Extraction Overview

Firstly, from the parse tree of the raw texts, the NP, pronoun and verb mentions are extracted. After that, the heuristic rule filter (Section 4.1) and WordNet hypernymy filters (Section 4.2) are applied to categorize the mentions into “Object”,

“Event” and “Ambiguous” categories. In the next step, the LDA model identifies a list of high priority event word/phrases. Then the categorised mentions will be further refined with the list of high priority mentions from LDA (Section 4.3). As the final results, the refined mentions will be passed to the mention pair resolvers.

4.1 Heuristic-Based Extraction

Heuristics rules are crafted using linguistic intuitions. First of all, all verb mentions (excluding modal verbs) are considered as event mentions. Secondly, since pronouns have too little information to classify them as event-pronouns versus object-pronouns, all the pronouns will be resolved by both event resolvers and object resolvers. Lastly, all the noun phrases ending with “-tion”, “-ing” or “-ment” are considered as event mentions⁷.

4.2 WordNet-Based Extraction

The heuristics in the previous section suffer from low coverage, especially for the case of noun phrases. Therefore, we propose to use the WordNet Hypernymy relation information as a semantic knowledge source for distinguishing event noun phrases and object noun phrases.

All the noun phrases are subject to a categorization as event NPs, object NPs and ambiguous NPs. This categorization is done automatically using its hypernymy information from WordNet⁸. A list of event hypernyms and another list of object hypernyms are collected from the training corpus. If an NP’s hypernym matches event/object hypernym list, it will be classified as an event/object NP. If an NP’s hypernym matches none or both of the event and object hypernym list, it is classified

⁷ In this thesis, we focus in English corpus only. Therefore, the heuristic features are also language-specific.

⁸ Instead of conducting a word sense disambiguation, we use the first sense in WordNet.

as an ambiguous NP. A sample of the selected WordNet hypernymy list is tabulated in Table 4.1.

Event Hypernymy List	Object Hypernymy List
Human Act	Location
Operation	Device
Happening Occurrence	Artifact
Change of State	Living Thing
Killing	Natural Object
...	...

Table 4.1 : Hypernymy List for Event v.s. Object NPs

In total, there are twenty-one hypernyms for event NPs and twenty-seven hypernyms for object NPs⁹. Since the ambiguous NPs may be either event or object, we present them to both event resolvers and object resolvers. In Figure 4.2, we illustrate the WordNet Hypernymy filter selection with an example. Given the mention is “Invasion”, we first obtain its hypernymy hierarchy from WordNet as “Invasion” → “Attack/Onslaught” → “Military_Operation” → “Operation”. Then we cross check with the Event Hypernymy List. Because “Invasion” belongs to “Operation” hypernymy and “Operation” is in “Event Hypernymy List”, mention “Invasion” will be classified as an event mention.

Readers should take note that these twenty-one and twenty-seven hypernymies are not meant to find all the events and objects mentions. In this study we only focus on the event/object mentions which are in the coreference chains. The selected hypernymies are just enough to find those mentions in coreference chains.

⁹ The full hypernym list is given in Appendix A2.

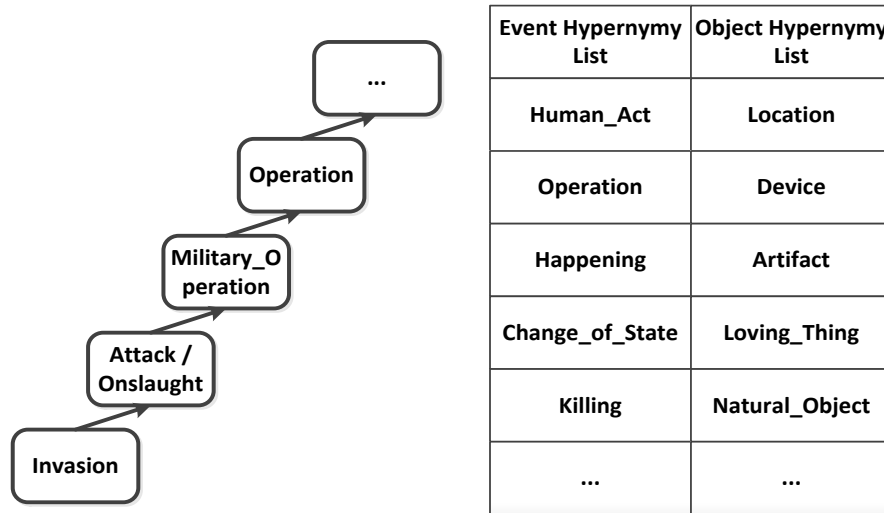


Figure 4.2: WordNet Hypernymy Filters

4.3 Topic-Based Extraction

According to our experiments, one serious problem for the resolution system using the previous event mention extraction methods is the excessive number of false positive predictions of the mention-pair classifiers. Since the event mention identification is done before the coreference resolution, the mention extractor has to emphasize on the recall in order to include most of the event mentions for the mention-pair classifiers. As shown in our observations, the mention-pair models will produce a large number of false positive links. The overwhelming number of false positive links will mislead the chain formation process. Therefore we propose a dedicated event detection module to enhance the prediction of mention-pair classifiers.

In our observation, each article will have its own central topic. Intuitively, the events closely related to the article topic have a higher probability of re-occurring in the article and thus forming coreference chains. For example, if a news article is talking about a large fall in the Dow Jones’s Index of New York Stock Exchange, the event mentions about the “index fall” will re-occur as “fall”, “drop”, “plunge”, “dive”

and their morphological variations. These mentions are highly likely to co-refer with each other. Other event mentions are less likely to be repeated and thus no coreference chains are formed. Therefore we propose to use a topic modelling module to find these high-priority key event mentions for each topic of the articles.

Since the OntoNotes 4.0 corpus is not labelled with article topics, we use Latent Dirichlet allocation (LDA) in an unsupervised way to cluster the articles¹⁰.

4.3.1 LDA-Based Topic Modeling

Latent Dirichlet Allocation as introduced in (Blei et al., 2003) is a three-level hierarchical Bayesian model. Each document is represented as a set of N words, and the collection has M documents. Each word w in a document is generated from a topic distribution, which is a multinomial distribution over words. The topic indicator Z of the word w is assumed to have a multinomial distribution over topics, which in turn has a Dirichlet prior with a hyper-parameter.

In our work, we use the training portion of the corpus to create the LDA model and form a set of document clusters S . Each cluster s in S is considered a topic though we do not have a specific semantically meaningful label (such as “sports” or “stocks”) for it. Each cluster s is associated with a list of words which is used to identify the topic related event mentions. This list is obtained from the LDA output. Since each word is assigned a probability that it belongs to the topic, we rank the list by this probability and use the top fifteen¹¹ words in the list as key words for the documents in cluster s .

During resolution, each testing document is labelled with one of the clusters in S from the LDA model trained above. After that, the associated key word list is used

¹⁰ We use the open source LDA from <http://www.cs.princeton.edu/~blei/lda-c/>

¹¹ The number of top keywords, 15, is empirically selected.

to identify the topic related events. Only those mentions in the key list are considered as an event mention and presented to the mention pair classifiers.

Besides the set of key words, there is a list of common phrases to use as event mentions. Such common phrases include “be”, “state”, “seem”, “say”, “announce” and etc. and their morphological forms¹².

4.3.2 Combined versus Separated Topic Models

In event coreference resolution task, we have to handle the differences between syntactic categories, namely the verbs and noun phrases. Thus, we have two different settings to construct the LDA topic models. The first one simply uses a combined list mixing both verbs and noun phrases. Alternatively, we can construct two separated key word lists for verbs and noun phrases respectively.

Using a single list may suffer from an unbalanced list that contains only one type of the mentions (only verbs or only noun phrases). It will miss a number of event mentions in the article. Since we cannot decide which setting is better theoretically, we find the better setting in an empirical way.

4.4 Chapter Summary

In this chapter, we have a deep look into the event mention detection subtask. At first, we use language specific heuristics filter such as “-tion” and WordNet hypernymy filter to identify event mentions. Furthermore, we propose to use topic-dependent prior event mention list to further refine the extracted mentions. The refined event mentions will be passed to the mention pair resolvers in the next chapter.

¹² The complete list of common phrases is given in Appendix A3

Chapter 5: Mention-Pair Resolvers

After extracting the potential event mentions, we will present how to predict the coreferent mention pairs. In this chapter, we will investigate the phenomenon between different syntactic categories. Figure 5.1 shows an overview of the mention-pair resolvers and our proposed techniques (utilizing competing classifiers' results and a better training instance selection strategy). Recalling from Figure 3.1 which is the overview of the two-step framework, the techniques proposed in this chapter are working on the "Mention-Pair Resolution" step.

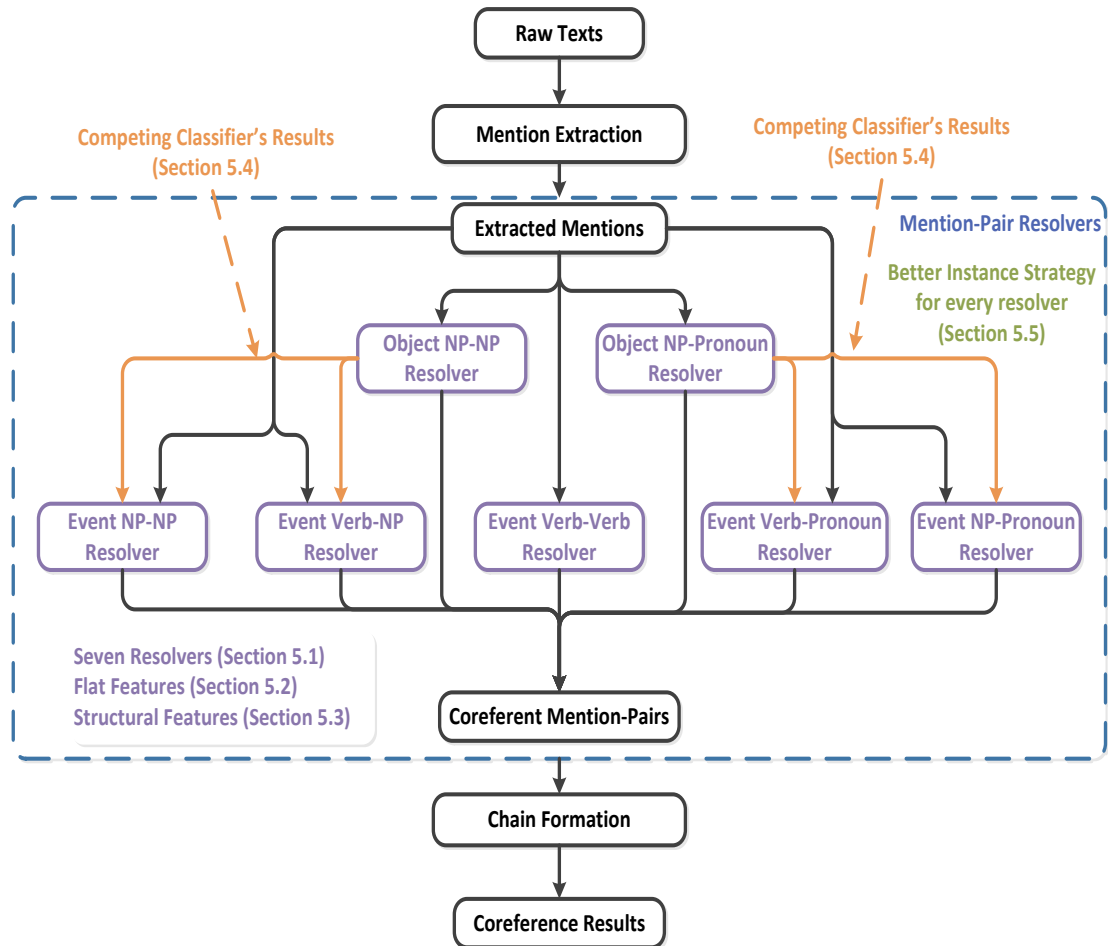


Figure 5.1: Mention Pair Resolvers Overview

5.1 Seven Distinct Mention-Pair Resolvers

One major difficulty of event coreference lies in the gap between different syntactic types of mentions (e.g. nouns, verbs and pronouns). Different syntactic types of coreferent mentions show very different characteristics which require distinct features to resolve them. Following this observation, we have built five distinct resolution models for event coreferences involving noun phrases, pronouns and verbs. They are the Verb-Pronoun, Verb-NP, Verb-Verb, NP-NP and NP-Pronoun resolvers¹³. Conventionally, pronouns can only appear as anaphors but not antecedents. Therefore we do not train Pronoun-Pronoun, Pronoun-Verb and Pronoun-NP resolvers. In addition, we find the effective feature sets for Verb-NP and NP-Verb resolvers are the same. Therefore the Verb-NP resolver will handle the Verb-NP mention pairs in both forward and backward directions.

With respect to the differences between object NPs and Event NPs, we train two distinct models to handle object NP-NP and event NP-NP resolution separately with distinct features. Similarly, we train separate resolvers with distinct features for event/object NP-Pronoun. In total, we have seven distinct mention-pair resolvers for different syntactic and semantic types of mentions. Five of them focus on event coreference while the other two focus on object coreference. The object coreference results are used to enhance event coreference performance by ruling out inappropriate anaphors.

SVM Learning Model

In theory, any discriminative learning algorithm can be used to learn a classifier for pronoun resolution. In our study, we use Support Vector Machine (Vapnik, 1995) to allow the use of kernels to incorporate the structural feature. One advantage of SVM

¹³ The mention-pair resolvers are not sorted according to any order.

is that we can use a tree kernel approach to capture syntactic parse tree information in a high-dimensional space.

Suppose the training set S consists of labeled vectors $\{(x_i, y_i)\}$, where x_i is the feature vector of a training instance and y_i is its class label. The classifier learned by SVM is:

$$f(x) = \text{sign}\left(\sum_{i=1} y_i a_i x \cdot x_i + b\right)$$

where a_i is the learned parameter for a support vector x_i . An instance x is classified as positive if $f(x) > 0$.

5.2 Flat Features

In this section we will investigate the flat feature space for mention-pair resolvers. Flat features refer to the features without structural information. Our investigation starts with an analysis of why the feature set used for conventional object resolution system fails for event coreference resolution (Section 5.2.1). We then move on to understand the syntactic and semantic difficulties with event coreference (Section 5.2.2). After that, we will propose our novel features for the challenging event coreference resolution task (Section 5.2.3.1~5.2.3.12). The proposed feature sets are designed with intuition from various knowledge sources including lexical, contextual, and syntactic and many others.

5.2.1 Failure of Conventional Features

In this section, we will examine the conventional features proposed for object coreference resolution. Table 5.1 gives a list of some features used in conventional noun phrase anaphora/coreference resolution which focus on object entity (Soon et al., 2001; Ng and Cardie, 2002b; Yang et al., 2003; Luo et al., 2004).

However, most of these features are not useful for our task except the shallow positional features. In event anaphora resolution, we focus on event entity instead of real objects. Thus the features describing object characteristics such as number agreement, gender agreement and name alias will no longer function here. Also, our anaphor and antecedent pair consists of a verb and a noun phrase. The difference in word syntactic category will introduce extra difficulties using the conventional lexical features such as string matching and head matching. Furthermore, the difference in word syntactic category will cripple the noun phrase characteristic features for half of the pair. Grammatical features on appositive structure are no longer useful as well.

<i>Conventional Features</i>	<i>Applicable to Event Anaphora Resolution</i>
<i>Object Characteristics</i>	
Number Agreement	No
Gender Agreement	No
Name alias	No
<i>Grammatical Feature</i>	
Appositive Structure	No

Table 5.1: Features for Conventional ObjectNP Resolution

5.2.2 Complications of Event Coreference Resolution

After showing the failure of the conventional features, we conduct an investigation on why the conventional features failed. Event coreference resolution incurs more difficulties as compared to traditional object coreference resolution in two aspects, syntactic and semantic. We will elaborate the difficulties in detail.

Syntactic Difficulties

In a syntactic view, object coreference resolution only involves mentions from the noun category while event coreference involves mentions from verbs as well. This

syntactic difference will cripple the traditional coreference features. The crippled features include mention characteristics features such as “if NP is a proper name”, semantic features such as “number/gender agreement” and grammatical features such as “appositive structure”. In addition, the event NP-Pronoun/NP-NP resolution requires very different linguistic features from the traditional ones. For example, previous semantic compatibility features only focus on measuring the compatibility between object such as “person”, “location” and etc. Event cases generally fall in the “other” category which provides no useful information in distinguishing different events.

Semantic Difficulties

In a semantic view, an object (such as a person, location, organization and etc.) is uniquely defined by its name (e.g. Barack Obama) while an event requires its role information to distinguish itself from other events. For example, “the crash yesterday” --- “crash in 1968” share the same event type, an air crash, but they are likely to be different events by their time argument. Similarly, “murder of Joe” --- “murder of John” and “conflict in Middle East” --- “conflict in Afghanistan” also shares same event types but distinguished by the patient and location arguments respectively.

5.2.3 Features for Event Coreference Resolution

After examining the difficulties in event coreference resolution, in the next several sub-sections, we are going to present the features we selected for our event coreference mention-pair resolvers. Different mention pair resolvers utilize different sets of features. In Table 5.3, the features used for various mention pair resolvers are tabulated. The leftmost column listed the feature groups. The middle column briefly explains the feature group. The rightmost column lists the mention pair resolvers

which the feature is applied to. In the rightmost column, e stands for event; o stands for object. Similarly, V, N and P stand for Verb, Noun Phrase and Pronoun respectively. For example, the second row on “String-Matching” feature group, the feature group is applied to event NP-NP resolver (eNN), object NP-NP resolver (oNN) and Verb-Verb resolver (VV). Similarly, eNP stands for event NP-Pronoun resolver and VN stands for Verb-NP resolver.

In the next twelve sub-sections, we will explain each feature group in detail. The commonly used features such as position features will only be briefly introduced. The dedicated features for event coreference such as the synonymy relation features and Event WordNet Hypernymy features will be explained in more details.

5.2.3.1 Positional Features

A set of positional features is employed in our resolution system. They are employed from the conventional noun phrase anaphora resolution. Positional features measure the separating distance between an anaphor and its candidate in various units. These features are tabulated in Table 5.2. In general, the closer candidate is preferred. These features are especially helpful for pronoun resolution as pronouns usually come with too little information. The notation that those pronouns prefer the closer candidate is a common observation for pronoun resolution. It is also commented in (Yang et al., 2006 and Ng and Cardie 2002b).

<i>Positional Features: M_i: Mention i; M_j: Mention j</i>	
SentDist	# of Sentences between M_i and M_j ;
PhraseDist	# of NPs between M_i and M_j ;
WordDist	# of words between M_i and M_j ;

Table 5.2: Positional Features

Feature Group	Detail	Used in
Positional	Sentence Distance, Word Distance, Phrase Distance;	All
String-Matching	Full-Match, Partial-Match, Head-Match, Contained-In, Cosine Similarity;	eNN, oNN, VV
Grammatical	Subject/Object in main/sub clauses	Except VV
Mention Characteristics (NP Type)	Definite / Indefinite / Proper Name	oNN, eNN, VN, oNP, eNP
Mention Characteristics (Verb Type)	Predicative / Passive / Common	VN, VP, VV
Mention Characteristics (Pronoun Type)	Possessive/Reflexive/Common	oNP, VP, eNP
Mention-Semantic (NE-Semantic)	Named entity semantic type	oNN
Mention-Semantic (WN-Object-Semantic)	WordNet semantic types of object	oNN, oNP
Mention-Semantic (WN-Event-Semantic)	WordNet semantic types of event	eNN, eNP, VN
Fixed-Pairing Feature	Fixed Pairing of two mentions	VN, eNN
Morphological Relation	If anaphor and antecedent are morphological	VN
Synonymy Relation	If anaphor and antecedent share synonym list	eNN, VV, VN
Surrounding Words / POS	Non-stop-words near the anaphor and antecedent	eNP, oNP
Contexts-Information	Non-stop-words near the anaphor and antecedent	eNN, VN, VV
Argument-Matching	Event arguments from pre-modifiers and PP-attachments	VN, VV, eNN
NP-Antecedent	existing NP chains information	VN, eNN, oNN
Structural Information	Minimum-Expansion	Except o/eNN

Table 5.3: Feature List (e: Event; o: Object; V: Verb; P: Pronoun; N: Noun Phrase)

5.2.3.2 String-Matching Features

This group of features measures the lexical similarity between two mentions. There are five features in this group, namely, Exact-Match, Partial-Match, Head-Match¹⁴, One-Contained-Another, and Cosine-Similarity. Table 5.4 elaborates the five features with explanations. Among them, the “Partial-Match” feature is only applied if the two mentions are both multi-word expressions. The “head” of a phrase is extracted from the parse tree using a set of rules. The word vector of a mention is the bag-of-word vector representation of a given mention.

Feature	Evaluation (M_i : Mention i ; M_j : Mention j)
Exact-Match	0: if M_i is different from M_j . 1: if M_i is same as M_j .
Partial-Match	0: if M_i and M_j have no overlapping word. 1: if M_i and M_j have at least 1 overlapping word.
Head-Match	0: if head of M_i is different from head of M_j . 1: if head of M_i is same as head of M_j .
One-Contained-Another	0: if neither M_i nor M_j is a substring of the other. 1: if M_i is a substring of M_j or M_j is a substring of M_i .
Cosine Similarity	The cosine similarity between the word vector of M_i and word vector of M_j .

Table 5.4: String-Matching Features

5.2.3.3 Grammatical Features

This set of features aims to capture the grammatical roles of the anaphor and an antecedent candidate in a sentence. The details of this set of features are tabulated in Table 5.5. These features capture the grammatical preferences for a given anaphor.

¹⁴ The details on how to extract phrase head can be found in Appendix A1.

NP or Pronoun: M	
Sbj_Main	1 if M is subject in main clause; else 0.
Sbj_Sub	1 if M is subject in sub-clause; else 0.
Obj_Main	1 if M is object in main clause; else 0.
Obj_Sub	1 if M is object in sub-clause; else 0.
Verb: V	
Main	1 if V in main clause; else 0.
Sub	1 if V in sub-clause; else 0.

Table5.5: Grammatical Features

5.2.3.4 Mention-Characteristics Features

A set of phrasal characteristic features is employed in our resolution system. They are inspired by conventional noun phrase anaphora resolution. This set of mention-characteristics features includes three sub-categories: namely, NP-type features, Verb-type features and Pronoun-type features. These features are tabulated in Table 5.6 below.

<i>NP Type Features:</i>		<i>M_i: Mention i</i>
NP_i_Def	1 if M_i is definite; else 0;	
NP_i_Demo	1 if M_i is demonstrative; else 0;	
NP_i_First	1 if M_i is the first NP in its sentence;	
NP_i_Prop	1 if M_i is a proper name;	
<i>Verb Type Features:</i>		<i>M_i: Mention i</i>
V_i_Pred	1 if M_i is a predicative verb;	
V_i_Pass	1 if M_i is a verb in passive mode;	
V_i_Comm	1 if M_i is a common verb;	
<i>Pronoun Type Features:</i>		<i>M_i: Mention i</i>
Pr_i_Poss	1 if M_i is a possessive pronoun;	
Pr_i_Refl	1 if M_i is a reflexive pronoun;	
Pr_i_Comm	1 if M_i is a common pronoun;	

Table5.6: Mention Characteristic Features

5.2.3.5 Mention-Semantic Features

Conventional features try to match mentions into semantic categories like person, location, etc. Then, a conventional resolver evaluates the semantic-matching features to pair-up mentions from the same semantic type. In our object resolvers, we also employ this type of semantic features. They help to identify objects which are represented by named entity mentions. Besides the commonly used Named Entity semantic features, we also utilize the WordNet hypernymy information. The WordNet hypernymy information helps to identify object which are represented by the nominal mentions.

However, event NPs exhibit a very different hierarchy in WordNet from the object NPs. A set of dedicated event hierarchy matching features is proposed to match events of the same type. Such rules will match from a WordNet hypernyms to several surface words or sub-hypernyms in the hypernymy hierarchy. For example, mentions under hypernymy class “Communication” will be matched to surface word “say”, “announce” and “tell” or mentions from sub-hypernymy class “transmission”, “mail” and “verbal communication”. These rules are generated from linguistic intuitions and error analysis from our corpus¹⁵.

5.2.3.6 Fixed Pairing Feature

Fixed pairing is a list of common referential usage between an anaphor and its antecedent. For example, “*say --- information*” and “*announce --- statement*” are commonly used in a referential relation. From a linguistic point of view, “*information*” is the patient role in a “*say*” action. The relation between “*say*” and “*information*” is different from the synonymy and morphology relation described previously. The fixed pairing list is automatically generated from the training data by

¹⁵ The full list of Event Incompatibility / Compatibility Rules is given in Appendix A6.

recording any encountered pairs of the head of NP anaphor and its verb antecedent occurring three times or more¹⁶. The feature is 1 if a candidate anaphor pair makes a hit in the pairing list and 0 if the pair does not exist in the pairing list¹⁷.

5.2.3.7 Morphological Features

Morphological features capture the inflectional and derivational relationship between the anaphor and its antecedent candidate, especially for verb-NP pairs. Morphological features help to bridge the gap between different word syntactic categories. This feature represents how close the anaphor and an antecedent candidate are in their meanings. A candidate with an inflectional or derivational relation with the anaphor is preferred over others to be the antecedent. For example, “*confess*” will be a better antecedent choice for “*confession*” compared to other verbs. WordNet is used to find the morphological forms of a given word. This feature is particularly useful for a Verb-NP mention pair because Verb-NP resolver requires morphological information to bridge the semantic gap between verbs and noun phrases.

5.2.3.8 Synonymy Feature

Synonym features are also used to capture the similarity in meanings between the anaphor and its antecedent candidate. For example, “*assault*” is a preferred candidate for anaphor “*attack*” (for the noun category). In the actual resolution, synonyms are also generated from the derivational forms of the anaphor and the candidates. This is to overcome the gap in word syntactic categories between an anaphor and a candidate. A list of synonyms (including synonyms of derivational forms) is generated for an anaphor and its candidate separately. The synonym feature will be evaluated by comparing the two lists. Feature values include cases as “Both are In the others’

¹⁶ The occurring threshold, 3, is empirically selected.

¹⁷ The full list fixed pairings is given in Appendix A5.

synonym **List (BIL)**”, “**One In** the other’s **List (OIL)**”, “**L**ists are **O**verlapping (**LO**)” and “**L**ists are **M**utually **E**xclusive (**LME**)”. These four values are considered as ordinal with a descending order of BIL>OIL>LO>LME. A higher order indicates a more similar word meaning. WordNet is used for synonym list generation. Although this feature is used for both the Verb-NP and event NP-NP resolvers, it is shown to be critical for the Verb-NP mention-pair resolver.

Figure 5.2 illustrates the scenario evaluating the synonymy feature for the pair “attack” --- “assault”. From the WordNet, we first obtain the synonym lists for both mentions “attack” and “assault”. “Attack” has synonyms as “assail”, “lash out”, “contend”, “snipe”, “assault”, “fight” and etc. “Assault” has synonyms as “attack”, “rape”, “violate”, “ravish”, “set on” and etc. There is one mention “assail” in both lists. “Attack” is in “Assault” synonymy list while “Assault” is in “Attack” synonymy list. The final feature value is “**Both are In** the others’ synonym **List (BIL)**”.

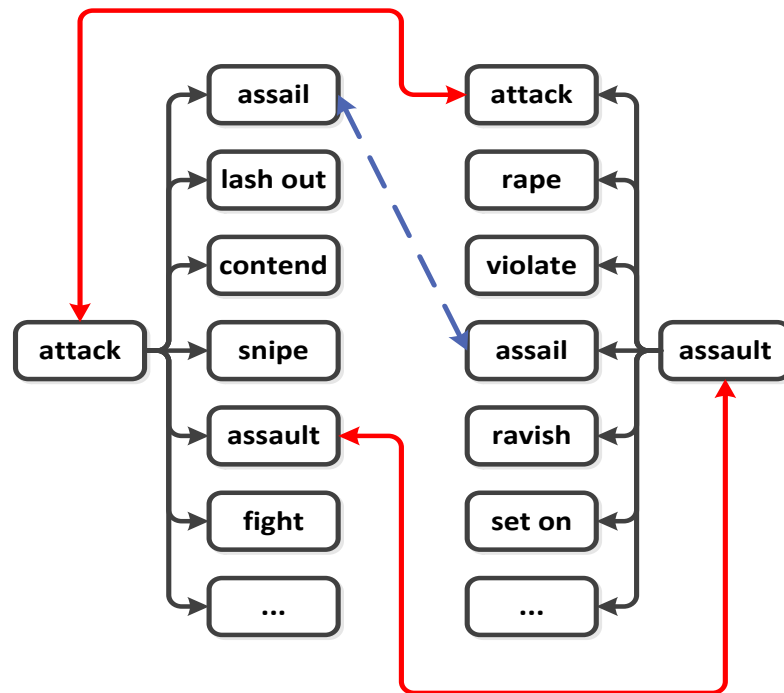


Figure 5.2: Illustration for Synonymy List Feature

5.2.3.9 Surrounding Words/POS Tags and Co-occurrences

This group of features measures the matching of surrounding words/POS-tags and co-occurrences of surrounding words/POS-tags. It consists of two sub-groups of features.

Surrounding Words and POS Tags

The intuition behind this set of features is to find potential surface words that occur most frequently with the positive instance. Since most of verbs occur before the pronoun occurrence, we have built a frequency table from the preceding five words of verb to succeeding five surface words of the pronoun. After the frequency table is built, we select those words with confidence $> 70\%$ ¹⁸ as features. Similar to Surrounding Words, we have built a frequency table to select surrounding POS tags which occur most frequently with a positive instance.

Co-occurrences of Surrounding Words

The intuition behind this set of features is to capture potential surface patterns such as “*It caused...*” and “*It leads to*”. These patterns are associated with strong indication that the pronoun “*it*” is an event type pronoun. The range for the co-occurrences is from preceding five words to succeeding give words. All possible combinations of the word positions are used for a co-occurrence word pattern. For example “*it leads to*” will generate a pattern as “*S1_S2_lead_to*” where *S1* and *S2* mean the succeeding position 1 and 2. Similar to the previously mentioned surrounding words feature, we will compile corpus statistics analysis and select the co-occurrence patterns with a

¹⁸ $Confidence = \frac{\# \text{ of } word_i \text{ occurred with positive instance}}{\# \text{ of } word_i \text{ occurrences}}$. The 70% threshold is empirically selected based on training data.

confidence greater than 70%. Following the same process, we have also examined the co-occurrence patterns for surrounding POS tags.

5.2.3.10 Contextual Information Features

This group of features captures the contextual information using different degrees of matching. There are three sub-groups in this category. The first group measures exact matching while the second group measuring the degree of matching using cosine similarity. The last group measures the identified coreferential relations in context.

Contextual Phrases Features

This group of features measures the similarity and referential relation that exist in the contexts of an anaphor and its candidate. These features are derived based on the following intuitions. First, an event is not only represented by its main action verb, but also the information (e.g. roles of action) extracted from surrounding phrases. Second, when an event is referred in a later occurrence, the related information may reoccur in the contexts. Therefore, this group of feature is designed to capture such knowledge. There are two features in this group.

Context Word Similarity

This feature measures the similarity between an anaphor's context and its candidate's context. Stop words (such as "in", "the" and etc.) are removed from contexts before calculating the similarity. The similarity is calculated based on a window of ± 5 context words. The number of words in common is used to represent the contextual similarity. Inflectional and derivational forms in the contextual words are considered as matching words.

Coreferential Relation in Contexts

This feature is 1 if an object coreferential relation exists between the anaphor's context and its candidate's context. The idea is to capture matching roles of action in two contexts. For example,

“[George W. Bush]₁ {approved}₂ the new military plan {[The president]₁ 's decision}₂ agitated various anti-war groups ...”.

(Example 5.1)

By knowing that *[George W. Bush]₁* and *[The president]₁* corefer with each other, *{approved}₂* is a preferable candidate for *{The president's decision}₂* as they share a common attribute value “*President Bush*”.

5.2.3.11 Event-Arguments Matching Feature

Event NPs have different characteristics from the object NPs. Event NPs require the event roles to distinguish it from other events while the object NPs are quite self-explanatory. The conventional features such as string-matching and head-matching will not work properly when handling cases like “conflicts in Middle East” vs. “conflicts in Afghanistan”. In our approach, a sophisticated argument matching feature is proposed to capture such information. Argument information is extracted automatically from the pre-modifiers and prepositional phrase attachments¹⁹.

5.2.3.12 NP-Antecedent Features

When an NP corefers with a previous one, people will naturally replace the original phrase with a concise expression. By using the full expression from an NP's

¹⁹ More details can be found in Appendix A4.

antecedent, we can obtain extra knowledge for the later concise expression. For the antecedent knowledge of NPs, we use our trained object coreference resolution results for this information. There are three features in this group.

Morphological Feature with NP's Antecedent

This feature is evaluated by comparing each of the NP's antecedents with the verb for an inflectional or derivational relation. It is considered as a morphological relation if one of the NP's antecedents is an inflectional or derivational word from the verb.

Synonym Feature with NP's Antecedent

Similar to the Section 5.2.3.8, the synonym list from the NP coreferential expressions is used to compare with the verb's synonym list. The final feature value is taken to be the highest order as described in the previous section on synonym features.

Named Entity Feature with NP's Antecedent

This feature is used to rule out inappropriate NPs for event anaphoric relation. Consider the object coreferential expressions “George W. Bush” and “the president”. The first one will be marked as named entity but not the latter. By using the object NP's coreference knowledge, we can rule out the inappropriate NP “the president” as it refers to a named entity.

In the above Section 5.2.3, we have explained each group of flat features in details. These features include information from positional, grammatical, syntactic and semantic aspects. Some of them are borrowed from the conventional object coreference resolution such as grammatical and positional features. Some are solely designed for event coreference resolution such as morphological and synonymy

features. After introducing these flat features, we will move on to Section 5.3 which incorporates the structural information in an implicit way.

5.3 Structural Information

A parse tree that covers an event anaphoric noun phrase and its antecedent candidate could provide us much syntactic information related to the pair. The commonly used syntactic knowledge for noun phrase resolution, such as grammatical roles or the governing relations, can be directly described by the tree structure. Other syntactic knowledge that may be helpful for resolution could also be implicitly represented in the parse tree. Therefore, by comparing the common sub-structures between two trees we can find out to what degree the two trees contain similar syntactic information. This can be done using a convolution tree kernel. The value returned from the tree kernel reflects the similarity between the two instances in syntax. Such syntactic similarity can be further combined with other knowledge to compute the overall similarity between two instances, through a composite kernel. Although there are other methods to incorporate the structural information, we choose the convolution tree kernel because it can incorporate the structural knowledge in an implicit way without manual intervention on structural feature formulation. The convolution tree kernel has shown its success as (Moschitti, 2004;2006; and Yang et al., 2006).

Normally, parsing is done at the sentence level. However, in many cases a noun phrase and an antecedent candidate do not occur in the same sentence. To present their syntactic properties and relations in a single tree structure, we construct a syntax tree for an entire text by attaching the parse trees of all its sentences to a pseudo root node. Having obtained the parse tree of a text, we shall consider how to select the appropriate portion of the tree as the structured feature for a given instance. As each

instance is related to a noun phrase and a candidate, the structured feature at least should be able to cover both of these two expressions.

Generally, the more substructure of the tree is included, the more syntactic information is provided. However, at the same time the noise originating from parsing errors will be introduced. In our study, we examine three possible structured features that contain different substructures of the parse tree.

5.3.1 Minimal-Expansion Tree

This feature records the minimal structure covering both the pronoun and the candidate in the parse tree. It only includes the nodes occurring in the shortest path connecting the pronoun and the candidate, via the nearest commonly commanding node. When the pronoun and antecedent are from different sentences, we will find a path through the pseudo “TOP” node which links all the parse trees of the sentences of an article. Considering the following example,

*“This was an all-white, all-Christian community that all the sudden was taken over -- not taken over, that's a very bad choice of words, but **[invaded]** by, perhaps different groups. **[It]** began when a Hasidic Jewish family bought one of the town's two meat-packing plants 13 years ago.”*

(Example 5.2)

The Minimum-Expansion structural feature of the instance {*invaded*, *it*} is circled with a solid line in Figure 5.3. Basically, it consists of a syntactic path connecting the two entities.

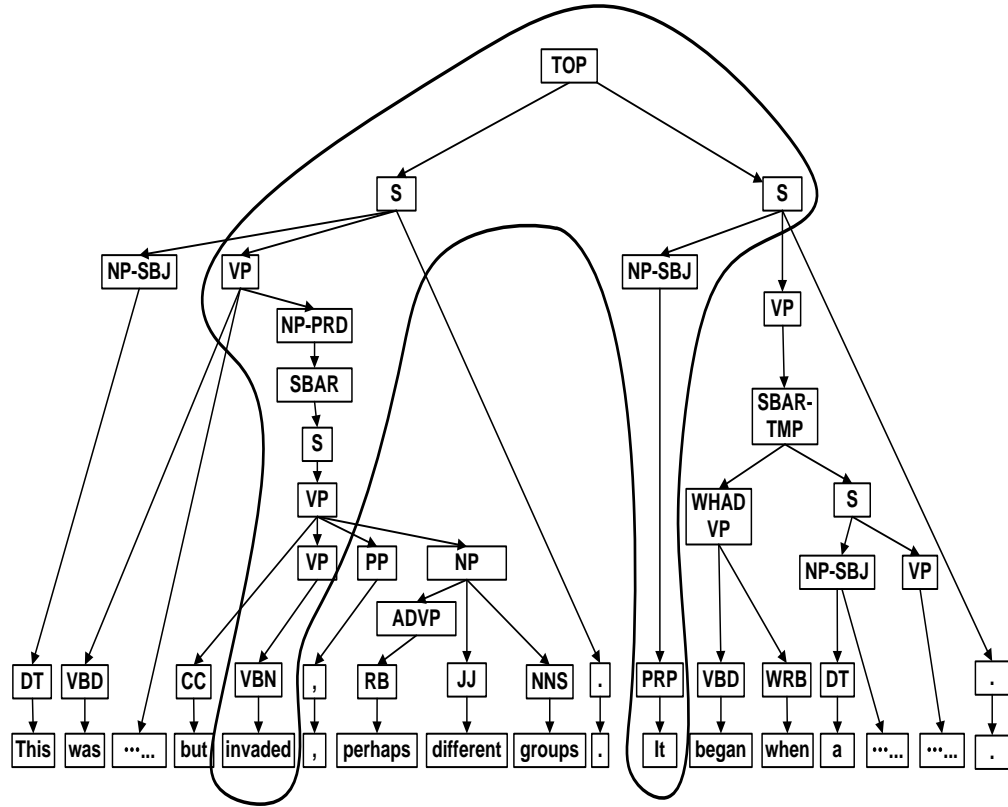


Figure 5.3: Minimum-Expansion Tree

5.3.2 Simple-Expansion Tree

Intuitively, the Minimum-Expansion tree could, to some degree, describe the syntactic relationship between the candidate and the pronoun. However, it is incapable of capturing the syntactic properties of the candidate or the pronoun, because the tree structure surrounding the expression is not taken into consideration. To incorporate such information, the feature Simple-Expansion not only contains all the nodes in *Min-Expansion*, but also includes the first-level children of these nodes²⁰ excluding the punctuation. For the same example above, the simple-expansion structural feature of the instance {*invaded*, *it*} is circled with a dashed line in Figure 5.4. We can see that on the right sentence's tree, "TOP→S→VP" is not further expanded as there are

²⁰ If the pronoun and the candidate are not in the same sentence, we will not include the nodes denoting the sentence before the candidate or after the pronoun.

no more nodes to include below “VP” node. Similarly, the node “NP” for “*perhaps different groups*” is included to provide a clue that we have a noun phrase at the object position of the candidate verb.

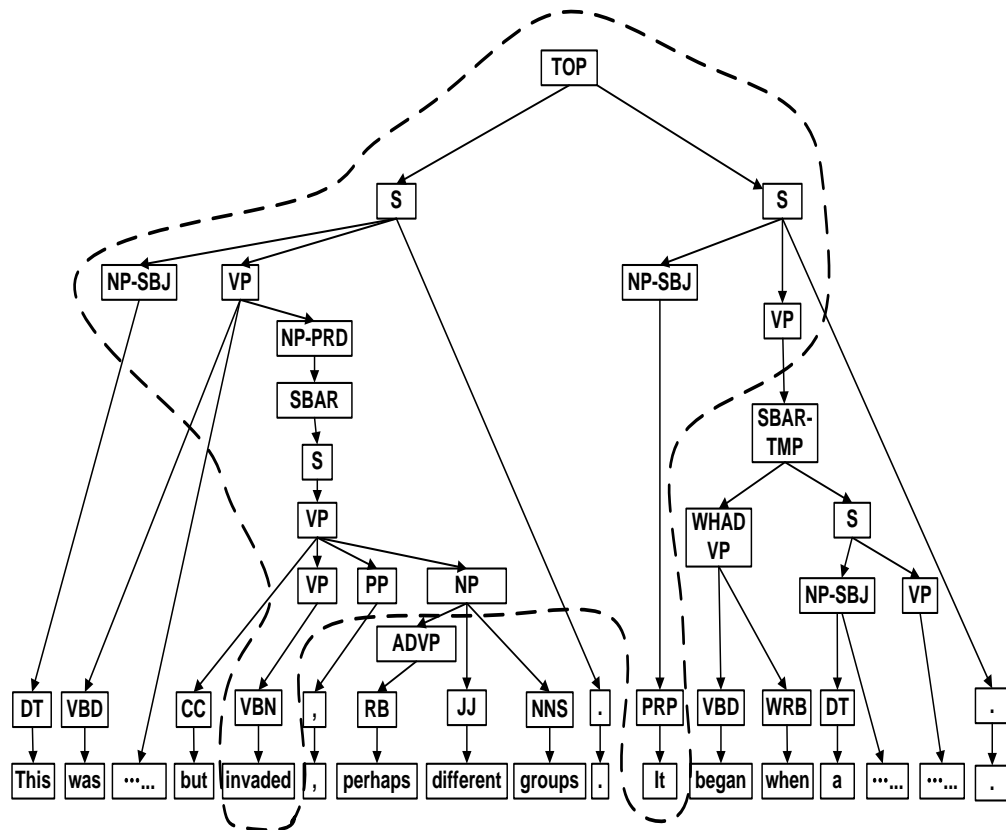


Figure 5.4: Simple-Expansion Tree

5.3.3 Full-Expansion Tree

This feature focuses on the whole tree structure between the candidate and pronoun. It not only includes all the nodes in Simple-Expansion, but also the nodes (beneath the nearest commanding parent) that cover the words between the candidate and the pronoun²¹. Such a feature keeps the most information related to the pronoun-candidate pair in comparison to the other two trees.

²¹ We will not expand the nodes denoting the sentences other than where the pronoun and the candidate occur.

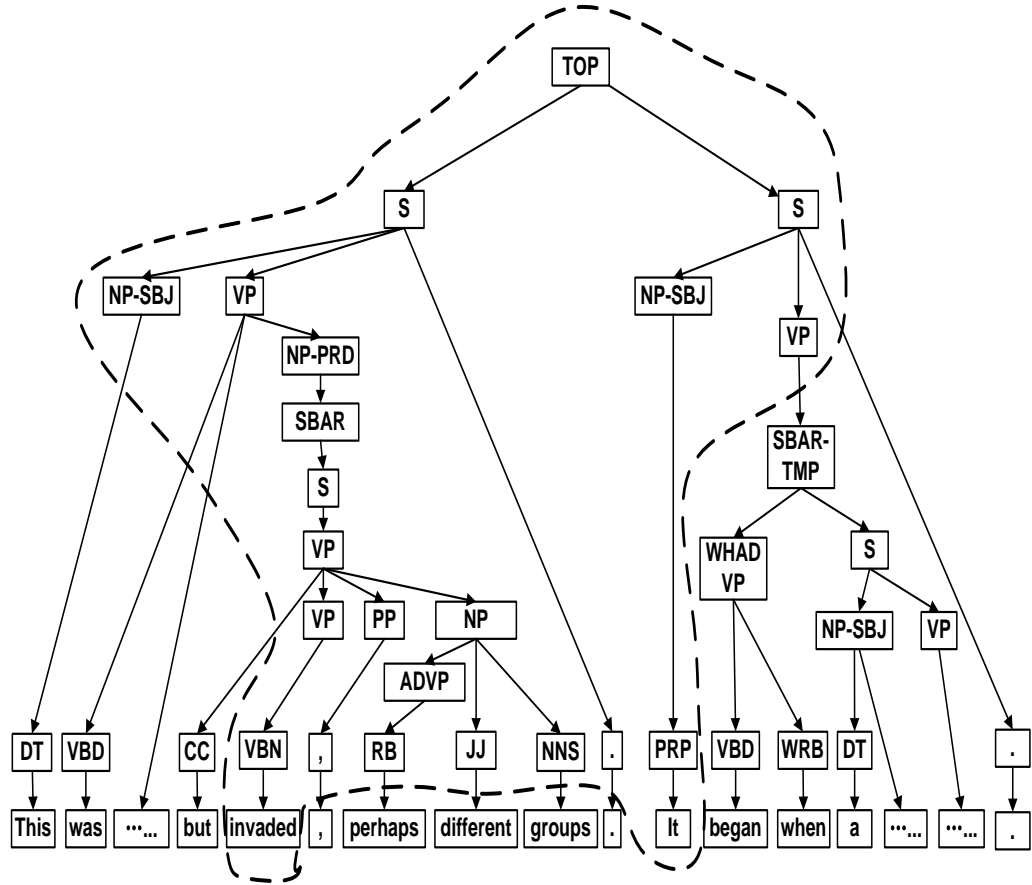


Figure 5.5: Full-Expansion Tree

Figure 5.5 shows the structure for the feature Full-Expansion of the instance $\{\textit{invaded}, \textit{it}\}$. As illustrated, the “NP” node for “*perhaps different groups*” is further expanded to the POS level. All its child nodes are included in the full-expansion tree except the surface words.

In Sections 5.3.1~5.3.3, we have introduced three expansion trees to encode the structural information. From minimum-expansion to full-expansion, more and more contextual and structural information is incorporated. However, more noises are introduced as well. Since we cannot decide which one is better conceptually, we will compare the three expansion trees by their empirical performances in Section 8.4.2.

5.3.4 Incorporate Structural Knowledge through Convolution Tree Kernel

Given structural knowledge in the form of a parse tree, we use the same convolution tree kernel as described in (Collins and Duffy, 2002) and (Moschitti, 2004) to incorporate it into the SVM model in Section 5.1.

Generally, we can represent a parse tree T by a vector of integer counts of each sub-tree type (regardless of its ancestors):

$$\phi(T) = (\# \text{ of subtrees of type } 1, \dots, \\ \# \text{ of subtree of type } i, \dots, \# \text{ of subtree of type } n)$$

This results in a very high dimensionality since the number of different sub-trees is exponential in its size. Thus it is computationally infeasible to directly use the feature vector $\phi(T)$. To solve the computational issue, the tree kernel function is introduced which is capable of calculating the dot product between the above high dimensional vectors efficiently. The kernel function is defined as follows:

$$\begin{aligned} K(T_1, T_2) &= \langle fv(T_1), fv(T_2) \rangle \\ &= \sum_i fv(T_1)[i] \cdot fv(T_2)[i] \\ &= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \sum_i I_i(n_1) \times I_i(n_2) \end{aligned}$$

where N_1 and N_2 are the sets of all nodes in trees T_1 and T_2 , respectively, and $I_i(n)$ is the indicator function that is 1 if and only if a sub-tree of type i occurs with root at node n and zero otherwise.

Collins and Duffy (2002) show that $K(T_1, T_2)$ is an instance of convolution kernels over tree structures, and which can be computed in $O(|N_1| \times |N_2|)$ by the following recursive definitions (Let $\Delta(n_1, n_2) = \sum_i I_i(n_1) * I_i(n_2)$):

(1) if n_1 and n_2 do not have the same syntactic tag or their children are different

then $\Delta(n_1, n_2) = 0$;

(2) else if their children are leaves (i.e. POS tags), then $\Delta(n_1, n_2) = 1 \times \lambda$;

(3) else $\Delta(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + \Delta(ch(n_1, j), ch(n_2, j)))$

where $nc(n_1)$ is the number of the children of n_1 , $ch(n, j)$ is the j^{th} child of node n and λ ($0 < \lambda < 1$) is the decay factor in order to make the kernel value less variable with respect to the tree sizes. In addition, the recursive rule (3) holds because given two nodes with the same children, one can construct common sub-trees using these children and common sub-trees for further offspring.

Besides the above convolution parse tree kernel $K_{tree}(x_1, x_2) = K(T_1, T_2)$ defined to capture the syntactic information between two instances x_1 and x_2 , we also use another kernel K_{flat} to capture flat features.

The syntactic tree knowledge from the tree kernel K_{tree} is combined with the flat feature kernel K_{flat} linearly:

$$K_{comp}(x_1, x_2) = K_{tree}(x_1, x_2) + K_{flat}(x_1, x_2)$$

Both of the kernels are normalized by:

$$K(x_1, x_2) = \frac{K(x_1, x_2)}{\sqrt{K(x_1, x_2) \cdot K(x_1, x_2)}}$$

5.4 Utilizing Competing Classifiers' Results

For the same mention, different mention-pair resolvers will resolve it to different antecedents. Some of these resolution results contradict each other. In the following example:

“USA Today reports {some evidence} that has been uncovered shows Bin Laden financed [the attack] and assigned one of his top assistants to supervise [it].”

(Example 5.3)

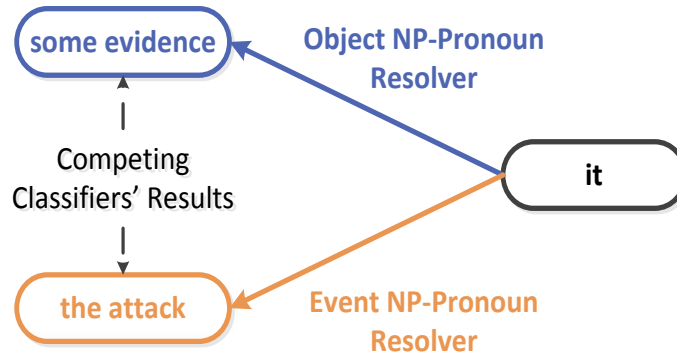


Figure 5.6: Competing Classifiers' Results

Figure 5.6 shows the competing relation of the example above. For the anaphor *[it]*, the event NP-Pronoun resolver may pick *[the attack]* as the antecedent while the object NP-Pronoun resolver may pick *{some evidence}* as the antecedent. Instead of choosing one as the final resolution result from these contradicting outputs, we feed the object resolver results into the event resolvers as a feature and re-train the event resolvers²². The idea is to provide the learning models with a confidence on how likely the anaphor refers to an object.

5.5 Better Instance Selection Strategy

As we mentioned previously, the traditional training instance selection strategy as in (Ng & Cardie, 2002) has a significant weakness. The original purpose of mention pair resolvers is to identify any two coreferent mentions (not restricted to the closest one). By using the previous training instance selection strategy, the selected training

²² The SVM-outputs from object resolvers are transformed into a confidence value in the range of $[-1, 1]$. The transformation is done using a sigmoid function. After that the confidence values are used as a feature for event resolvers.

instances actually represent a sample space of locally closest preferable mention vs. locally non-preferable mentions. In most of previous works, it shows a reasonably good performance when using the “best-link” chain formation technique. Our empirical investigation (in Section 8.4.2) shows it actually misguided the graph partitioning methods. Therefore, we propose a revised training instance selection strategy which reflects the true sample space of the original coreferent/non-coreferent status between mentions. In brief, our revised strategy exhaustively selects all the coreferent mention-pairs as positive instances and non-coreferent pairs as negative instances regardless of their closeness to the anaphor. Consider the following example:

“...linking {Saudi terrorist Osama Bin Laden} to [the bombing]. {USA Today} reports {some evidence} that has been uncovered shows {Bin Laden} financed [the attack] and assigned one of his {top assistants} to supervise [it].”

(Example 5.4)

	Conventional Strategy	Our Strategy
<i>Positive Instances</i>	<i>[the attack]–[it]</i>	<i>[the attack]–[it]</i>
		<i>[the bombing]–[it]</i>
<i>Negative Instances</i>	<i>{top assistants}–[it]</i>	<i>{top assistants}–[it]</i>
		<i>{Bin Laden}–[it]</i>
		<i>{USA Today}–[it]</i>
		<i>{some evidence}–[it]</i>
		<i>{ Saudi terrorist Osama Bin Laden }–[it]</i>

Table 5.7: Better Instance Selection Strategy

In Table 5.7, the traditional instance selection scheme will only select *[the attack]–[it]* as a positive instance and *{top assistants}–[it]* as a negative instance. Our

revised instance selection scheme will select an additional positive instance [*the bombing*]-[*it*] and additional negative instances such as {*Bin Laden*}-[*it*], {*USA Today*}-[*it*] and other NP mentions in curly brackets. Thus the full sample space is represented using our training instance selection strategy.

5.6 Chapter Summary

In this chapter, we have elaborated the seven mention pair resolvers in details. Multiple groups of features are incorporated into the mention pair resolvers including positional, grammatical, and structural and many others. On top of the carefully designed features for each individual resolver, we also propose two methods to improve the mention pair resolution performance. The first method is to utilize the competing classifiers' results which improve the event resolvers using object probability of a given mention. The second method is a revised training instance selection strategy which helps to produce better chain formation results.

After identifying the coreferent mention pairs, we will move on to form the coreference chains using the coreferent mention pairs. In the next two chapters, we will introduce two graph partitioning approaches to form the coreference chains. The two methods have their own advantages and disadvantages. The first one is spectral graph partitioning which is introduced in Chapter 6. The second one is random walk graph partitioning which will be introduced in Chapter 7.

Chapter 6: Chain Formation using Spectral Graph

Partitioning

After obtaining the coreferent mention pairs in Chapter 5, we will move on to the chain formation step. The first method we proposed is spectral graph partitioning. The very first reason to use this method is its robustness and efficiency in computation. In addition, spectral graph partitioning also enables us to conveniently incorporate the pronoun coreference information. Furthermore, we proposed techniques to employ the linguistic knowledge to improve clustering results. Figure 6.1 shows an overview of spectral graph partitioning model and our proposed improvement techniques. Recall from Figure 3.1 which is the overview of the two-step framework, the techniques proposed in this chapter concern the “Chain-Formation” step.

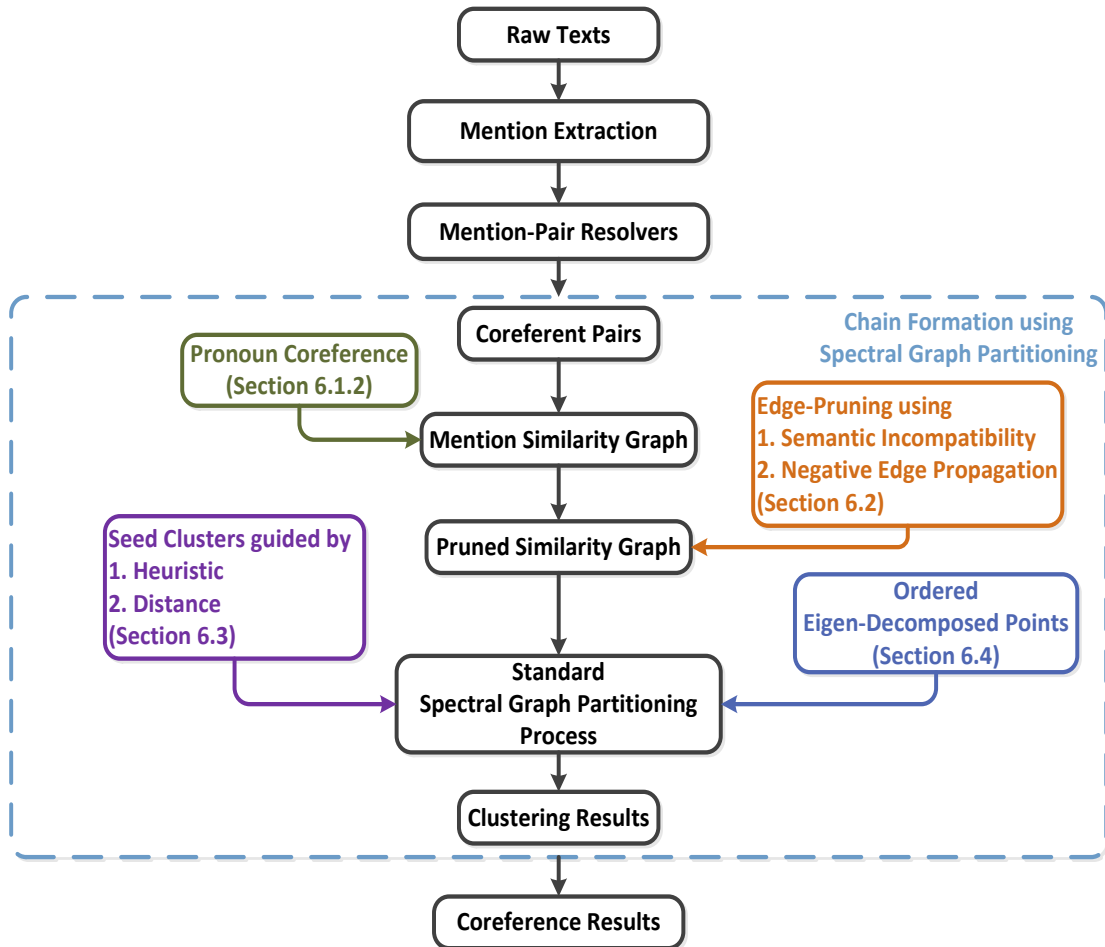


Figure 6.1: Overview of Spectral Graph Partitioning

6.1 Brief Introduction on Spectral Graph Partitioning

Spectral graph partitioning (also known as spectral clustering) has made its success in a number of fields such as image segmentation (Shi and Malik, 2000) and gene expression clustering (Shamir and Sharan, 2002).

Compared to the “traditional algorithms” such as k -means or minimum-cut, spectral clustering has many fundamental advantages. Results obtained by spectral clustering often outperform the traditional approaches, and spectral clustering is very simple to implement and a clustering can be found efficiently by standard linear algebra methods. More attractively, according to (Luxburg, 2006), spectral clustering does not intrinsically suffer from the local optima problem.

6.1.1 Applying Spectral Graph Partitioning to Event Coreference Resolution

After deriving the potential coreferent mention pairs using the SVM classifiers, we further use spectral graph partitioning to form the globally optimized coreference chains. The spectral clustering process is illustrated in Figure 6.2.

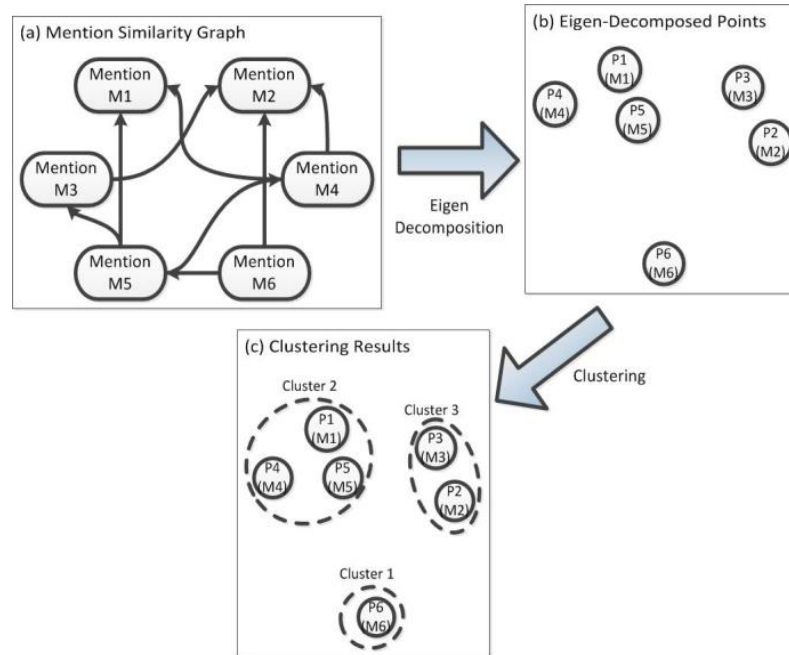


Figure 6.2: Spectral Graph Partitioning Process

The similarity graph is formed using the SVM confidence²³ outputs. The nodes of the similarity graph are the event mentions in the document (Mentions M1-M6 in Figure 6.2: Spectral Graph Partitioning Process). The edges are the positive coreferent links identified by the mention-pair classifiers. The weight of an edge is the corresponding SVM confidence output²⁴.

After forming the similarity graph, we obtain the Eigen-vectors of the matrix representation of the similarity graph. The Eigen-vectors are ranked according to their corresponding Eigen-values following a descending order. After that, we use the top eight Eigen vectors to create the Eigen-decomposed points for the event mentions²⁵. Each of the original event mentions is represented as an Eigen-decomposed point which is denoted by a coordinate in the transformed Euclidean space.

After representing each event mention as an Eigen-decomposed point, the clustering is then conducted by using the Euclidean distance between the points. The points within a radius r is clustered together to form an event chain²⁶. We did not use the conventional k-mean clustering because of two reasons. First, the performance of k -means greatly depends on the choice of k . However, the choice of k (corresponding to number of events in a document) is hard to decide. Second, we are only interested in the event mentions. Therefore, we only pick the high priority event mentions and group the mentions in close proximity together as one event. Our modified version of spectral graph partitioning can avoid the hard decision of k and concentrate on the event mentions. At the same time, the computational complexity is also reduced as we

²³ Confidence is computed from kernel outputs using the sigmoid function.

²⁴ We consider an edge is positive if its SVM output is positive. (the corresponding confidence value will be > 0.5)

²⁵ Eight is selected empirically through corpus investigation of the training set. More details can be found in Appendix B1.

²⁶ The radius r is empirically defined using the training part of the corpus. More details can be found in Appendix B1.

can ignore a large number of object mentions in the original mention graph. Figure 6.3 gives the pseudo-process of our spectral graph partitioning²⁷.

-
- Given a set of points (mentions) $S = \{s_1, \dots, s_n\}$ in R^l that we want to cluster into at most k subsets:
1. Form the similarity matrix (affinity matrix) $A \in R^{n \times n}$ where A_{ij} is the SVM confidence value between s_i and s_j ;
 2. Define D to be the diagonal matrix whose (i, i) -elements is the sum of A 's i^{th} row, and construct matrix $L = D^{-1/2} A D^{-1/2}$ ²⁸;
 3. Find x_1, x_2, \dots, x_k , the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form matrix $X = [x_1 x_2 \dots x_k] \in R^{n \times k}$ by stacking the eigenvectors in columns;
 4. Form the matrix Y from X by renormalizing each X 's rows to have unit length (i.e. $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}$);
 5. Treat each row of Y as a point in R^k , from the select event points (mentions) group the points in close proximity of an event points as an event cluster.
-

Figure 6.3: Algorithm of Our Spectral Graph Partitioning

6.1.2 Incorporating Pronoun Coreference Information

Besides the advantages of spectral graph partitioning model above, one particular reason to employ it is that the previous best-cut approach failed to incorporate pronoun information in their similarity graph. It may not be an issue in object coreference as pronouns comprise only a relatively small proportion (9.78% of the object mentions in OntoNotes 4.0 are pronouns). However, in event coreference, pronouns contribute to 18.8% of the event mentions. As we further demonstrate in our corpus investigation, event chains are relatively more sparse and shorter than object chains. In fact, a significant proportion of the event chains consists only two mentions: the pronoun and its verb/NP antecedent. Removing pronouns from the similarity

²⁷ More details about spectral graph partitioning can be found in Appendix A9.

²⁸ Readers familiar with spectral graph theory may be more familiar with the Laplacian $I-L$. However as replacing L with $I-L$ would complicate our later discussion, and only changes the eigenvalues (from λ_i to $1-\lambda_i$) and not the eigenvectors, we instead use L .

graph will break a significant proportion²⁹ of the event chains. Thus we propose this spectral graph partitioning approach to overcome this inappropriateness of the previous models.

In this work, we propose several sophisticated enhancements to make spectral clustering a more capable method for event coreference resolution. These enhancements utilize semantic knowledge and model characteristics of spectral clustering.

In the following three sub-sections, we will present our proposed techniques. Pruning the inappropriate edges utilizes semantic and linguistic knowledge. Since we found the performance of spectral clustering is affected by the ordering of points and the existence of seed clusters by prior knowledge, we propose to form the seed clusters using two kinds of knowledge source and ordering the Eigen-decomposed points to utilize this model-specific characteristic of spectral clustering.

6.2 Pruning of Inappropriate Edges

The first technique we propose is to prune the inappropriate edges in the similarity graph. The pruning is conducted by two kinds of heuristics. The first one is semantic incompatibility. The other is one-step negative edge propagation. This technique works on the similarity graph corresponding to Stage (a) in Figure 6.2.

6.2.1 Eliminating Semantic Incompatibility

Semantic incompatibility rules are used to eliminate inappropriate edges between incompatible mentions in the similarity graph. Although a number of features in the mention-pair models are designed to capture such incompatibilities, SVM can only produce soft constraints. Thus, the hard constraints such as semantic incompatibility

²⁹ According to our observation, 34.6% of the event chains will be broken if pronouns are ignored from chain formation.

cannot be enforced directly in the SVM models. The semantic incompatibility rules here are an enforcement of the hard constraints to overcome the shortcoming of the SVM model. To accommodate more sophisticated incompatibility constraints, we designed the constraints not only using the surface words of the event mentions, but also the WordNet hypernymy relations of the event mentions. For example, the mention “commence” should not be linked with any mentions under the hypernymy “communication”. There are a total of twenty-eight of such rules created from the error analysis on training data³⁰.

Furthermore, the event arguments are checked for the two event mentions. This rule is to filter out cases as “conflicts in Middle East” vs. “conflicts in Afghanistan”. The role of the argument is decided by syntactic heuristics. Only location and date-time arguments are matched.

6.2.2 Propagating the Negative Edges

As we mentioned above, the positive outputs from SVM models are used to form the positive edges in the similarity graph. We utilized the strong negative SVM outputs as negative edges to prune inappropriate positive edges. The strong negative edges are propagated one step to detect the potential false positive edges. This pruning is particularly effective for the less informative mentions such as pronouns and short noun phrases.

For example, the event pronoun “it” can be resolved to two verb mentions “attack” and “announce” by the Verb-Pronoun resolver. The SVM confidence output for edge “attack”---“it” is 0.8 and that for edge “announce”---“it” is 0.6. The situation is illustrated in Figure 6.4.

³⁰ The full list of Incompatibility Pruning Rules can be found in Appendix A7.

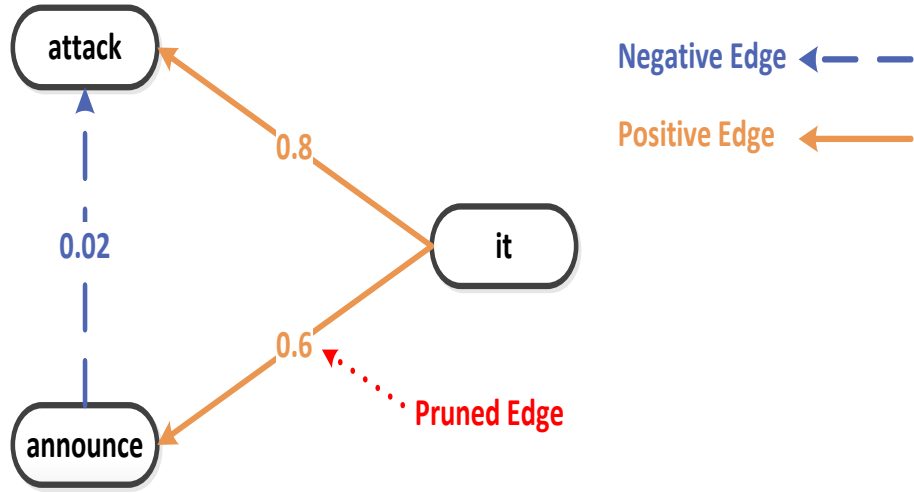


Figure 6.4: Negative Edge Propagation

If we use only the positive SVM outputs, both edges “attack”---“it” and “announce”---“it” are included in the similarity graph. However, the SVM output from Verb-Verb resolver for edge “attack”---“announce” may be a negative value and produce a low confidence of 0.02. By propagating this negative edge “attack”---“announce”, we know there is a conflict between the two edges “attack”---“it” and “announce”---“it”. Thus we will choose only the “attack”---“it” edge with a higher confidence and prune the edge “announce”---“it” from the similarity graph.

6.3 Seed Cluster Creation

The second technique we proposed to enhance the spectral clustering process is to form seed clusters before running the clustering. The seed clusters are created using two kinds of heuristics. The first kind is formed using semantic knowledge. The second kind is formed using the proximity of the points. This technique works on the Eigen decomposed points corresponding to Stage (b) in Figure 6.2.

6.3.1 Knowledge Guided Seed Clusters

This kind of rule is used to create the seed clusters using semantic knowledge. As we mentioned, the SVM model cannot enforce hard constraints. It cannot guarantee to link two mentions although they can be identified using semantic knowledge.

The rules we used here include two types:

1) Fixed pairing of head words:

A list of fixed pairing of head-words is collected from training corpus. These rules are used to link mentions like “say”---“statement”. These word pairs generally cannot be resolved by other features we have employed. Therefore we create these word pairs from error analysis of the training data as predefined background knowledge. There are in total eighteen such pairs³¹. The set of fixed-pairing words are the same as in Section 5.2.9.

2) Compatible Event Arguments with compatible head word

Two mentions are linked together if the head words are synonyms of each other and have at least one compatible event arguments.

These heuristically formed clusters are very useful to connect event chains that are separated as several small clusters after spectral decomposition. This separation may be caused by the n -sentence window we assigned for mention-pair resolvers. Thus if the same event chain is separated as two or more clusters after spectral decomposition, we need these heuristic rules to join them back. The effect of such seed clusters is demonstrated in Figure 6.5(a).

³¹ The full list of Fixed Pairing Words can be found in Appendix A5.

6.3.2 Proximity Guided Seed Clusters

This heuristic will form the clusters of points very close to each other. We empirically choose a small radius e as 0.1×10^{-4} ³². All the points that can be fitted into such a small cluster will be formed before running the clustering of all points. For example, in Figure 6.2(b), P1 and P5 are very close to each other, so they will form a seed cluster before running the clustering step. These close distance points are usually cases of very strong coreference pairs such as multiple mentions of “the confession”. These points should form a cluster. However, the order of points in the clustering process will affect the final results. These points may be accidentally separated. Therefore the distance guided seed clusters will prevent such drawbacks and lead to a better clustering results. The effect of the distance-guided seed clusters is illustrated in Figure 6.5(b).

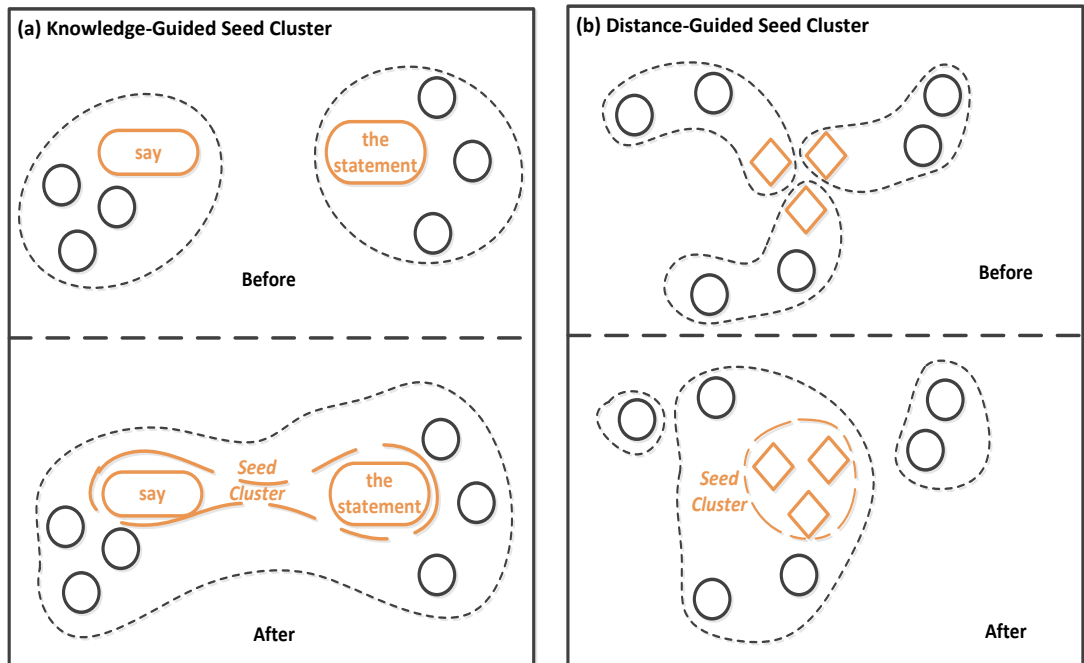


Figure 6.5: Results Before and After Applying Seed Clusters

³² The value of e is empirically chosen by analysis of the training corpus. More details can be found in Appendix B1.

6.4 Ordering of Eigen-Decomposed Points

After the Eigen-decomposition of the Laplacian matrix of similarity graph, we represent each event mention as a point in the Euclidean space. Though we simplify the clustering process by this decomposition, we also lose textual meaning of the event mentions. In order to make up for such loss, we ordered the decomposed points by their textual expression.

The ordering is done in the following way:

- 1) The verbs points are put in front of the noun phrases and pronouns;
- 2) The noun phrases points are put in front of the pronouns;
- 3) When comparing between two verbs or two noun phrases, the point with longer mention string is put in front of the shorter one.

By putting the verbs in the beginning of the list, each different event chain will have a verb to form its cluster. The pronouns are put at the end of list and are prohibited from creating new clusters because they carry very little information. NPs are ranked by their string length by assuming longer strings convey more information. This technique also works in Stage (b) of Figure 6.2. The ordering effect is elaborated in Figure 6.6.

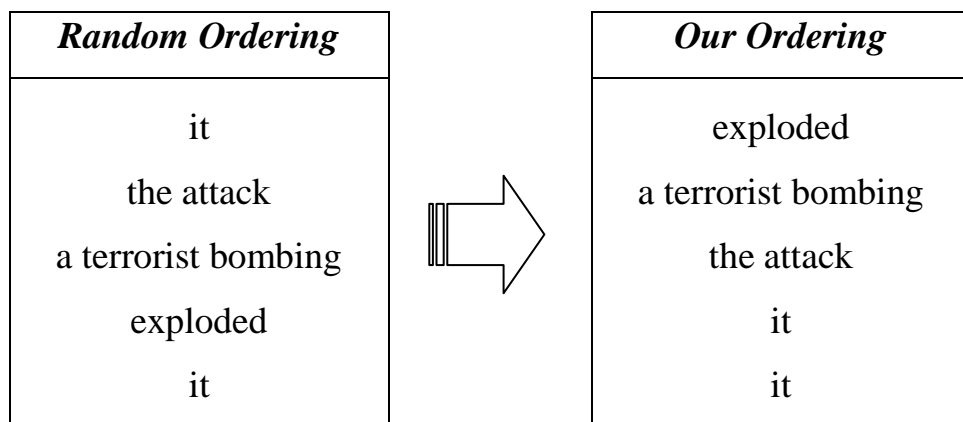


Figure 6.6: Ordering of Points

6.5 Chapter Summary

In this chapter, we have shown how to form the event chains using spectral graph partitioning method. In addition, we proposed to improve the chain formation performance by utilizing the pruning of the inappropriate edges, the seed clusters and the ordering of the decomposed points. These methods incorporated the linguistic knowledge and proximity heuristics.

In the next chapter, we will introduce another chain formation approach using the random walk graph partitioning. With the new approach, we are able to utilize the linguistic knowledge in a dynamic way.

Chapter 7: Chain Formation through Random Walks

In Chapter 6, we have discussed a chain formation technique using spectral graph partitioning. In this chapter we will present another chain formation method by random walks through the mention graph. Although spectral graph partitioning shows its advantages in the chain formation task, the random walk graph partitioning model demonstrates its own specialties in various ways:

Firstly, in the modeling aspect, it can achieve similar results as spectral graph partitioning. Secondly, it can model certain corpus statistical knowledge through the terminating criteria and the termination probability. Thirdly, it can incorporate the relevant linguistic knowledge as constraints and preferences. Instead of the static usage of such knowledge in spectral graph partitioning, they are imposed dynamically in the random walks model. Furthermore, the random walk model enables consistency checking at the chain level instead of at the mention-pairs level in spectral graph partitioning. Last but not least, the random walk model further employs the object mention nodes to prune the inappropriate chains.

The overview of the random walk mode (with necessary modifications) and our proposed techniques (incorporation of linguistic constraints and preferences and pruning using object mention graph) is illustrated in Figure 7.1. Recall from Figure 3.1 which is the overview of the two-step framework, the techniques we propose in this chapter work on the “Chain-Formation” step.

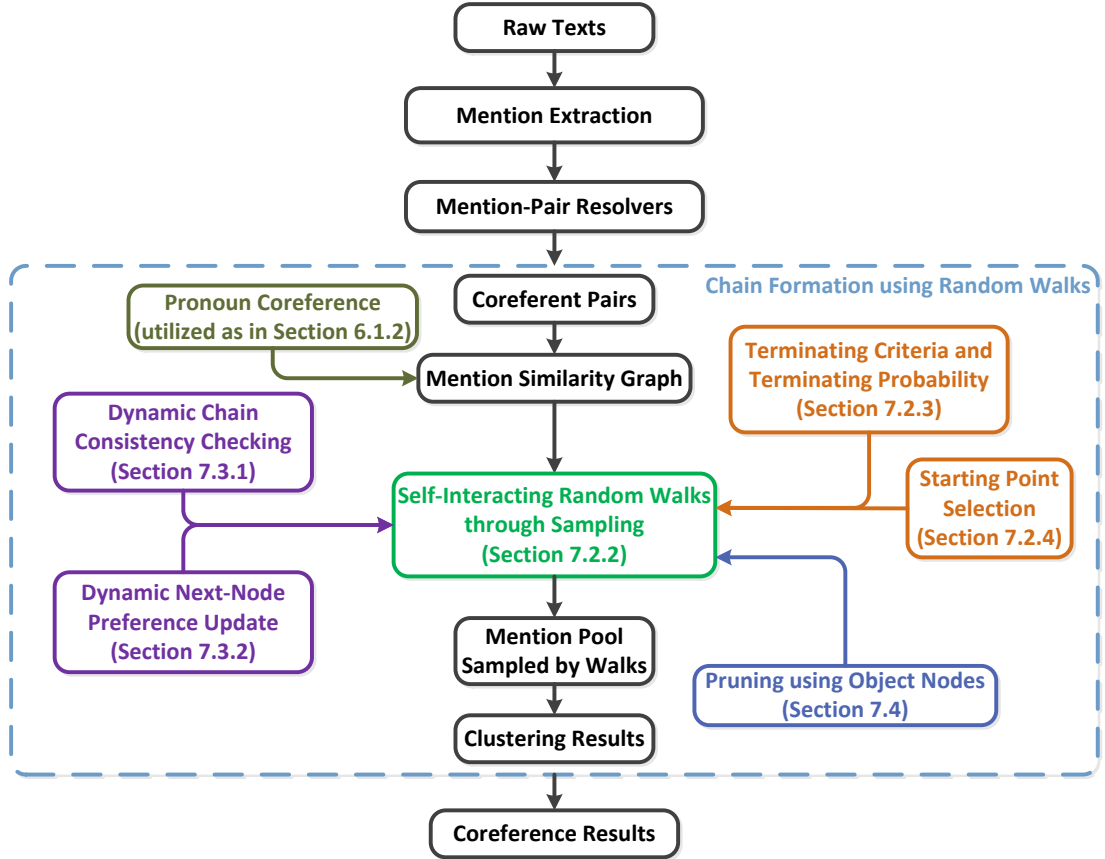


Figure 7.1: Overview of Random Walk Model

7.1 Brief Introduction to Random Walk

The random walk model has made its success in several NLP applications such as polarity classification (Hassan and Radev, 2010), semantic similarity (Ramage et al., 2009) and semantic relatedness (Hughes and Ramage, 2007; Yeh et al., 2009).

A conventional random walk model works as a graph partitioning method like the spectral graph partitioning model. Given a weighted graph G with vertices (nodes) set V and edges set E , a random walk W starting from a node n_0 will move from a node i to another node j with a probability P_{ij} . This probability is calculated by normalizing the edge weights of node i . ($P_{ij} = w_{ij} / \sum_k w_{ik}$ where w_{ij} is weight of edge between node i and j , $\sum_k w_{ik}$ is the sum of all edges connected to node i .) Without any terminating condition, if we repeat the random walk process a

sufficiently large number of times, the random walk W will eventually become stationary and be trapped in densely connected sub-graphs³³. Therefore, a stationary transition matrix can be derived in conventional random walk models to identify the most probable final nodes of a random walk³⁴. This traditional stationary transition probability based random walk was used in (Ramage et al., 2009; Hughes and Ramage, 2007; Yeh et al., 2009).

7.2 Random Walk Model for Event Coreference Resolution

In this section, we will present how to apply the random walk model to event coreference resolution. The conventional stationary transition probability can be applied directly to coreference resolution as a graph partitioning algorithm. In Section 7.2.1, we will illustrate how to apply the conventional approach. In Section 7.2.2, we will introduce the sampling way to apply random walk model to coreference resolution.

7.2.1 Random Walks Through Stationary Transition Probability

Similar to the process in Section 6.1.1, we can form the mention similarity graph using the mention pair classifiers' confidence outputs. Given the mention similarity graph G with set of vertices $V = \{M_1, M_2, \dots, M_n\}$ where M_i s are mentions in G and set of edges E with a set of edge weights W where $e_{i,j} \in E$ denotes an edge in G and $w_{ij} \in W$ is the confidence from the mention pair resolver linked M_j to M_i . Let H be

³³ In a weighted graph, "densely connected" is subjected to the normalization by edge weights.

Although for certain graphs such as non-Ergodic graphs, the transition probability may not converge to a stationary distribution. For event coreference mention graph in this thesis, although we do not have a theoretical proof for the Ergodicity, we have conducted empirical investigation on this issue. Throughout our experiments, all the event coreference mention graphs converge to a stationary transition probability distribution.

³⁴ More elaboration on random walk model is given in Appendix A10.

the transition matrix of G . Let H_{ij} be the (i, j) -element of H which is the transition probability from M_i to M_j . H_{ij} is calculated as $H_{ij} = \frac{w_{ij}}{\sum_{M_k \in N(M_i)} w_{ik}}$ where $N(M_i)$ is a function that returns the set of neighbors of M_i . By a series of matrix multiplication of H , we can derive the stationary transition probability matrix T . By setting a threshold p , we can obtain a connectivity matrix C where $C_{ij} = 1$ if $T_{ij} \geq p$ and $C_{ij} = 0$ otherwise. $C_{ij} = 1$ indicates that the mention M_i and M_j are in the same cluster. According to the connectivity matrix C , we can partition the mentions in mention graph G into a set of clusters. Each cluster corresponds to an event. The derivation of T involves a rigorous proof and lengthy explanation for which other researchers are writing an entire book (Aldous and Fill, 2001) and papers (Lovász, 1993) on this topic. Instead of repeating the proof, we would like to direct the reader to (Aldous and Fill, 2001) and (Lovász, 1993) for mathematical details.

However, the conventional random walk model lacks consideration for certain special characteristics of event coreference. First of all, for the event coreference task, we are more interested in all the nodes visited by the random walks instead of the final node of the random walks. All the nodes visited by a random walk are considered as mentions of the same entity. Secondly, the conventional random walk model assumes an infinite length of the walk whereas event coreference chains are in general very short. In addition, the conventional model fails to incorporate the linguistic constraints and preferences (as in Section 7.3) at all. Therefore, we have made three meaningful modifications to the conventional random walk model to make it more suitable to the event coreference task.

7.2.2 Random Walks Through Sampling Method

The list of nodes visited by a walk is “random” depending on the choice of the neighboring nodes. Different random walks may be produced from the same starting nodes. Instead of deriving the stationary transition probability, we conducted a reasonably large number of random walks from the same starting node. A random walk begins from a starting node n_0 . A walk at a currently visited node n_i will choose its next-hop node n_{i+1} randomly from its neighboring nodes set N_i . The probability that the walk will choose n_{i+1} to move to is the normalized weight among all the edge weights from n_i to every member in N_i . The walk will continue till it fulfills one of the terminating criteria presented in the next sub-section. When a walk is finished, we consider all nodes traversed along the path of the walk n_0, n_1, \dots, n_j as mentions in an event.

After obtaining the set of random walks, a mention is included in the event chain if it appears more than a threshold t number of times in the observed random walks³⁵. This sampling way of random walk accounts for three unique characteristics of the event coreference phenomena.

Firstly, event coreference chains are generally short in length but the traditional stationary transition probability matrix describes an eventually stationary situation of the random walk which is equivalent to a walk with unlimited length. Thus in this sampling random walk, we can conveniently limit the length of the walks by a terminating criterion on the number of nodes in the current walk.

Secondly, the mentions in the original text appear in a natural order. In literature, we usually assume the latter mention refers back to a prior mention but not

³⁵ In this particular work, we empirically select the number of sampled walks to be 100, and the mention inclusion threshold to be 70 occurrences. More details can be found in Appendix B1.

the opposite direction. In such a sampling random walk, we can conveniently model this intuition as a terminating probability that varies with the mention's position in the article. In addition, this modification may reduce the computation complexity as shorter walks are produced³⁶.

Last but most importantly, we have introduced a number of linguistic constraints and preferences in a later section (Section 7.3) which helps to boost the random walk model performance. These constraints and preferences depend on the previously visited nodes by the current walk. Such self-interacting walks are intractable using the traditional stationary transition probability approach. Therefore the modification to the conventional random walk is necessary and also benefits from the incorporated linguistic constraints and preferences.

There are a number of works in NLP research utilizing random walk models. The majority of them fall in the closed form solution category. The most related work is (Hassan and Radev, 2010) which followed a sampling approach as we do in this chapter. However, (Hassan and Radev, 2010) employed a different sampling based random walk from ours; though their major focus was still on the final node of the random walk. The stationary transition based approach is theoretically capable of handling their problem. Due to the intractable size of their graph, they adopted the sampling method by sampling the final node of random walks. Our focus is on the set of visited nodes by the walk with additional terminating criteria. Thus we are sampling all the nodes visited by the random walk under specific necessary conditions. The conventional stationary transition based approach fails to solve our problem as it can at most handle the limited length of a walk but not the linguistic

³⁶ The reduction in complexity here does not refer to change in complexity class. The overall complexity class will remain but there will be a complexity reduction by a linear term.

constraints and preferences using self-interacting walks. Thus we propose this necessary modification to the random walk model.

7.2.3 Incorporating Corpus Knowledge through Terminating Criteria and Terminating Probability

As random walks traverse in the mention graph, we introduce three terminating criteria.

Firstly, a random walk is terminated when it comes back to a node visited by it. This early termination aims to prevent random walks from oscillating among a few densely connected nodes.

Secondly, a random walk is limited to eight steps which is the length of the longest event chain observed in the OntoNotes 4.0 Corpus.

Thirdly, each node is associated with a terminating probability estimated³⁷ from the training corpus using the type of the mentions³⁸, the position in the text and the length of current walk. This is based on the observation that most of the event chains are generally short (2.72 mentions) in length. Therefore, random walks for event coreference should prefer to terminate early instead of traversing till they reach a stationary situation. Similarly, if we reach a node corresponding to a mention very near to the beginning of a text, it should prefer to stop instead of moving on. This is because if a mention appearing later in the text refers back to mentions in prior text, then mentions appearing earlier in the text are generally discourse-new mentions.

³⁷ The terminating probability is estimated from the training corpus using a linear regression on three factors mentioned above.

³⁸ Type of mentions includes verb, pronoun, definiteNP, ProperNP, indefiniteNP and ComplexNP.

7.2.4 Incorporating Mention Knowledge through Starting Points Selection

Since our focus is on event coreference, only mentions representing events are selected as starting nodes for our random walk. This set of starting nodes consists of all the verbal mentions and event noun phrases. The verbs are ordered with higher precedence than event noun phrases. The order of the verbs will be sorted according to their topic-related priority in the event verb key word list produced by the LDA topic models (in Section 4.3). Similarly, the order of the event noun phrases are sorted by their topic-related priority in event noun phrases key word list.

7.3 Incorporating Linguistics Knowledge in a Dynamic Way

The random walk process can be conveniently equipped with linguistic constraints and preferences to guide the walking process when selecting the next-hop node to move to. Although the spectral graph partitioning method is able to incorporate linguistic knowledge as well, the random walk manages to incorporate it in a dynamic way. Spectral graph partitioning can only apply linguistic constraints before clustering. Random walks can check such constraints during execution, when the sampled walks are traversing the mention graph. While spectral graph partitioning can only check the consistency between two-mentions (negative-edge-pruning can check consistency among three nodes but it is still a limited number of nodes.), random walk can check the chain consistency conditioned on all the traversed nodes in the current walk. Figure 7.2 demonstrates the differences between spectral graph partitioning and the random walk model.

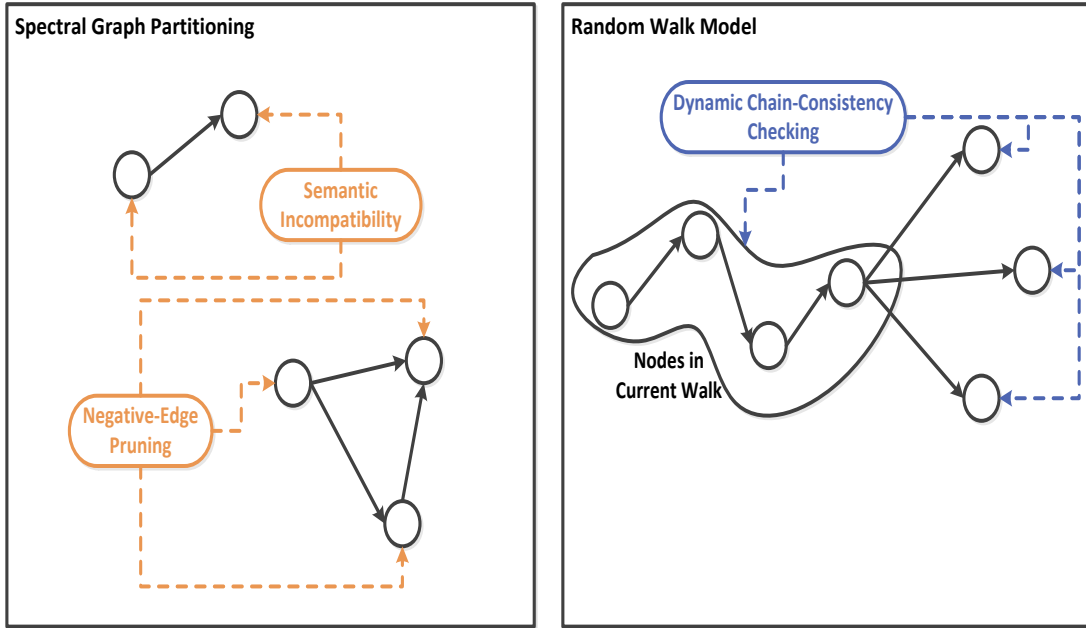


Figure 7.2: Spectral Graph Partitioning v.s. Random Walk Model

7.3.1 Dynamic Chain Consistency Enforcement in Random Walk

We have crafted a set of twenty-eight pruning rules³⁹ to eliminate next-hop nodes which may cause inconsistency in an event chain. A neighboring node of the current node will be disqualified for the random walk if it triggers one of the pruning rules. To enforce chain consistency, a next-hop node is tested against all the nodes currently in the walk. Such pruning rules are crafted based on linguistic intuition and error analysis on the training corpus. The pruning rules include five types:

1) *Conflicting Event Semantics:*

This rule is fired if a next-hop node and one of the current walk nodes belong to conflicting event semantics in WordNet⁴⁰. For instance, this rule is to eliminate improper linking to “*announcement*” (belongs to “communication”) given current node is “*invasion*” (belongs to “military_operation”).

³⁹ The full set of pruning rules can be found in Appendix A7.

⁴⁰ The event semantic is obtained from the WordNet Hypernymy relations. More details can be found in Appendix A6.

2) Conflicting Event Arguments:

This rule is to filter out cases as “conflicts in Middle East” vs. “conflicts in Afghanistan”. These two mentions shall not be linked as they have conflicting arguments. The role of the event argument is decided by heuristics⁴¹. Only location and date-time arguments are checked.

3) Conflicting Number Agreement:

This rule is to prune improper links between singular and plural “conflicts” and “suicide”---“attack”. The next-hop node has to be number compatible with all the nodes in the current walk.

4) Conflicting Text-Span:

This rule will prune cases when a next-hop node is spanned by or overlapped with one of the nodes in the current walk. Since we are generating mentions from the parse trees with padding rules, there are mentions that have overlapping text-span. This rule is to remove those overlapping mentions.

5) Conflicting Governing-Node:

This rule will eliminate cases where two mentions are governed by the same VP node in the parse tree. This follows the intuition that if two mentions are governed by the same VP, they are most likely to be two different roles of the same event which are very unlikely to be coreferent.

Some of the above-mentioned constraints are utilized in the mention-pair SVM models introduced previously to calculate the similarity between mentions. However,

⁴¹ The event arguments are identified using pre-modifiers and propositional phrase attachment. They are then spotted as date/time or location by surface patterns. More details can be found in Appendix A4.

SVM models only consider linguistic constraints between two mentions as soft features, which means that two mentions may still obtain a high score even if they violate one of the hard linguistic constraints. In contrast, by using this linguistically constrained guided random walk we can enforce the nodes' consistency at the set level.

7.3.2 Mention Preference Knowledge through Dynamic Probability Updating

In addition to the pruning rules, we also derive a list of preference rules to favor the next-hop nodes that satisfy linguistic preferences⁴². To maintain the “randomness” of the walk, instead of picking the node preferred by rules, we increase⁴³ the probability for selecting that node for the walk. Similar to the treatment of the pruning rules, the preference rules are tested against all the nodes in the current walk. These preference rules are derived from both linguistic knowledge and error analysis on the training corpus. The set of preference rules includes three types:

1) Shared/Compatible Event Semantics:

A neighboring node is preferred if it has the same/compatible event semantics from WordNet as the nodes of the current walk. This list of compatible event semantics is carefully chosen from the WordNet hypernymy relations to capture the small semantic differences.

⁴² There are total 19 such preference rules excluding the “fixed-pairing” category. The full list of preference rules can be found in Appendix A8.

⁴³ In this work we empirically choose to double the edge weights to increase its probability and normalized against other next-hop nodes to maintain the basic axioms of probability. More details on empirical choices can be found in Appendix B1.

2) Shared/Compatible Event Arguments:

A next-hop node is preferred if it shares a same/compatible argument as one of the nodes in the current walk. The resolved object coreference information is used when we decide whether two arguments of different nodes are compatible or not. The headword of an event is also counted as one argument of the event. In the actual implementation, when we manipulate the next-hop probability using shared/compatible argument preferences, the original edge probability is increased according to the number of arguments matched⁴⁴.

3) Fixed Pairing of Head-Words:

A list of fixed pairing of head-words is collected from the training corpus. These rules are used to prefer links like “say”---“statement”. These pairings are derived from the error analysis on the training corpus. There are in total eighteen such pairs⁴⁵.

7.4 Dynamic Chains Pruning using Object Mentions

In our proposed model above, only event and ambiguous noun phrases are used for final event chain formation. However, the ambiguous NPs also consist of object NPs which may introduce noise into the event coreference chains. Therefore we propose to incorporate a portion of object graph nodes which helps to rule out the object NPs. Object mention graphs in general are much larger and denser than the event mention graphs. Including the whole object graph will increase the computation complexity unnecessarily. Therefore, we only expand the mention graph by adding those object NP nodes having links with any of the ambiguous NP nodes. Those object nodes that

⁴⁴ A headword match will give a 50% increase in edge probability while compatible arguments will give a 25% edge probability increase. The total increase is capped at 100% (i.e. doubled). More details on empirical choices can be found in Appendix B1.

⁴⁵ Full list of Fixed Pairing words can be found in Appendix A5.

are only linked to other object nodes will not be added to keep a smaller graph for random walk.

After adding in the object nodes, we impose one more terminating criterion into the random walk model. Any random walks visiting an object node will be immediately terminated and discarded. Since we use a sampling approach for random walks, in order to maintain the size of samples, a new walk from the same starting node is conducted. An illustration of this condition is shown in Figure 7.3.

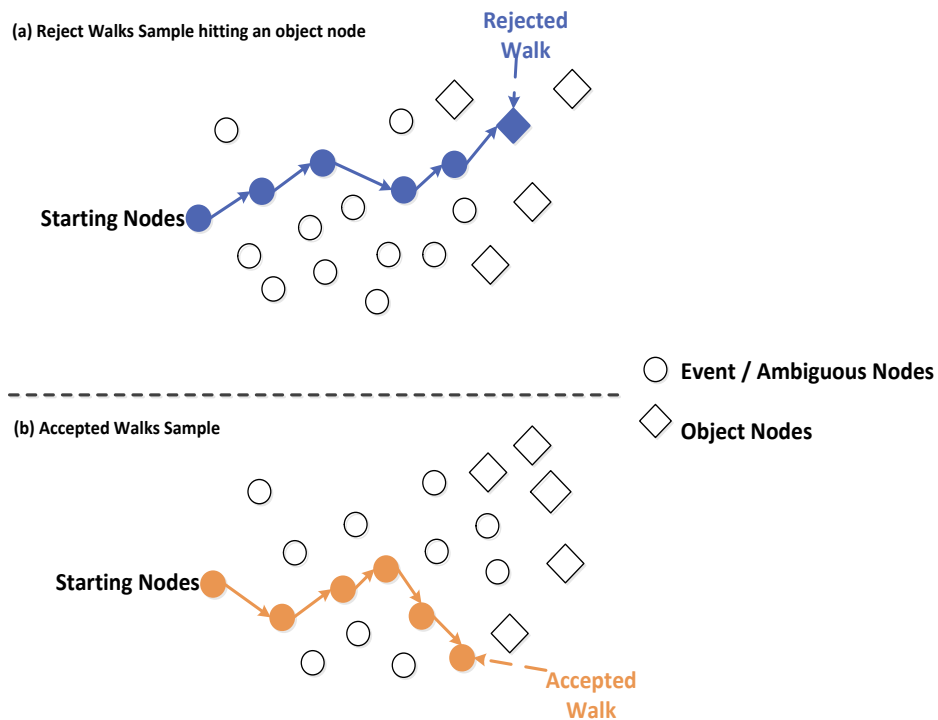


Figure 7.3: Object Nodes Pruning Situation

7.5 Chapter Summary

In this chapter, we have presented the second chain formation technique, random walk graph partitioning. Random walk model can capture the event coreference characteristics through various terminating criteria and probability. In addition, the random walk model is also capable to incorporate linguistics constraints and preferences. Comparing to the spectral graph partitioning approach in Chapter 6,

random walk graph partitioning is capable of enforcing the chain consistency dynamically. Furthermore, the chain consistency is enforced by comparing mention-to-chain consistency while the previous best-cut model and spectral graph partitioning model can only check consistency by mention-to-mention comparison. Last but not least, the random walk model utilizes the object mention graph information to prune the event chains.

In the next chapter (Chapter 8), we will present the experimental results to verify the effectiveness of all the previous proposed techniques.

Chapter 8: Experimental Results and Discussion

In this chapter, we will present our experiment results with discussion. Before showing experimental results, we will briefly discuss the corpus investigation, performance metrics and experimental settings. After that, the experimental results are presented according to the steps in our resolution framework, namely event mention extraction, mention-pair resolution and coreference chain formation. Each step will consist of a section discussing the techniques we proposed.

8.1 Introduction to OntoNotes 4.0 Corpus

The corpus we used is OntoNotes 4.0 which contains 600k words of English newswire, 200k word of English broadcast news, 200k words of English broadcast conversation and 300k words of English web text. OntoNotes4.0 provides gold annotation for parsing, named entity, and coreference.

8.1.1 Event Coreference Annotation

In this section, we will show how to identify the event coreference annotations from OntoNotes 4.0 Corpus. The original OntoNotes 4.0 Corpus is only annotated with coreference information. The annotation does not distinguish between event coreferences and object coreferences. We have conducted a semi-automated process to identify the event coreference annotations. The following four steps are essential steps to identify the event coreference annotations in OntoNotes 4.0 Corpus.

1. Include all the coreference chains that have at least one verb mention.
(OntoNotes 4.0 provides gold POS annotation, thus the verb mention detection is reliable.)

2. Exclude all the coreference chains that have at least one Named Entity (except “Event” category) annotation. (OntoNotes 4.0 provides gold NE annotation)
3. Exclude all the coreference chains that have at least one mention of the personal pronouns (including I, me, my, myself, you, your, yourself, yourselves, he, him, his, himself, she, her, herself, we, us, our, ourselves)
4. Exclude the coreference chains that have at least one mention belong to major continent / country / state / province / city list.
5. Manually exclude any other object coreference annotations.
6. The remaining annotations are considered as event coreference annotations.

8.1.2 Corpus Statistics

After the event coreference annotations are extracted from the original OntoNotes 4.0 corpus, we have gathered basic corpus statistics of event coreference distribution. The distribution of event coreference is tabulated below in Table 8.1.

	# of Articles	# of Chains	# of Mentions
Event	1414	3687	10012
Object	2068	20063	74956
Total	2078	23750	84968

Table 8.1: Corpus Distribution

The distribution of event chains is quite sparse. On average, an article contains only 2.6 event chains compared with 9.7 object chains. Furthermore, event chains are generally shorter than object chains. Each event chain contains 2.72 mentions comparing to 3.74 mentions in each object chain.

8.2 Performance Metrics

Different performance metrics are employed at different levels in our resolution framework. In this section, we will explain each metric in detail.

8.2.1 Event Mention Extraction Metric

For event mention extraction, we are only focusing on the coverage of such method. The coverage measures the percentage of event mentions in gold annotation that have been correctly extracted. Another comparison measure is the total number of mentions extracted. For a fixed level of coverage, the lower the total number of mentions would imply the better is the extraction result.

8.2.2 Mention-Pair Resolution Metric

At the mention-pair level, we use two different performance metrics to measure the resolution results.

The first one is precision/recall/F-score commonly used in many conventional coreference resolution systems. We refer to this measurement as the “best-candidate evaluation”. The best-candidate evaluation follows the traditional mention pair evaluation. It first groups mention-pair predictions by anaphor. Then an anaphor is correctly resolved as long as the candidate-anaphor pair with the highest resolver’s score is the true antecedent-anaphor pair. The other candidates’ resolution outputs are not counted at all.

As a possible counterpart, we propose the “coreferent-link evaluation” which counts each candidate-anaphor pair resolution separately. Intuitively, the best-candidate evaluation measures how well a resolver can rank the candidates while the coreferent link evaluation measures how well a resolver identifies coreferent pairs. Table 8.2 shows the difference between the two evaluations.

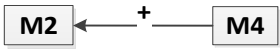
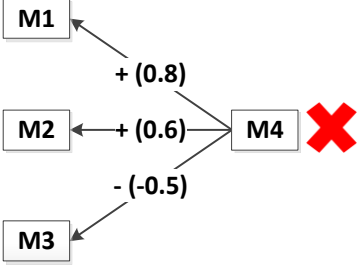
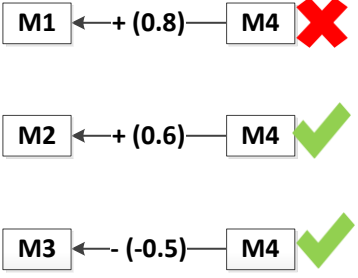
Gold Standard	
	
Best-Candidate Evaluation	Coreferent-Link Evaluation
	
Score: 0% (0/1)	Score: 66.7% (2/3)

Table 8.2: Two Mention-Pair Evaluations

In this example, the correct mention pair is “M2---M4”. The conventional best candidate evaluation will score 0%, as the highest ranked candidate forms an incorrect pair. However, the coreferent-link measure will give a 66.7% score as two of the pairs are classified correctly.

8.2.3 Event Chain Resolution Metric

At the coreference chain level, we evaluate over the commonly used B-Cubed F-Score (Bagga and Baldwin, 1998), which is a measure of the overlap of predicted clusters and true clusters. It is computed as the harmonic mean of precision(P),

$$P = \frac{1}{N} \sum_{d \in D} \left(\sum_{m \in d} \left(\frac{c_m}{p_m} \right) \right)$$

and recall(R),

$$R = \frac{1}{N} \sum_{d \in D} \left(\sum_{m \in d} \left(\frac{c_m}{t_m} \right) \right)$$

and F-score(F),

$$F = \frac{2 \cdot P \cdot R}{(P + R)}$$

where c_m is the number of mentions appearing both in m 's predicted cluster and in m 's true cluster, p_m is the size of the predicted cluster containing m , and t_m is the size of m 's true cluster. Finally, d represents a document from the set D , and N is the total number of mentions in D .

B-Cubed F-Score has the advantage of being able to measure the impact of singleton entities, and of giving more weight to the splitting or merging of larger entities. It also gives equal weight to all types of entities and mentions. For these reasons, we report our results using B-Cubed F-Score.

8.3 Experiment Settings

For each experiment conducted, we use the following data split. 600 articles are reserved to train the object NP-Pronoun and object NP-NP resolvers. Among the remaining 1478 articles, we randomly selected 1182 (80%) for training the five event resolvers while the other 296 articles are used for testing.

In order to separate the propagated errors from preprocessing procedures such as parsing and named entity recognition, we used OntoNotes 4.0 gold annotation for Parsing and Named Entities only. Coreferent mentions are generated by our system instead of using the gold mentions.

We perform the paired Student's t-test at 5% level of significance to verify the significance in performance differences⁴⁶. To make the paired t-test statistics sufficient, we conduct the experiments twenty times through a random sampling method to gather the performance data⁴⁷.

⁴⁶ The details on Student's t-Test can be found in Appendix B3.

⁴⁷ The details on the 20 runs through random sampling can be found in Appendix B2 and B3.

8.4 Experimental Results

The experimental results section is divided into four different sections. The first section will present the performance on event mention extraction. After that, mention-pair resolution results are presented. The last two sections will present the results of the two proposed chain formation methods separately. A full set of 20 experiment results is tabulated in Appendix C1. A full list of t-test p-values is tabulated in Appendix C2.

8.4.1 Event Mention Extraction Performances

The first set of experimental results is the event mention extraction coverage and mention number. We employ a natural baseline which simply includes all the mentions (verbs, pronouns and noun phrases) as event mentions. After that, we gradually introduce the mention extraction technique using heuristics, WordNet and topic-based event detection.

Using Heuristics and WordNet in (Section 4.1 & 4.2)

Table 8.3 shows the natural mention extraction baseline in the first row. The extraction performance using heuristics and WordNet knowledge is shown in the second row.

Event Mention Extraction System	Coverage	Extracted Mention Number
<i>Natural Mention Extraction</i>	100%	243056
<i>+ Heuristics & WordNet</i>	97.6%	96720

Table 8.3: Event Mention Extraction using Heuristics and WordNet

With a very small drop in coverage (2.4%), we managed to reduce the total number of extracted mention by 60.2%. Only two fifths of the original extracted mentions are

retained for resolution. Intuitively, for the same level of coverage, the less number of mentions is the better situation for the later mention-pair resolution.

Using Topic-Related Keyword List in (Section 4.3)

Although using heuristics and WordNet knowledge can significantly reduce the extracted mention number, over ninety thousand mentions are still too much for the latter resolution task. Therefore, we propose a better method of using topic-based keywords to further reduce the number of extracted mention. This set of experimental results is tabulated in Table 8.4.

Event Mention Extraction System	Coverage	Extracted Mention Number
<i>Natural Mention Extraction</i>	100%	243056
+ <i>Heuristics & WordNet</i>	97.6%	96720
+ <i>Combined Topic Model</i>	94.2%	52549
+ <i>Separated Topic Model</i>	95.4%	57902

Table 8.4: Event Mention Extraction using Topic-related Keywords

As the results suggest, both the combined and separated topic models manage to further reduce the total number of extracted mentions. They further reduce the number of extracted mentions by 40.1% ~ 45.7%. Compared to the natural mention extraction method, the topic models can reduce the total number of mentions by 76.2% ~ 78.4%. Both models show the usefulness in mention extraction. The combined topic model reduces more but covers less event mentions. The separated model performs in the opposite way. However, we need to decide a better model for further usage. Thus we propose to test the effectiveness on the actual mention-pair resolvers.

We use the Verb-NP resolver’s performance as a representative to demonstrate the effectiveness. Similar observations are obtained for other mention-pair resolvers

as well. We investigate the empirical differences between the combined topic model and the separated topic models in Table 8.5.

Verb-NP Resolver	Precision	Recall	F-score
<i>Basic Resolver (Using Heuristics & WordNet)</i>	55.3%	66.9%	60.5%
<i>+Combined Topic Model</i>	60.7%	63.8%	62.2%
<i>+Separated Topic Model</i>	64.4%	66.0%	65.2%
Event Chain B³	Precision	Recall	F-score
<i>BL Baseline (Using Heuristics & WordNet)</i>	28.2%	59.1%	38.2%
<i>+Topic Model</i>	31.7%	54.9%	40.2%
<i>SGP Baseline (Using Heuristics & WordNet)</i>	25.4%	68.2%	37.2%
<i>+Topic Model</i>	30.5%	66.7%	41.9%

Table 8.5: Event Detection Effect on Resolution System

The upper half of the table shows the performances of the Verb-NP resolver. Both of the topics modelling settings show significant improvements. As we see, event detection using the combined topic model getting topic nouns and verbs at the same time yields a 5% increase in precision with a 3% trade-off in recall. On the other hand, using the separated topic models that treats topic nouns and verbs separately we get a greater improvement in precision (9%) with literally no trade-off in recall. The reason is that we find verb mentions are much less frequent than the noun phrase mentions. Thus the combined model is overwhelmed by the noun phrase mentions. Verb mentions are merely detected as a result of that. However, the separated topic models managed to avoid this problem.

For the chain level measurement, shown in the lower half of Table 8.5, we use a spectral graph partitioning approach without any enhancements as a baseline (shown as “SGP Baseline”). The separated event detection yields a 4% improvement in B³ F-score. Since the separated topic model event detection shows an empirical advantage

over the combined event detection model, in the rest of this chapter, the event detection module refers to the separated topic model event detection.

8.4.2 Mention-Pair Resolution Performances

After the event mentions are extracted, they are passed to the seven mention-pair resolvers to identify the coreferent pairs. In this section, we will present the experimental results on the mention-pair resolvers. First of all, we will examine the usefulness of our flat feature set through a representative resolver. The newly added features will be tested for effectiveness. Then, we will investigate the improvement by introducing the structural information through the tree kernel. After finalizing the feature set, we will illustrate the improvements from our two novel techniques (utilizing competing classifiers' results and better instance selection strategy).

Selected Flat Feature Analysis in (Section 5.2)

At first, we are investigating the effectiveness of flat features using the Verb-NP resolver. The effectiveness of an individual feature is measured in a leave-one-out manner, that is, the performance loss by removing a particular feature from the feature list. The greater performance drop after removing a feature, the more effective that feature is. Instead of showing all the seven distinct resolvers, we choose the Verb-NP resolver as a representative because Verb-NP resolver has the most special flat features proposed in this thesis. Similar improvements are observed in other mention-pair resolvers. Table 8.6 presents the results of this set of experiments.

In Table 8.6, the first row shows the performance using all the flat features. Each line below is the performance after removing the feature in that line from the resolution system. The observations in Table 8.6 suggest that all the features we have discussed in Chapter 5 contribute a significant part in the resolution system. For most

of the features (except position), the overall system is almost not functioning for the identification of the correct antecedent. The performance drops for most of the features are over 30% in F-score. The conclusion we can draw from these observations is that the flat features are co-dependent in performing the event-anaphoric noun phrase resolution task. Each feature’s individual contribution is hard to separate from the overall performance. All of them are essential parts in the resolution system.

<i>Feature</i>	Precision	Recall	F-score
<i>ALL</i>	43.87%	42.86%	43.35%
<i>-Morph</i>	8.74%	5.84%	6.99%
<i>-Synonym</i>	7.24%	4.63%	5.64%
<i>-Fixed Pair</i>	9.94%	5.43%	7.01%
<i>-Cont_Sim</i>	10.35%	4.63%	6.37%
<i>-Cont_Coref</i>	8.17%	4.43%	5.72%
<i>-Ante_Morph</i>	11.00%	6.64%	8.26%
<i>-Ante_Syn</i>	11.95%	7.04%	8.84%
<i>-Ante_NE</i>	10.36%	7.24%	8.51%
<i>-Gram_Role</i>	11.76%	6.64%	8.45%
<i>-Position</i>	47.47%	32.11%	38.31%

Table 8.6: Flat Feature Effectiveness

We note that the use of the positional features incurs a 5.04% drop in F-score. Although it is comparatively smaller than performance drop of other features, it is still a significant part in the overall performance. Especially, after removing positional features, the recall decreases by 10.75%. Therefore, in the later experiments, all the flat features are used for event-anaphoric noun phrase resolution.

Structured Information Analysis in (Section 5.3)

In the next set of experiments, we aim to investigate the effectiveness of each single knowledge source. Table 8.7 reports the performance of each individual experiment. The Verb-NP resolver is selected as the representative.

	Precision	Recall	F-score
<i>Flat</i>	43.87%	42.86%	43.35%
<i>Min-Exp</i>	33.35%	19.95%	24.82%
<i>Simple-Exp</i>	22.22%	8.45%	12.24%
<i>Full-Exp</i>	33.33%	5.63%	9.63%

Table 8.7: Contribution from Single Knowledge Source

From Table 8.7, the flat feature set yields a baseline system with 43.35% F-score. By using each tree structure alone, we can only achieve a performance of 24.82% F-score using the minimum-expansion tree. These results indicate that the syntactic structural information alone cannot resolve event anaphoric noun phrases.

A composite kernel can be used to combine flat features with syntactic structure feature. The third set of experiments is conducted to verify the performances of various tree structures combined with flat features. The performances are reported in Table 8.8.

	Precision	Recall	F-score
<i>Flat features only</i>	43.87%	42.86%	43.35%
<i>Flat + Minimum-Expansion</i>	65.78%	53.60%	59.01%
<i>Flat + Simple-Expansion</i>	62.85%	49.64%	55.43%
<i>Flat + Full-Expansion</i>	64.56%	50.77%	56.77%

Table 8.8: Different Combinations of Syntactic Structural Knowledge

As Table 8.8 presents, all the three types of structural information improve the overall performance by over 10% in F-score. Obviously, syntactic structural

information is very useful for event anaphoric noun phrase resolution when combined with flat features. Minimum expansion tree performs better than the other two structures. The performance difference in simple expansion and full expansion are statistically insignificant. This result shows that contextual structural information is considered noisy rather than helpful in event anaphoric noun phrase resolution. The minimum structural information covering the anaphor and antecedent is the most helpful as it introduces the least amount of noises. This finding is different from the conclusion in conventional pronoun resolution as reported in (Yang et al., 2006;) where simple expansion tree performs best. We believe this difference is caused by the distance of separation from anaphor to antecedent.

Mention-Pair Baseline Models (BL Baseline) in (Section 5.2 & 5.3)

The next set of experimental results presented is the seven mention-pair resolvers using all presented features without any further improvement methods. The Verb-Verb resolver's performance is particularly low due to lack of training instances where only 66 positive instances are available from the corpus.

The coreference chains formed by the conventional Best-Link method give a 40.2% B^3 F-score. The Best-Link model provides us with a natural baseline model (BL Baseline) for comparison. In theory, the spectral graph partitioning can solve the same problem space as min-cut graph partitioning. Therefore, we can use the chain formation results from spectral graph partitioning to mimic the min-cut method performance. In addition to the BL Baseline, the coreference chains formed using spectral graph partitioning without any proposed improvements yields a B^3 F-score of 41.8% which serves as another baseline (SGP Baseline) for further comparison. The difference between BL Baseline and SGP Baseline is statistically significant. Using a

generic chain formation technique such as SGP gives a 1.6% F-score improvement which is about the same effect as (Nicolae and Nicolae, 2006)’s 0.9% MUC-F-score⁴⁸.

Mention-Pair Score	Precision	Recall	F-Score
Event Resolvers			
Verb-Pronoun	32.3%	68.3%	43.9%
Verb-NP	54.2%	68.5%	60.5%
Verb-Verb	19.8%	81.7%	31.9%
NP-Pronoun	46.6%	70.4%	56.1%
NP-NP	48.8%	60.0%	53.8%
Object Resolvers			
NP-NP	56.4%	66.7%	61.1%
NP-Pronoun	59.7%	82.7%	69.4%
Event Chain B³	Precision	Recall	F-Score
BL Baseline	31.7%	54.9%	40.2%
SGP Baseline	30.5%	66.7%	41.8% ⁴⁹

Table 8.9: Mention-Pair Performance

As the results show, the precision in general is particularly low. Our proposed techniques in mention-pair resolution mainly target to improve the precision. From Table 8.9 onwards, a **bold** number in F-score indicates that the system statistically significantly performs better than the system one line above it.

Utilizing Competing Classifiers’ Results (CC) in (Section 5.4)

Since the object resolvers’ results are in general better than those of the event resolver, we propose to utilize competing object classifiers’ results to improve the

⁴⁸ Readers should take note that the performances are not directly comparable as we used B³ evaluation and OntoNotes4.0 Corpus while (Nicolae and Nicolae, 2006) used MUC-Score and ACE-Phase 2 Corpus. The same level of performance is not rigorous conclusion.

⁴⁹ The difference between BL and SGP is statistically significant with p-value of 6.7595E-09. More t-test’s p-values can be found in Appendix C2.

event resolvers’ performance. The experiment results are tabulated below in Table 8.10.

Mention-Pair	Precision	Recall	F-Score
Event Verb-Pronoun Resolver			
w/o object info	32.3%	68.3%	43.9%
with object info	45.0%	64.7%	53.0%
Event NP-Pronoun Resolver			
w/o object info	46.6%	70.4%	56.1%
with object info	57.8%	69.1%	62.9%
Event Chain B³	Precision	Recall	F-Score
BL Baseline	31.7%	54.9%	40.2%
BL Baseline + CC	38.6%	53.0%	44.7% ⁵⁰
SGP Baseline	30.5%	66.7%	41.8%
SGP Baseline + CC	36.5%	65.7%	46.9% ⁵¹

Table 8.10: Performance using competing classifiers’ results

By incorporating the object coreference information, we improve the event coreference resolution significantly, by more than 9% F-score for the Verb-Pronoun resolver and about 7% F-score for the event NP-Pronoun resolver. Object coreference information improves pronoun resolution more than NP resolution. This is mainly because pronouns contain much less information than NPs. Such additional information greatly helps in preventing object pronouns from being mistakenly resolved by the event resolvers. Although object coreference is incorporated at the mention-pair level, we also measure its contribution to B³ score at the chain level. It improves the BL B³ F-score by 4.5%. At the same time, it improves the SGP B³ F-

⁵⁰ The difference between BL and BL+CC is statistically significant with p-value of 3.35848E-13. More t-test’s p-values can be found in Appendix C2.

⁵¹ The difference between SGP and SGP+CC is statistically significant with p-value of 5.8811E-13. More t-test’s p-values can be found in Appendix C2.

score from 41.8% to 46.9% which is a 5.1% improvement. This observation also shows the importance of the collective decision of competing classifiers.

Better Instance Selection Strategy (BIS) in (Section 5.5)

The second mention-pair level technique we proposed is a better training instance selection strategy. Table 8.11 shows improvement using the better instance selection strategy. At mention-pair level, we take the event NP-Pronoun resolver for demonstration. Similar behaviors are observed in other mention-pair models as well. In order to demonstrate the power of a better instance selection scheme, we evaluate the mention-pair results in two different ways, the best-candidate evaluation and the coreferent-link evaluation.

An interesting phenomenon is the performance evaluation using the best candidate actually drops 4.3% in F-measure when employing the revised instance selection scheme. However when we look at the coreferent link results, the revised instance selection scheme improves the performance by 2.8% F-measure. As a result, our revised instance selection scheme trains better classifiers with higher coreferent link prediction results. Since this coreferent link information is further used in the final chain formation step, our revised scheme contributes an improvement on the final event chain formation by 2.1% B³ F-Score for SGP model. As expected, the performance of BL model drops.

This observation shows that the traditional mention-pair model should be revised to maximize the coreferent link performance instead of the traditional best-candidate performance. This is because the coreferent link performance is more influential to the final chain formation process using graph partitioning approach.

Mention-Pair Score	Precision	Recall	F-Score
<i>Event NP-Pronoun using Best Candidate Evaluation</i>			
Basic Resolver +CC	57.8%	69.1%	62.9%
Basic Resolver +CC +NIS	52.0%	67.1%	58.6%
<i>Event NP-Pronoun using Coreferent Link Evaluation</i>			
Basic Resolver +CC	39.9%	64.0%	49.2%
Basic Resolver +CC +NIS	43.3%	65.4%	52.1%
Event Chain B³	Precision	Recall	F-Score
BL Baseline +CC	38.6%	53.0%	44.7%
BL Baseline + CC +BIS	35.3%	55.8%	43.2%
SGP Baseline +CC	36.5%	65.7%	46.9%
SGP Baseline +CC +BIS	39.3%	65.2%	49.0% ⁵²

Table 8.11: Performance using Better Instance Selection

After identifying the coreferent mention-pairs, we now move on to present the experimental results for the chain formation step. We illustrate the performance of the spectral graph partitioning technique first.

8.4.3 Event Chain Formation Performances using Spectral Graph Partitioning

In this subsection, we demonstrate the effectiveness of the four techniques we proposed in Chapter 6. Since the proposed techniques for resolve mention-pair are shown to be effective, we apply these techniques in the rest of this section. The performance analysis starts with the incorporation of pronoun coreference information.

Incorporating Pronoun Coreference Information (PCI) in (Section 6.1.2)

The first chain formation improvement we proposed is the spectral partitioning with pronoun information. The performance improvement is demonstrated in Table 8.12.

⁵² The difference between SGP+CC and SGP+CC+BIS is statistically significant with p-value of 2.172E-11. More t-test's p-values can be found in Appendix C2.

B³ Performance	Precision	Recall	F-score
BL+CC	38.6%	53.0%	44.7%
SGP+CC+BIS	39.3%	65.2%	49.0%
SGP+CC+BIS+PCI	40.4%	66.1%	50.1% ⁵³

Table 8.12: Performance using Pronoun Coreference Information

By incorporating the coreferent pronoun information, the performance is improved by 1.1% in F-measure. Although this improvement is not significant at the 5% level of significance, this incorporation is necessary for the other three techniques (Pruning of the Inappropriate Edges (PIE), Seed Clusters (SC) and Ordering of Decomposed Points (ODP)) to function properly. Therefore, we still incorporate the pronoun coreference information into our resolution system.

Pruning of Inappropriate Edges (PIE) in (Section 6.2)

The second set of experiments results we presented in Table 8.13 is the performance enhancement after applying the technique of pruning of inappropriate edges.

Event Chain B³	Precision	Recall	F-score
SGP+CC+BIS+PCI	40.4%	66.1%	50.1%
SGP+CC+BIS+PCI+PIE	45.5%	62.6%	52.7% ⁵⁴

Table 8.13: Pruning of Inappropriate Edges

As the results show, we achieve a 5% increase in precision, with a 3.5% trade-off in recall. Since the overall system suffers from the low-precision problem in general, such a trade-off gives a B³ F-score increment of 2.6% which is a significant improvement. The baseline system generally tends to output large event chains, as the precision is quite low. Such a large chain normally is a combination of mentions from

⁵³ The difference between SGP+CC+BIS and SGP+CC+BIS+PCI is **NOT** statistically significant with p-value of 0.06366406. More t-test's p-values can be found in Appendix C2.

⁵⁴ The difference between SGP+CC+BIS+PCI and SGP+CC+BIS+PCI+PIE is statistically significant with p-value of 2.9307E-12. More t-test's p-values can be found in Appendix C2.

two more events. By pruning the inappropriate edges, we have significantly reduced the sizes of the output chains. Therefore, the coreference results will be better for other applications.

Forming Seed Clusters (SC) in (Section 6.3)

The next set of performance results we present is the performance using the pre-formed seed clusters. This is shown in Table 8.14.

Event Chain B³	Precision	Recall	F-score
SGP+CC+BIS+PCI+PIE	45.5%	62.6%	52.7%
SGP+CC+BIS+PCI+PIE+SC	46.9%	67.5%	55.3% ⁵⁵

Table 8.14: Forming Seed Clusters

After applying the seed clusters, we get an overall 2.6% increment in B³ F-score. The improvement is mainly from the improvement in recall. As our observation shows, the main increment is from the rejoining of the separated clusters denoting the same event chain. The separation of event chain is mainly caused by the distance between mentions. By joining compatible small clusters into a large one, we managed to recover long event chains. At the same time, the small clusters of closely located points reduce errors of the ambiguous NPs and Pronouns as they could be in the small cluster with a more informative mention and thus they will be correctly resolved together with the informative mention.

Ordering of Decomposed Points (ODP) in (Section 6.4)

The last set of experiments presented is the performance after applying the ordering of decomposed points. They are presented in Table 8.15.

⁵⁵ The difference between SGP+CC+BIS+PCI+PIE and SGP+CC+BIS+PCI+PIE+SC is statistically significant with p-value of 2.6835E-10. More t-test's p-values can be found in Appendix C2.

Event Chain B³	Precision	Recall	F-score
SGP+CC+BIS+PCI+PIE+SC	46.9%	67.5%	55.3%
SGP+CC+BIS+PCI+PIE+SC+ODP	49.6%	67.3%	57.1% ⁵⁶

Table 8.15: Ordering of Decomposed Points

The final results show a significant improvement over the one using a random ordering of the points. This complies with our intuition that the more informative mentions should be considered first during spectrum clustering. The different events will have its own cluster formed instead of mixing together by the ambiguous NPs and pronouns in random ordering. In this subsection, we achieved a 57.1% B³ F-score using the spectral graph partitioning method.

8.4.4 Event Chain Formation Performances using Random Walk

As we discussed in Chapter 7, the random walk model shows its strengths in the chain formation process. In this subsection, we will show the effectiveness of our proposed techniques. First we will show that our modified version of the random walk model is a better choice for event coreference resolution.

Modified Random Walk (MRW) v.s. Conventional Random Walk (CRW) in (Section 7.2)

First of all, we will present the empirical support for our modified version of the random walk model versus the conventional random walk model. Table 8.16 shows the performance differences. The conventional random walk model is denoted by “CRW”. Our proposed modified version of random walk is denoted by “MRW” (short for Modified Random Walk). In this experimental setting, we have applied both

⁵⁶ The difference between SGP+CC+BIS+PCI+PIE+SC and SGP+CC+BIS+PCI+PIE+SC+ODP is statistically significant with p-value of 2.8176E-8. More t-test’s p-values can be found in Appendix C2.

competing classifiers’ results and better instance selection strategy to the random walk model.

B³ Performance	Precision	Recall	F-score
BL+CC	38.6%	53.0%	44.7%
CRW	37.3%	65.2%	47.4% ⁵⁷
MRW	42.2%	68.1%	52.1% ⁵⁸

Table 8.16: Modified v.s. Conventional Random Walk Model

Both of the conventional and our modified random walk model have statistically significant better performance than the BL+CC model. This shows that a chain formation process is beneficial to event coreference resolution. Moreover, our proposed modified random walk model significantly outperforms the conventional random walk model. This shows that the proposed modification to the random walk model is necessary and effective to apply the random walk model to the event coreference resolution task. Therefore, we will use the “MRW” from this point onwards. All the proposed techniques to the chain formation process will be applied to and tested on the “MRW” model collectively.

Incorporating Pronoun Coreference Information (PCI)

The pronoun coreference information is incorporated into the chain formation step as the necessary information for further use. The experimental results are tabulated in Table 8.17.

⁵⁷ The difference between BL+CC and CRW is statistically significant with p-value of 8.2254E-10. More t-test’s p-values can be found in Appendix C2.

⁵⁸ The difference between CRW and MRW is statistically significant with p-value of 9.7817E-14. More t-test’s p-values can be found in Appendix C2.

B³ Performance	Precision	Recall	F-score
MRW	42.2%	68.1%	52.1%
MRW +PCI	43.5%	70.3%	53.7% ⁵⁹

Table 8.17: Incorporate Pronoun Coreference into Random Walk

By incorporating the coreferent pronoun information, the performance is improved by 1.6% in F-measure. Although this improvement is not significant at the 5% level of significance, its incorporation is necessary for the two later techniques (LCP and OGI) to function properly. Therefore, we still incorporate the pronoun coreference information into our resolution system.

Incorporating Linguistic Constraints and Preferences (LCP) in (Section 7.3)

In Table 8.18, we present the performance comparison before and after enforcing the linguistic constraints and incorporating linguistic preferences in the random walk process. The “MRW +PCI” system corresponds to the best model in the previous subsection for comparison. The “MRW +PCI +LCP” system corresponds to the “MRW +PCI” system further extended with the enforcement of linguistic constraints and incorporation of linguistic preferences.

B³ Performance	Precision	Recall	F-Score
MRW +PCI	43.5%	70.3%	53.7%
MRW +PCI +LCP	47.1%	68.9%	56.0% ⁶⁰

Table 8.18: Enforcing Constraints and Preferences

As the results shown, the linguistic constraints and preferences incorporation brings us a 2.3% improvement in B³ F-score. Especially, the precision score is greatly improved. It shows the incorporation of linguistic constraints helps to accurately

⁵⁹ The difference between MRW and MRW+PCI is **NOT** statistically significant with p-value of 0.05945349. More t-test’s p-values can be found in Appendix C2.

⁶⁰ The difference between MRW+PCI and MRW+PCI+LCP is statistically significant with p-value of 1.0073E-07. More t-test’s p-values can be found in Appendix C2.

identify the event coreference chains. In a balanced overview between precision and recall, the improvement is roughly a trade-off between precision and recall as precision improves about 4% while recall decreases a similar amount. The final F-score improves as “MRW +PCI +LCP” provides a more balanced precision and recall than the system without linguistic constraints and preferences.

Pruning with Object Graph Information (OGI) in (Section 7.5)

Table 8.19 below demonstrates the performance improvement by further incorporating the object graph information. The “MRW+PCI+LCP” system corresponds to the system without using the object mention graph. The “MRW+PCI +LCP+OGI” system corresponds to the previous best-performing system with further extension of object mention graph information.

B³ Performance	Precision	Recall	F-Score
MRW +PCI +LCP	47.1%	68.9%	56.0%
MRW +PCI +LCP +OGI	50.7%	67.5%	57.9% ⁶¹

Table 8.19: Performance using Object Graph Information

As the results show, by utilizing the object graph information, we can further enhance the overall resolution performance by 1.9% in B³ F-score. This is mainly from the improvement in precision with only a small drop in recall. It shows by incorporating the object mention graph, we can identify the event coreference chains more precisely.

8.4.5 Comparing Spectral Graph Partitioning versus Random Walk

As the empirical results suggest, the spectral graph partitioning method and the random walk method show comparable performance (SGP 57.1% B³-F v.s MRW 57.9%

⁶¹ The difference between MRW+PCI+LCP and MRW+PCI+LCP+OGI is statistically significant with p-value of 5.41261E-11. More t-test’s p-values can be found in Appendix C2.

B³-F). In this section, we will conduct a deeper analysis on the advantages and disadvantages for both methods.

Mathematically, both of them are efficient and robust graph partitioning methods. The basic versions of both models (without any proposed improving techniques) can solve the same problem space in clustering task. However, when applying to the challenging event coreference resolution task, each of the two methods shows different capabilities to enhance the resolution performance. In Table 8.20, we will compare the different strengths of the two models.

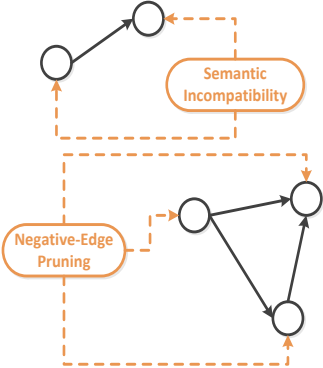
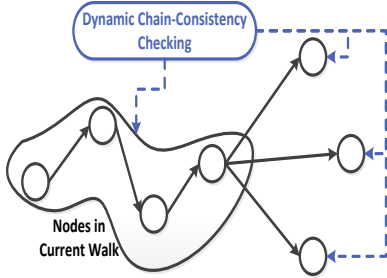
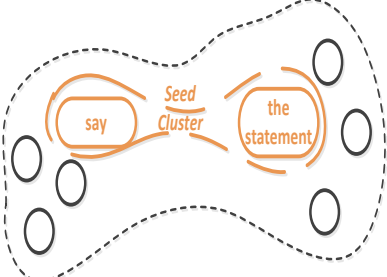
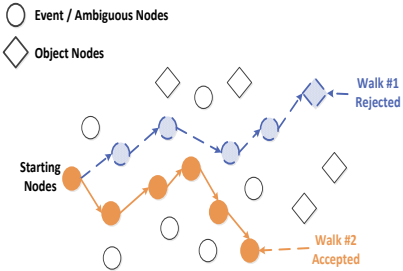
Strengths	Spectral Graph Partitioning	Random Walk
<p>(1) Consistency <i>Helps in enforcing consistency.</i></p>	<p>Static</p>  <p>The diagram shows a graph with nodes and edges. A dashed orange box labeled 'Semantic Incompatibility' highlights a path between two nodes. Another dashed orange box labeled 'Negative-Edge Pruning' highlights a path between two nodes, with one edge being pruned.</p>	<p>Dynamic</p>  <p>The diagram shows a graph with nodes and edges. A dashed blue box labeled 'Dynamic Chain-Consistency Checking' highlights a path between two nodes. A shaded area labeled 'Nodes in Current Walk' highlights a set of nodes.</p>
<p>(2) Seed Clusters <i>Helps in bridging long distance clusters</i></p>	 <p>The diagram shows a graph with nodes and edges. Two nodes, 'say' and 'the statement', are highlighted as 'Seed Clusters' within a dashed orange box.</p>	<p>Not Available</p>
<p>(3) Object Graph <i>Helps in pruning in appropriate nodes</i></p>	<p>Not Available</p>	 <p>The diagram shows a graph with nodes and edges. A legend indicates that circles represent 'Event / Ambiguous Nodes' and diamonds represent 'Object Nodes'. A path of nodes is highlighted, with 'Walk #1 Rejected' and 'Walk #2 Accepted' labeled.</p>

Table 8.20: Comparison between Spectral Graph Partitioning and Random Walk

The first row shows the models' abilities to enforce chain consistency. The MRW model shows a little advantage as it imposes a dynamic version. However, empirical results show that the two methods are empirically equivalent. The pruning of inappropriate edge (PIE) in SGP improves 2.6% B^3 -F while the linguistic constraints and preferences (LCP) in MRW improves 2.3% B^3 -F. The differences are statistically insignificant⁶² which gives the two a draw in the consistency checking capability.

The second row shows that SGP enables the clustering of long distance mentions using seed clusters generated by heuristics. Such capability is very helpful as though event chains are shorter in terms of mentions, they exhibit longer separation distances especially in the Verb-Verb and the Verb-NP cases. In addition, SGP by its own model property can also link two long-distance but densely connected mentions into one cluster even if they are not directly connected. Both these scenarios help in bridging the event mentions beyond our n-sentence window. Empirically, it also shows a significant improvement to B^3 -F. In contrast, the MRW model fails to recover from such losses. Currently, we leave it for the future work to solve. SGP beats MRW in this aspect.

The third row shows the MRW model is capable of pruning inappropriate walk samples using object graph information. This technique helps by keeping the event chain (walks) focused on event mentions, rejecting walks with ambiguous mentions which are close to object mentions. Empirically, it shows a significant improvement in B^3 -F. On the other hand, when including object mentions into SGP, it is hard to decide whether a cluster consisting of all of the event, ambiguous and object mentions

⁶² The difference between SGP+PCI+PIE and MRW+PCI+LCP is **NOT** statistically significant with a p-value of 0.055852875. More t-test's p-values can be found in Appendix C2.

is an event chain or object chain. We leave more in-depth comparisons as another future work in the current thesis. In this round, MRW beats SGP.

In summary, SGP and MRW show a draw of 2:2 in the previous three rounds of comparisons. Empirically, they also produced similar and comparable results. The difference in the final B³ F-scores (SGP 57.1% vs. MRW 57.9%) is statistically insignificant. Based on our finding here, we have to say both the SGP and MRW models are statistically equivalent⁶³, although the MRW model appears to be slightly better in our theoretical analysis and 0.8% B³-F better in our empirical study.

8.4.6 Randomly Selected Error Analysis

The error analysis for coreference resolution is more difficult than other NLP tasks such as Named Entity Recognition (NER). Error analysis for NER can be conducted at the feature level. However, coreference resolution involves a chain formation process. The difficulty of evaluating a clustering algorithm comes from two aspects. First, a clustering decision (e.g. to decide whether to include a mention given the current cluster) is in general hard to judge whether it is a good or bad decision. This is because in most of these cases, such a clustering decision makes certain cases correct while making some other cases wrong. Second, the final clustering result is a collective result from multiple clustering decisions. It is in general hard to identify which one makes the wrong move especially when the chain is formed dynamically such as our random walk model.

In this section, we have randomly selected 170 event chains (consist of 434 event mentions (287 mention pairs⁶⁴): 70 Pronouns, 46 Verbs and 318 Noun Phrases)

⁶³ The difference between SGP+PCI+PIE+ODP and MRW+PCI+LCP+OGI is **NOT** statistically significant with a p-value of 0.142003431. More t-test's p-values can be found in Appendix C2.

⁶⁴ The measurement of mention pairs is subjected to the n-sentence window introduced in Chapter 3.

from one of the twenty runs of experiments. These 170 event chains give a 58.9% F-score (SGP) and 56.2% F-score (RW) around the same level as the overall resolution performance. The 170 chains are manually examined to gain a comprehensive understanding of the causes of error. After manually spotting the possible causes of error, we try to make manual corrections if possible. The corrected results will be presented to the resolution system to verify if any improvements can be gained.

Inaccurate Confidence from Mention-Pair Prediction

The first source of error that attracted our attention is the inaccurate confidence from mention-pair prediction. Both of our chain formation techniques depend on the confidence outputs from the mention-pair resolvers. The inaccurate outputs of the confidence misguide the chain formation techniques in forming incorrect chains. Based on our investigation, we have identified 96 mention pairs (33.4% out of the 287 mention pairs), in which the correct antecedent is not the highest confidence one in all the outputs. We proposed a manual correction as reordering the outputs by giving the highest confidence to the correct antecedent. After correction, the 170 event chain performance is improved by 1.7% F-score (SGP) and 0.4% (RW). However, these 170 chains are too small to give any statistical significance analysis. We can only imply that the inaccurate confidence estimation by the mention pair resolvers is one of the major causes of the errors.

Long Chains Due to Wrong Pronoun Prediction

The second major cause of error in our error study is the inaccurate predictions of pronouns. These inaccurate resolutions give multiple positive antecedent predictions for one pronoun. Each of the positive predictions will produce a positive edge in the mention graph. In the chain formation step, these positive edges may bring incorrect

mentions into the event chain. Out of the 170 event chains, we have identified 59 chains which have inappropriate mentions brought in by edges from pronouns. We proposed manual corrections by removing the incorrect pronoun predictions. The final results are improved by 2.4% F-score in SGP and 3.1% F-score in RW. Due to the limited number of samples, we cannot conduct statistical significance analysis. We can only intuitively infer that the wrong pronoun predictions are one of the major error causes in the current resolution system.

Empirical Decision when Selecting Heuristic Rules

This error is not referring to any particular heuristic rules or preferences we have used. It is a rather common scenario when we make decision whether to include a certain heuristic rule. The fixed pairing set of rules is easy to decide, as it makes a comparatively big improvement. However other rules are much more difficult to decide. The difficulties not only come from choosing a single rule, but also the scenario becomes even harder when considering collective effects from multiple rules. During the actual selection process, empirical impact is an important factor. Within the 170 event chains, we find 4 new rules that can help to improve the final results. However, when we put them into the resolution system, they improved the 170 chains performance but decreased the overall performance on all testing data. At the current stage, our current rule set is the best based on the experiments we have conducted. However, we hope to find a better rule set as the current one still makes a significant number of wrong decisions.

These three sources of error are not the only ones in our error analysis. They are the major ones that have drawn our attentions. Due to the difficulty in conducting error analysis for clustering results, we can only manually process a limited number of error

cases. With the limited number of error cases, we find these three sources of error to be more critical than others. The errors with only one or two occurrences will not be discussed here.

8.5 Chapter Summary

In this chapter, we have shown various sets of experimental results on our proposed techniques at different steps. Most of them have shown statistically significant improvements (except incorporating the pronoun coreference information). Since we have proposed two chain formation techniques, we have presented an in-depth comparison between the spectral graph partitioning and random walk graph partitioning. From the aspect of the knowledge they can incorporate, the two models have their own pros and cons. From the empirical aspect, there are no statistically significant differences between them. Therefore, in this study, we can only conclude that the spectral graph partitioning and the random walk graph partitioning are equivalent.

Last but not least, we have presented an error analysis based on 170 event chains randomly selected from one experiment. We have identified three major sources of the errors. Correction to these errors may lead to further improvement to the resolution performance.

Chapter 9: Conclusion and Future Work

9.1 Conclusion

The purpose of this thesis is to investigate, formulate and propose a feasible and well-performed solution to the challenging event coreference resolution task which lacks attention in the literature. To the best of our knowledge, we are the first to perform a systematic and in-depth study in the literature on event coreference resolution. We adopt the two-step resolution framework and propose a number of novel features, methods and improvements at various stages in the resolution process.

At the mention extraction stage, we have proposed a heuristic plus a WordNet semantic approach for detecting potential event mentions. After that, a separated LDA topic model is introduced into the mention extraction task to detect topic specific high priority event mentions. The empirical results show a huge 78% reduction in the number of mentions extracted which induces a significant 4% B³-F improvement in the event coreference chains resolved.

At the mention-pair resolution stage, we have proposed a number of novel features to bridge the syntactic and semantic gaps discovered in event coreference resolution. Following a divide-and-conquer philosophy, we have created seven distinct mention-pair resolvers to tackle the challenging task. In addition, two effective techniques (utilizing competing classifiers' results and new instance selection strategy) are applied to the mention-pair resolvers. Each of them contributes significant improvements in both the mention-pair and chain formation performance.

Prior to the chain formation stage, we have proposed two very different methods (Spectral Graph Partitioning and Random Walk Model) to form the final coreference chains. Each of the methods has its own specific capabilities dedicated to the event coreference phenomenon. Both of the methods are capable of incorporating pronoun

coreference information which has been intentionally omitted in previous graph partitioning approaches.

For the spectral graph partitioning method, we have proposed three enhancements. Pruning of inappropriate edges enforces chain consistency and linguistic constraints. Selecting seed clusters and ordering of decomposed points provide the spectral graph partitioning model with mention preference knowledge. All these three techniques show significant improvements over the basic spectral graph partitioning model. The spectral graph partitioning approach demonstrates a final score of 57.1% B³-F.

The second chain formation technique, the random walk model, is for the first time adapted and modified for the event coreference resolution task. A sampling approach of the random walk model is adapted to facilitate the self-interacting walks. The sampling random walk model is further modified to utilize the corpus statistical knowledge using the terminating criteria and probability. In addition, two novel techniques are further applied to the random walk model to improve the performance. Linguistic constraints and preferences are utilized in a dynamic way comparing the static use in spectral graph partitioning. Last but not least, the information from object mention graph is used to prune the inappropriate walks from the samples. All the adaptations and improvements show significant increases in chain-level measurements B³ F-score. The random walk model achieves a 57.9% B³-F which is also the highest score reported in this work.

In conclusion, this thesis provides a systematic linguistics and empirical study for the new and challenging event coreference resolution task. It also proposes a computational solution with the state-of-the-art performances. Last but not least, it serves as a foundation for any further research work on event coreference resolution.

9.2 Future Work

With the insights gained from the current work, we would like to explore the following areas to further improve event coreference resolution.

9.2.1 Employing Ensemble Models

The spectral graph partitioning model and random walk model utilize different knowledge and show different resolution capabilities. A natural extension to the current resolution framework is to employ a model ensemble method and make collective decisions from both chain formation models. For further enhancement, each individual mention-pair model can be replaced with a multi-pass ensemble of classification models. The collective decisions are expected to be better than each individual classifier. This future work serves as an engineering improvement to the current resolution system.

9.2.2 Incorporating more Semantic Knowledge

Although we have incorporated event semantic knowledge from the WordNet, it is not a dedicated event semantic dictionary for event coreference resolution. A rather large portion of semantic information is missing in the current work. A carefully designed and dedicated event hierarchy dictionary (to serve as an ontology) could be a possible extension to the current work. Although building a complete event hierarchy on everything is not feasible, building a reasonable sized event hierarchy on a specific domain (e.g. protein-protein interaction) is still a feasible solution. Other potential helpful knowledge includes semantic role labeling results, verb senses and verb frames.

9.2.3 Knowledge Deep Parsing

Knowledge gap happens to be a serious problem in the current work. Some cases require a certain amount of world knowledge to resolve. A knowledge-deep parsing method such as discourse parsing using discourse representation theory could be a valuable knowledge source for closing such knowledge gaps. A world knowledge databank will also benefit the event coreference resolution.

Bibliography

- Aldous, D. and Fill, J. 2001. Reversible Markov Chains and Random Walks on Graphs. Draft-Book. <http://www.stat.berkeley.edu/~aldous/RWG/book.html>.
- Apache Mahout. Spectral Clustering. Online resource at <https://cwiki.apache.org/MAHOUT/spectral-clustering.html>
- Asher, N. 1993. *Reference to Abstract Objects in Discourse*. Kluwer Academic Publisher.
- Bagga, A. and Baldwin, B. 1998. Algorithms for Scoring Coreference Chains. In *Proceedings of the 1st International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference (LREC-1998)*. Granada, Spain. May, 1998.
- Bejan, C. and Harabagiu, S. 2010. Unsupervised Event Coreference Resolution with Rich Linguistic Features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*. Uppsala, Sweden. July 2010.
- Blei, D.M.; Ng, A.Y. and Jordan, M.I.. 2003. Latent Dirichlet allocation. In *Journal of Machine Learning Research*. Year 2003, Vol. 3, Page 993-1022.
- Bunescu, R.C. 2012. Adaptive Clustering for Coreference Resolution with Deterministic Rules and Web-Based Language Models. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics (SEM-12)*. Montréal, Canada. June, 2012.
- Byron, D. 2002. Resolving Pronominal Reference to Abstract Entities, In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*. Philadelphia, USA. July, 2002.
- Cai, J. and Strube, M. 2010. End-to-End Coreference Resolution via Hypergraph Partitioning. In *Proceedings of the 23rd International Conference on Computational Linguistics (CoLing-10)*. Beijing, China. August, 2010.
- Cai, J.; Mjrdicza-Maydt, E. and Strube, M. 2011. Unrestricted Coreference Resolution via Global Hypergraph Partitioning. In *Proceedings of the 15th Conference on Computational Natural Language Learning: Shared Task (CoNLL-11)*. Portland, USA. June, 2011.
- Chen, B.; Su, J. and Tan, C.L. 2010a. A Twin-Candidate Based Approach for Event Pronoun Resolution using Composite Kernel. In *Proceedings of the 23rd International Conference on Computational Linguistics (CoLing-10)*. Beijing, China. August, 2010.
- Chen, B.; Su, J. and Tan, C.L. 2010b. Resolving Noun Phrases to Their Verbal Mentions. In *Proceeding of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP-10)*. Cambridge, USA. October, 2010.

- Chen, B.; Su, J.; Pan, J.S. and Tan, C.L. 2011. A Unified Event Coreference Resolution by Integrating Multiple Resolvers. In *Proceeding of the 5th International Joint Conference on Natural Language Processing (IJCNLP-11)*. Chiang Mai, Thailand. November, 2011.
- Chen, B.; Su, J. and Tan, C.L. (to appear in 2013). A Random Walk Down the Mention Graph for Event Coreference Resolution. In *Journal of Artificial Intelligence Research*.
- Collins, M. 1997. Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*. Madrid, Spain. July, 1997.
- Collins, M. and Duffy, N. 2002. New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures and the Voted Perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*. Philadelphia, USA. July, 2002.
- Davidson, D. 1969. The Individuation of Events. In *N. Rescher et al., eds., Honor of Carl G. Hempel, Dordrecht: Reidel. Reprinted in D. Davidson, ed., Essays on Actions and Events*, 2001. Oxford: Clarendon Press. Page 295-309.
- Davidson, D. 1985. Reply to Quine on Events. In *E. LePore and B. McLaughlin, eds., Essays on Actions and Events: Perspectives on the Philosophy of Donald Davidson*. Oxford: Blackwell. Page 172–176.
- Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Hassan, A. and Radev, D. 2010. Identifying Text Polarity Using Random Walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*. Uppsala, Sweden. July, 2010.
- Hovy, E.; Marcus, M.; Palmer, M.; Ramshaw, L. and Weischedel, R. 2006. OntoNotes: The 90% Solution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL-06)*. New York City, USA. June, 2006.
- Hughes, T. and Ramage, D. 2007. Lexical Semantic Relatedness with Random Graph Walks. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-07)*. Prague, Czech Republic. June, 2007.
- Inoue, N.; Ovchinnikova, E.; Inui, K. and Hobbs, J. 2012. Coreference Resolution with ILP-based Weighted Abduction. In *Proceedings of the 24th International Conference on Computational Linguistics (CoLing-12)*. Mumbai, India. December, 2010.
- Joachims, T. 1999. Making Large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola (ed.). MIT Press.

- Joachims, T. 2001. A Statistical Learning Model of Text Classification with Support Vector Machines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM-SIGIR-01)*. New Orleans, USA. September, 2001.
- Jurafsky, D. and Martin, J.H. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River.
- Lovász, L. 1993. Random Walks on Graphs: A Survey. In *Bolyai Society Mathematical Studies Series 2 – Combinatorics*. Year 1993. Vol. 2. Page 1-46.
- Luo, X.; Ittycheriah, A.; Jing, H.; Kambhatla, N. and Roukos, S. 2004. A Mention-Synchronous Coreference Resolution Algorithm based on the Bell Tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. Barcelona, Spain. July, 2004.
- Luxburg, U. 2006. A Tutorial on Spectral Clustering. In *MPI Technical Reports No. 149*. Tübingen: Max Planck Institute for Biological Cybernetic.
- Martschat, S.; Cai, J.; Broscheit, S.; Mjrdicza-Maydt, E. and Strube, M. 2012. A Multigraph Model for Coreference Resolution. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning: Shared Task (EMNLP-CoNLL-12)*. Jeju Island, Korea. July, 2012.
- Moschitti, A. 2004. A Study on Convolution Kernels for Shallow Semantic Parsing. In *Proceedings of the 42nd Annual Meeting for Association of Computational Linguistics (ACL-04)*. Barcelona, Spain. July, 2004.
- Moschitti, A. 2006. Making Tree Kernels Practical for Natural Language Learning. In *Proceedings of the 11th International Conference on European Association for Computational Linguistics (EACL-06)*. Trento, Italy. April, 2006.
- Müller, C. 2007. Resolving It, This, and That in Unrestricted Multi-Party Dialog. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*. Prague, Czech Republic. June, 2007.
- Ng, A.; Jordan, M. and Weiss, Y. 2002. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems*. Year 2002. Vol. 14. Page 849-856. MIT Press.
- Ng, V. and Cardie, C. 2002a. Combining Sample Selection and Error-Driven Pruning for Machine Learning of Coreference Rules. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP-02)*. Philadelphia, USA. July, 2002.
- Ng, V. and Cardie, C. 2002b. Improving Machine Learning Approaches to Coreference Resolution. In *Proceedings of the 40th Annual Conference for*

- Association of Computational Linguistics (ACL-02)*. Philadelphia, USA. July, 2002.
- Nicolae, C. and Nicolae, G. 2006. BESTCUT: A Graph Algorithm for Coreference Resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP-06)*. Sydney, Australia. July, 2006.
- Pan, S.J.; Tsang, I.W.; Kwok, J.T. and Yang.Q. 2009. Domain Adaptation via Transfer Component Analysis. In *Proceedings of the 21st International Conference on Artificial Intelligence (IJCAI-09)*. Los Angeles, USA. July, 2009.
- Poon, H. and Domingos, P. 2008. Joint Unsupervised Coreference Resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*. Edinburgh, UK. October, 2008.
- Pradhan, S.; Ramshaw, L.; Weischedel, R.; MacBride, J. and Micciulla, L. 2007. Unrestricted Coreference: Identifying Entities and Events in OntoNotes. In *Proceedings of the IEEE 2007 International Conference on Semantic Computing (ICSC-07)*. Irvine, USA. September, 2007.
- Rahman, A. and Ng, V. 2009. Supervised Models for Coreference Resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Suntec City, Singapore. August, 2009.
- Ramage, D.; Rafferty, A. and Manning, C. 2009. Random Walks for Text Semantic Similarity. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing in the Joint Conference of the 47th Annual Meeting of the Association of Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP-09)*. SUNTEC City, Singapore. August, 2009.
- Ribeiro, B. and Towsley, D. 2010. Estimating and Sampling Graphs with Multidimensional Random Walks. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (ACM-SIGCOMM-IMC-10)*. Melbourne, Australia. November, 2010.
- Shamir, R. and Sharan, R. 2001. Algorithmic Approaches to Clustering Gene Expression Data. In *Current Topics in Computational Molecular Biology*. MIT Press. Page 269–300.
- Shi, J. and Malik, J. 2000. Normalized Cuts and Image Segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*. Year 2000. Vol. 22. Number 8. Page 888-905.
- Shier, R. 2004. Statistics: 1.1 Paired T-Tests. Online Resource at <http://mlsc.lboro.ac.uk/resources/statistics/Pairedtttest.pdf>
- Sidner, C.L. 1986. Focusing in the Comprehension of Definite Anaphora. In *Readings in Natural Language Processing*. Morgan Kaufmann Publishers Inc. Page 363-394.

- Soon, W.M.; Ng, H.T. and Lim, D.C.Y. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. In *Computational Linguistics*. Year 2001. Vol. 27. Issue 4. Page 521– 544.
- Stoyanov, V.; Cardie, C.; Gilbert, N.; Riloff, E.; Buttler, D. and Hysom, D. 2010. Coreference Resolution with Reconcile. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*. Uppsala, Sweden. July, 2010.
- Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer.
- Versley, Y.; Ponzetto, S.P.; Poesio, M.; Eidelman, V.; Jern, A.; Smith, J.; Yang, X. and Moschitti, A. 2008. BART: A Modular Toolkit for Coreference Resolution. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC-08)*. Marrakech, Morocco. May, 2008.
- Wang, P.; Murai, F. and Towsley, D. 2012. Sampling Directed Graphs with Random Walks. In *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (IEEE-INFOCOMM-12)*. Orlando, USA. March, 2012.
- Wick, M.; Singh, S. and McCallum, A. 2012. A Discriminative Hierarchical Model for Fast Coreference at Large Scale. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL-12)*. Jeju, Korea. July, 2012.
- Wikipedia. Random Walk. Online Resource at http://en.wikipedia.org/wiki/Random_walk
- Yang, X.; Zhou, G.; Su, J. and Tan, C.L. 2003. Coreference Resolution using Competition Learning Approach. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan. July, 2003.
- Yang, X.; Su, J. and Tan, C.L. 2006. Kernel-Based Pronoun Resolution with Structured Syntactic Knowledge. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia. July, 2006.
- Yang, X.; Su, J. and Tan, C.L. 2008. A Twin-Candidates Model for Learning-Based Coreference Resolution. In *Computational Linguistics*. Year 2008. Vol. 34. Issue 3. Page 327-356.
- Yeh, E.; Ramage, D.; Manning, C.; Agirre, E.; Soroa, A. and Taldea, I. 2009. WikiWalk: Random Walks on Wikipedia for Semantic Relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing in the Joint Conference of the 47th Annual Meeting of the Association of Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP-09)*. SUNTEC City, Singapore. August, 2009.
- Yin, J.; Hu, D.H. and Yang, Q. 2009. Spatio-Temporal Event Detection using Dynamic Conditional Random Fields. In *Proceedings of the 21st International Conference on Artificial Intelligence (IJCAI-09)*. Los Angeles, USA. July, 2009.

Appendix A: Model Design Details

In Appendix Section A, we have ten subsections covering various model design details.

Appendix A1: How to Identify Mention Heads from Parse Tree

In Appendix Section A1, we will show how to extract the head of a phrase. We follow (Collins, 1997)’s method to extract. Readers may refer to the head-word table in <http://people.csail.mit.edu/mcollins/papers/heads> for quick access.

Rules for NP Head Extraction:

Remove ADJPs, QPs, and also NPs which dominate a possessive (tagged POS, e.g. (NP (NP the man 's) telescope) becomes (NP the man 's telescope)). These are recovered as a post-processing stage after parsing.

The following rules are then used to recover the NP head:

- If the last word is tagged POS, return (last-word);
- Else search from right to left for the first child which is an NN, NNP, NNPS, NNS, NX, POS, or JJR
- Else search from left to right for first child which is an NP
- Else search from right to left for the first child which is a \$, ADJP or PRN
- Else search from right to left for the first child which is a CD
- Else search from right to left for the first child which is a JJ, JJS, RB or QP
- Else return the last word.

Instructions for Tree Head Table:

The first column is the non-terminal. The second column indicates where you start when you are looking for a head (left is for head-initial categories, right is for head-

final categories). The rest of the line is a list of non-terminal and pre-terminal categories which represent the head rule.

Tree Head Table:

Label	Direction	Head Rule
ADJP	Right	NNS QP NN \$ ADVP JJ VBN VBG ADJP JJR NP JJS DT FW RBR RBS SBAR RB
ADVP	Left	RB RBR RBS FW ADVP TO CD JJR JJ IN NP JJS NN
CONJP	Left	CC RB IN
FRAG	Left	
INTJ	Right	
LST	Left	LS:
NAC	Right	NN NNS NNP NNPS NP NAC EX \$ CD QP PRP VBG JJ JJS JJR ADJP FW
PP	Left	IN TO VBG VBN RP FW
PRN	Right	
PRT	Left	RP
QP	Right	\$ IN NNS NN JJ RB DT CD NCD QP JJR JJS
RRC	Left	VP NP ADVP ADJP PP
S	Right	TO IN VP S SBAR ADJP UCP NP
SBAR	Right	WHNP WHPP WHADVP WHADJP IN DT S SQ SINV SBAR FRAG
SBARQ	Right	SQ S SINV SBARQ FRAG
SINV	Right	VBZ VBD VBP VB MD VP S SINV ADJP NP
SQ	Right	VBZ VBD VBP VB MD VP SQ
UCP	Left	
VP	Right	TO VBD VBN MD VBZ VB VBG VBP VP ADJP NN NNS NP
WHADJP	Right	CC WRB JJ ADJP
WHADVP	Left	CC WRB
WHNP	Right	WDT WP WP\$ WHADJP WHPP WHNP
WHPP	Left	IN TO FW

Appendix A2: WordNet Hypernym Lists for Event and Object

In Appendix Section A2, we will show the WordNet hypernym lists for events and objects.

Event Hypernym List (21 words): Human_Act; Military_Operation; Happening; Occurrence; Killing; Change_of_State; Attack; Plan_of_Action; Maneuver; Discharge; Acquisition; Aggression; Policy; Care; Death; Procession; Transgression; Ceremony; Change_of_Magnitude; Social Policy; Water_Sport.

Object Hypernym List (27 words): Location; Device; Artifact; Living_Thing; Natural_Object; Administrative_District; Skilled_Worker; Corporate_Executive; Male; Female; Businessperson; Municipality; Food; Calender_Day; Calender_Month; World_Organization; Mammal; Bird; Chemical; Print_Media; Body_Part; Monetary_Unit; Place_of_Business; Person_of_Color; Metric_Uint; Mass_Unit; Building_Complex.

Appendix A3: Common Phrases

In Appendix Section A3, we will show the common phrases we used in addition to the LDA identified key event word list.

Common Phrases: “be”, “decide”, “determine”, “get”, “take”, “make”, “do”, “seem”, “consider”, “state”, “announce”, “speak”, “tell” and their derivational forms.

Appendix A4: Event Argument Extraction and Matching

In Appendix Section A4, we will show how the event arguments are extracted and matched. We only conducted a simple argument extraction. We only extract the time, location and actuator/patient. We use only the attached prepositional phrases (from parse tree) and pre-modifiers (in XXX's format) to identify the arguments. For time argument, we only identify the Named Entity, the names of 12 months, the 4(or 2) digits year, the time with "am/pm" and their combinations. For location argument, we only identify the Named Entity, a list of geo-location names (including the names of 7 continents, continents name with directions such as East Asia, North America, common geo-locations as Far East, Middle East.), a list of country/province/state/city names and a list of acronyms of common country/state/city. For actuator/patient role, we only identify the person category of Name Entities. The time/location argument can be easily caught with regular expressions.

For example, "I study in Singapore." The prepositional phrase "in Singapore" is attached to the verb "study". From country name list, we find "Singapore" belongs to location.

Appendix A5: Fixed Pairings of Words

In Appendix Section A5, we will show the 18 fixed pairing of words.

Fixed Pairs: “say / announce / speak – statement”; “say / tell – words”; “bill / policy – measure”; “trouble – misfortune”; “ceremony – celebrate”, “plan – proposal”; “cut – decrease”; “attack – bombing / blast / explosion”; “administration – rule / reign”; “investigation – study / research”;

Appendix A6: Event Semantic Compatibility/Incompatibility

In Appendix Section A6, we will show the event semantic compatibility / incompatibility. These checking criteria are defined in terms of the surface words and the WordNet hypernyms.

Important Compatibility Pairs (16 pairs): “Attribute – Form”; “Pathological_State – Shock / Collapse”; “Signal – Alarm / Recording”; “Illness – Growth / Collapse / Ague”; “Law - Prohibition”; “Ill_Health – Affliction / Infection”; “Case – Civil_Suit / Class_Action / Criminal_Suit / Countersuit”.

Important Incompatibility Pairs (22 pairs): “Operation – Surgical_Procedure”; “Speech_Act – Concession / Discord / Prayer”; “Group_Action – Defense / Warfare / Manufacture”; “Due_Process – Denial / Judgment”; “Selling – Capitalization”; “Social_Event – Stage_Dance / Movie / Picture / Attraction”; “Transaction – Business / Finance”; “Management – Supervision / Finance / Homemaking”; “Work – Job / Housework / Loose_End”.

Appendix A7: Semantic Incompatibility Pruning Rules

In Appendix Section A7, we will show the entire set of 28 semantic incompatibility pruning rules.

Rule 1: If the two mentions have number disagreement;

Rule 2: If the two mention have overlapping text span;

Rule 3~5: if the two mentions are governed by a common parent VP / PP / NP node;

Rule 6: If the two mentions have non-matched time/location arguments; (Event argument is extracted as in Appendix A4.)

Rule 7~28: The incompatibility pairs in Appendix A6.

Appendix A8: Semantic Compatibility Preference Rules

In Appendix Section A8, we will show entire set of 19 semantic compatibility preference rules.

Rule 1: if the two mentions have exact argument match for time/location;

Rule 2: if the two mentions have head-word match for actuator/patient;

Rule 3: if the two mentions have compatible time / location; (the compatible locations is defined as one location is a larger concept contained the other such as “North America” – “New York”; the compatible times is defined as one time unit is a more general concept than another such as “March, 1983” – “1983” but not “March” – “1983”)

Rule 4~19: the compatible pairs in Appendix A6.

Appendix A9: Spectral Graph Partitioning

In Appendix Section A9, we will show a brief introduction of the conventional spectral graph partitioning model. The definition and formulation is quite standard in the research community. Instead of restating them, we would like to reproduce an easy to understand version from CWiki Apache website⁶⁵.

Spectral clustering, a more powerful and specialized algorithm (compared to k -means), derives its name from spectral analysis of a graph, which is how the data are represented. Each object to be clustered can initially be represented as an n -dimensional numeric vector, but the difference with this algorithm is that there must also be some method for performing a comparison between each object and expressing this comparison as a scalar.

This n by n comparison of all objects with all others forms the *affinity* matrix, which can be intuitively thought of as a rough representation of an underlying undirected, weighted, and fully connected graph whose edges express the relative relationships, or affinities, between each pair of objects in the original data. This affinity matrix forms the basis from which the two spectral clustering algorithms operate.

The equation by which the affinities are calculated can vary depending on the user's circumstances; typically, the equation takes the form of: $e^{(-\frac{d^2}{c})}$ where d is the Euclidean distance between a pair of points and c is a scaling factor. c is often calculated relative to a k -neighborhood of closest points to the current point; all other affinities are set to 0 outside of the neighborhood. Again, this formula can vary

⁶⁵ <https://cwiki.apache.org/MAHOUT/spectral-clustering.html>

depending on the situation (e.g. a fully connected graph would ignore the k -neighborhood and calculate affinities for all pairs of points).

The spectral clustering is often use together with k-means clustering. This consists of a few basic steps of generalized spectral clustering, followed by standard k -means clustering over the intermediate results. Again, this process begins with an affinity matrix A - whether or not it is fully connected depends on the user's need.

A is then transformed into a pseudo-Laplacian matrix via a multiplication with a diagonal matrix whose entries consist of the sums of the rows of A . The sums are modified to be the inverse square root of their original values. The final operation looks something like:

$$L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

L has some properties that are of interest to us; most importantly, while it is symmetric like A , it has a more stable eigen-decomposition. L is decomposed into its constituent eigenvectors and corresponding eigenvalues (though the latter will not be needed for future calculations); the matrix of eigenvectors, U , is what we are now interested in.

Assuming U is a column matrix (the eigenvectors comprise the columns), then we will now use the *rows* of U as proxy data for the original data points. We will run each row through standard k -means clustering, and the label that each proxy point receives will be transparently assigned to the corresponding original data point, resulting in the final clustering assignments.

Appendix A10: Random Walk Graph Partitioning

In Appendix Section A10, we will show a brief introduction of the conventional Random Walk Model. Conventional random walk model is a well-established graph partitioning model. We have extracted the brief but essential explanations from Wikipedia⁶⁶ for the readers to digest easily.

A random walk is a mathematical formalization of a path that consists of a succession of random steps. A random walk of length k on a possibly infinite graph G with a root 0 is a stochastic process with random variables X_1, X_2, \dots, X_k such that $X_1 = 0$ and X_{i+1} is a vertex chosen uniformly at random from the neighbours of X_i . Then the number $P_{v,w,k}(G)$ is the probability that a random walk of length k starting at v ends at w . In particular, if G is a graph with root 0 , $P_{0,0,2k}$ is the probability that a $2k$ -step random walk returns to 0 . We can also construct a matrix T_k which the (i, j) -element of T_k is $T_{i,j,k} = P_{i,j,k}(G)$ for $i, j \in G$. The matrix T_k denotes the probability of the final node if we start the walk from node i and walk for k steps. In addition, if we make $k \rightarrow \infty$, we will get a matrix T_∞ which shows the probability of the final node for infinite number of steps. The matrix T_∞ is also referred as the stationary transition probability.

There are two ways to derive the matrix T_∞ . The first method is a closed form solution for T_∞ . We will not extend our study to the closed form solution as it may take a whole chapter to gain a thorough understanding. We would like to direct reader to a comprehensive survey paper on random walk (Lovász, 1993) for the closed form solution.

The second way is estimate the T_∞ through sampling. The sampling technique is easier to understand. Basically, it just conducts a sufficient large number of random

⁶⁶ http://en.wikipedia.org/wiki/Random_walk

walks and estimates the entries in T_∞ using the sample mean. The sampling way is used when the graph size is infeasible for a closed form solution (e.g. Hassan and Radev, 2010; Wang et al., 2012) or the random walks required certain special characteristics such as the self-interacting capability in this thesis and in (Riberio & Twosley, 2010). The self-interacting capability enables us to incorporate the linguistics constraints and preferences in a dynamic way.

Appendix B Empirical Model Settings

In Appendix Section B, we will show all the details on empirical model settings

Appendix B1: How to Tune Parameters with Training Data

In Appendix Section B1, we will show how to tune the parameters with the training data. Readers may be more familiar with the concept of the “development” data. In our case, the tuned parameters are from unsupervised model such as the Spectral Graph Partitioning, Random Walk Model and LDA Topic Models.

In the rest of this section, we will list all the empirical parameter we tuned with the training data.

For LDA Topic Modeling, we use the training data to select best size of key word list.

For Spectral Graph Partitioning model, we use the training data to select:

- (1). Best number of eigenvectors = 8;
- (2). Proximity Radius $e=0.1 \times 10^{-4}$;

For Spectral Graph Partitioning model, we use the training data to select:

- (1). Random Walk Sample Size = 100;
- (2). Mention Inclusion Threshold = 70;
- (3). Random Walk Preference Rules Weights Judgments

Appendix B2: 20 Runs of Experiments through Random Sampling of Training and Testing Data

In Appendix Section B2, we will show the detailed process to conduct 20 runs of experiments through random sampling of training and testing data.

For each run of the experiments, we split the corpus into an 80:20 proportion. 80% of corpus will be used for training while the other 20% of corpus will be used for testing. Thus for each run of experiment, the training data and testing data are mutually exclusive. For each run of experiment, the 80:20 split is random sampled. The random sampling process is repeated 20 times to create 20 different runs of experiment with different training/testing data.

For Example, we have 5 documents in the corpus $D = \{D_1, D_2, D_3, D_4, D_5\}$. One randomly sampled Training/Testing split can be $D_{Train} = \{D_1, D_3, D_4, D_5\}$ and $D_{Test} = \{D_2\}$. Another randomly sampled Training/Testing split can be $D_{Train} = \{D_1, D_2, D_4, D_5\}$ and $D_{Test} = \{D_3\}$. In this simple example, we cannot conduct 20 random sampling processes. But in OntoNotes4.0 corpus with 2000+ documents, 20 random sampling processing can be conducted without repetition.

The same process is also used by other researchers such as (Yin et al, 2009) & (Pan et al, 2009). Both of these works are from a top-ranked conference: International Joint Conference of Artificial Intelligence (IJCAI).

Appendix B3: Student's paired t-Test for Statistical Significances

In Appendix Section B3 we will show the details on the two-sample paired Student's t-test. Student's paired t-test is a well-defined hypothesis test for comparing the difference between two related samples. Part of the following information is taken from (Shier, 2004) on the Mathematical Learning Support Centre Website⁶⁷ for an easy understanding.

General Information on paired Student's t-Test

A paired t-test is used to compare two population means where you have two samples in which observations in one sample can be paired with observations in the other sample. Examples of where this might occur are:

- Before-and-after observations on the same subjects
- A comparison of two different methods of measurement or two different treatments where the measurements/treatments are applied to the same subjects.

Our scenario falls in the second cases which we try to compare the performances between two models applied to the same set of training/testing data.

Level of Significance

We select the most commonly used level of significance $\alpha = 5\%$.

One-Tailed vs. Two-Tailed t-Test

In our cases, we are comparing an improved model $M_{Improved}$ (improved with a proposed technique) with an ordinary model $M_{Ordinary}$ (without the proposed technique). We have a prior knowledge that the performance of $M_{Improved}$ is better

⁶⁷ <http://mlsc.lboro.ac.uk/resources/statistics/Pairedttest.pdf>

than the performance of $M_{Ordinary}$. Therefore, the one-tailed paired t-test is used instead of the two-tailed one. In other words, the hypothesis test is set as

$$H_0: Perf(M_{Improved}) - Perf(M_{Ordinary}) = 0$$

$$H_1: Perf(M_{Improved}) - Perf(M_{Ordinary}) > 0$$

Procedure to Conduct t-Test

We use the Microsoft Excel's built-in T-Test function to conduct out one-tailed two-sample paired t-test with 5% level of significance. Although t-test works on fewer samples, in statistical study, a sample size of 20 is in general more meaningful for conducting Student's t-test.

Appendix C: Experimental Results

In Appendix Section C, we will show all the experiment records for this thesis.

Appendix C1: 20 Sets of Experimental Results

In Appendix Section C1, we will show all the 20 sets of Experiments Results.

Experiment Set 1:

Model	P	R	F
BL	31.7%	54.9%	40.2%
BL+CC	38.6%	53.0%	44.7%
SGP	30.5%	66.7%	41.9%
SGP+CC	36.5%	65.7%	46.9%
SGP+CC+BIS	39.3%	65.2%	49.0%
SGP+CC+BIS+PCI	40.4%	66.1%	50.1%
SGP+CC+BIS+PCI+PIE	45.5%	62.6%	52.7%
SGP+CC+BIS+PCI+PIE+SC	46.9%	67.5%	55.3%
SGP+CC+BIS+PCI+PIE+SC+ODP	49.6%	67.3%	57.1%
CRW	37.3%	65.2%	47.5%
MRW	42.2%	68.1%	52.1%
MRW+PCI	43.5%	70.3%	53.7%
MRW+PCI+LCP	47.1%	68.9%	56.0%
MRW+PCI+LCP+OGI	50.7%	67.5%	57.9%

Experiment Set 2:

Model	P	R	F
BL	29.9%	53.7%	38.4%
BL+CC	34.4%	50.1%	40.8%
SGP	28.2%	65.6%	39.4%
SGP+CC	33.8%	62.9%	44.0%
SGP+CC+BIS	36.8%	62.1%	46.2%
SGP+CC+BIS+PCI	37.1%	63.6%	46.9%
SGP+CC+BIS+PCI+PIE	41.9%	60.7%	49.6%
SGP+CC+BIS+PCI+PIE+SC	44.1%	63.7%	52.1%
SGP+CC+BIS+PCI+PIE+SC+ODP	48.1%	61.1%	53.8%
CRW	33.2%	60.7%	42.9%
MRW	36.3%	62.1%	45.8%
MRW+PCI	35.9%	63.3%	45.8%
MRW+PCI+LCP	40.2%	60.8%	48.4%
MRW+PCI+LCP+OGI	44.5%	59.4%	50.9%

Experiment Set 3:

Model	P	R	F
BL	30.3%	56.2%	39.4%
BL+CC	35.3%	52.1%	42.1%
SGP	27.7%	67.6%	39.3%
SGP+CC	34.3%	65.9%	45.1%
SGP+CC+BIS	38.3%	65.6%	48.4%
SGP+CC+BIS+PCI	39.4%	66.8%	49.6%
SGP+CC+BIS+PCI+PIE	44.8%	62.1%	52.1%
SGP+CC+BIS+PCI+PIE+SC	46.0%	67.3%	54.6%
SGP+CC+BIS+PCI+PIE+SC+ODP	49.4%	64.1%	55.8%
CRW	35.7%	61.2%	45.1%
MRW	39.2%	62.7%	48.2%
MRW+PCI	41.7%	61.0%	49.5%
MRW+PCI+LCP	44.0%	59.4%	50.6%
MRW+PCI+LCP+OGI	47.3%	58.0%	52.1%

Experiment Set 4:

Model	P	R	F
BL	33.7%	50.5%	40.4%
BL+CC	38.2%	49.4%	43.1%
SGP	31.3%	66.2%	42.5%
SGP+CC	37.3%	64.3%	47.2%
SGP+CC+BIS	41.1%	62.1%	49.5%
SGP+CC+BIS+PCI	42.4%	64.0%	51.0%
SGP+CC+BIS+PCI+PIE	46.9%	63.7%	54.0%
SGP+CC+BIS+PCI+PIE+SC	49.0%	69.1%	57.3%
SGP+CC+BIS+PCI+PIE+SC+ODP	51.3%	66.9%	58.1%
CRW	31.9%	59.7%	41.6%
MRW	34.9%	61.8%	44.6%
MRW+PCI	35.1%	62.2%	44.9%
MRW+PCI+LCP	39.8%	59.4%	47.7%
MRW+PCI+LCP+OGI	42.6%	57.8%	49.0%

Experiment Set 5:

Model	P	R	F
BL	29.4%	57.5%	38.9%
BL+CC	36.0%	55.8%	43.8%
SGP	30.3%	69.7%	42.2%
SGP+CC	37.9%	67.1%	48.4%
SGP+CC+BIS	40.2%	66.7%	50.2%
SGP+CC+BIS+PCI	39.7%	67.4%	50.0%
SGP+CC+BIS+PCI+PIE	45.4%	64.7%	53.4%
SGP+CC+BIS+PCI+PIE+SC	46.7%	68.3%	55.5%
SGP+CC+BIS+PCI+PIE+SC+ODP	49.1%	66.0%	56.3%
CRW	39.7%	64.8%	49.2%
MRW	44.7%	65.2%	53.0%
MRW+PCI	45.9%	66.3%	54.2%
MRW+PCI+LCP	48.5%	63.2%	54.9%
MRW+PCI+LCP+OGI	53.7%	61.9%	57.5%

Experiment Set 6:

Model	P	R	F
BL	31.3%	55.5%	40.0%
BL+CC	38.5%	51.4%	44.0%
SGP	32.1%	66.8%	43.4%
SGP+CC	39.3%	64.1%	48.7%
SGP+CC+BIS	41.1%	64.7%	50.3%
SGP+CC+BIS+PCI	40.4%	65.8%	50.1%
SGP+CC+BIS+PCI+PIE	46.4%	61.7%	53.0%
SGP+CC+BIS+PCI+PIE+SC	47.9%	63.3%	54.5%
SGP+CC+BIS+PCI+PIE+SC+ODP	48.5%	64.1%	55.2%
CRW	35.3%	63.7%	45.4%
MRW	39.8%	64.5%	49.2%
MRW+PCI	41.1%	66.0%	50.7%
MRW+PCI+LCP	46.8%	63.3%	53.8%
MRW+PCI+LCP+OGI	48.3%	61.6%	54.1%

Experiment Set 7:

Model	P	R	F
BL	29.7%	54.7%	38.5%
BL+CC	35.8%	52.2%	42.5%
SGP	29.8%	64.6%	40.8%
SGP+CC	32.7%	62.0%	42.8%
SGP+CC+BIS	36.1%	60.9%	45.3%
SGP+CC+BIS+PCI	36.9%	61.1%	46.0%
SGP+CC+BIS+PCI+PIE	40.1%	59.7%	48.0%
SGP+CC+BIS+PCI+PIE+SC	41.2%	63.8%	50.1%
SGP+CC+BIS+PCI+PIE+SC+ODP	45.4%	62.1%	52.5%
CRW	36.8%	65.1%	47.0%
MRW	42.0%	67.3%	51.7%
MRW+PCI	43.1%	68.1%	52.8%
MRW+PCI+LCP	45.7%	65.0%	53.7%
MRW+PCI+LCP+OGI	49.8%	63.8%	55.9%

Experiment Set 8:

Model	P	R	F
BL	35.4%	51.9%	42.1%
BL+CC	41.6%	48.7%	44.9%
SGP	33.7%	69.3%	45.3%
SGP+CC	40.1%	63.6%	49.2%
SGP+CC+BIS	42.7%	61.2%	50.3%
SGP+CC+BIS+PCI	40.9%	63.3%	49.7%
SGP+CC+BIS+PCI+PIE	46.3%	60.1%	52.3%
SGP+CC+BIS+PCI+PIE+SC	47.7%	65.2%	55.1%
SGP+CC+BIS+PCI+PIE+SC+ODP	49.3%	63.7%	55.6%
CRW	34.2%	66.1%	45.1%
MRW	39.6%	68.2%	50.1%
MRW+PCI	40.2%	69.1%	50.8%
MRW+PCI+LCP	44.3%	64.7%	52.6%
MRW+PCI+LCP+OGI	49.2%	62.2%	54.9%

Experiment Set 9:

Model	P	R	F
BL	31.3%	54.9%	39.9%
BL+CC	39.4%	53.2%	45.3%
SGP	29.2%	68.1%	40.9%
SGP+CC	31.9%	66.0%	43.0%
SGP+CC+BIS	34.3%	64.7%	44.8%
SGP+CC+BIS+PCI	36.1%	65.5%	46.5%
SGP+CC+BIS+PCI+PIE	41.8%	62.2%	50.0%
SGP+CC+BIS+PCI+PIE+SC	43.4%	65.2%	52.1%
SGP+CC+BIS+PCI+PIE+SC+ODP	46.3%	64.9%	54.0%
CRW	35.7%	64.7%	46.0%
MRW	37.8%	65.0%	47.8%
MRW+PCI	36.4%	65.2%	46.7%
MRW+PCI+LCP	40.7%	63.8%	49.7%
MRW+PCI+LCP+OGI	45.8%	59.4%	51.7%

Experiment Set 10:

Model	P	R	F
BL	31.9%	55.0%	40.4%
BL+CC	41.1%	51.4%	45.7%
SGP	32.3%	66.9%	43.6%
SGP+CC	38.0%	64.9%	47.9%
SGP+CC+BIS	39.7%	64.1%	49.0%
SGP+CC+BIS+PCI	37.9%	66.0%	48.2%
SGP+CC+BIS+PCI+PIE	43.6%	60.1%	50.5%
SGP+CC+BIS+PCI+PIE+SC	43.7%	64.4%	52.1%
SGP+CC+BIS+PCI+PIE+SC+ODP	45.9%	62.9%	53.1%
CRW	38.2%	65.3%	48.2%
MRW	42.4%	67.2%	52.0%
MRW+PCI	43.0%	68.3%	52.8%
MRW+PCI+LCP	45.9%	64.1%	53.5%
MRW+PCI+LCP+OGI	48.7%	63.0%	54.9%

Experiment Set 11:

Model	P	R	F
BL	32.1%	56.7%	41.0%
BL+CC	40.2%	54.9%	46.4%
SGP	33.3%	68.2%	44.7%
SGP+CC	38.7%	66.2%	48.8%
SGP+CC+BIS	41.8%	65.6%	51.1%
SGP+CC+BIS+PCI	40.7%	67.0%	50.6%
SGP+CC+BIS+PCI+PIE	45.2%	65.0%	53.3%
SGP+CC+BIS+PCI+PIE+SC	46.7%	65.0%	54.4%
SGP+CC+BIS+PCI+PIE+SC+ODP	48.2%	66.7%	56.0%
CRW	40.7%	68.2%	51.0%
MRW	44.6%	70.3%	54.6%
MRW+PCI	45.0%	69.4%	54.6%
MRW+PCI+LCP	47.3%	65.9%	55.1%
MRW+PCI+LCP+OGI	51.4%	63.7%	56.9%

Experiment Set 12:

Model	P	R	F
BL	30.1%	56.2%	39.2%
BL+CC	35.6%	55.7%	43.4%
SGP	29.3%	65.9%	40.6%
SGP+CC	35.3%	64.0%	45.5%
SGP+CC+BIS	37.3%	64.1%	47.2%
SGP+CC+BIS+PCI	38.2%	65.3%	48.2%
SGP+CC+BIS+PCI+PIE	41.6%	63.7%	50.3%
SGP+CC+BIS+PCI+PIE+SC	44.1%	69.2%	53.9%
SGP+CC+BIS+PCI+PIE+SC+ODP	45.8%	67.7%	54.6%
CRW	37.2%	64.9%	47.3%
MRW	41.2%	65.7%	50.6%
MRW+PCI	40.8%	65.1%	50.2%
MRW+PCI+LCP	44.1%	62.7%	51.8%
MRW+PCI+LCP+OGI	45.9%	61.8%	52.7%

Experiment Set 13:

Model	P	R	F
BL	29.8%	52.1%	37.9%
BL+CC	33.0%	50.6%	39.9%
SGP	30.1%	63.9%	40.9%
SGP+CC	36.7%	64.1%	46.7%
SGP+CC+BIS	38.8%	63.3%	48.1%
SGP+CC+BIS+PCI	39.9%	64.4%	49.3%
SGP+CC+BIS+PCI+PIE	47.2%	59.8%	52.8%
SGP+CC+BIS+PCI+PIE+SC	49.1%	63.3%	55.3%
SGP+CC+BIS+PCI+PIE+SC+ODP	53.7%	60.6%	56.9%
CRW	38.4%	68.1%	49.1%
MRW	41.9%	71.3%	52.8%
MRW+PCI	39.3%	70.0%	50.3%
MRW+PCI+LCP	43.2%	66.6%	52.4%
MRW+PCI+LCP+OGI	48.3%	64.1%	55.1%

Experiment Set 14:

Model	P	R	F
BL	30.7%	54.4%	39.2%
BL+CC	36.9%	53.7%	43.7%
SGP	31.2%	65.1%	42.2%
SGP+CC	37.0%	63.9%	46.9%
SGP+CC+BIS	38.8%	62.8%	48.0%
SGP+CC+BIS+PCI	36.7%	63.0%	46.4%
SGP+CC+BIS+PCI+PIE	44.2%	62.1%	51.6%
SGP+CC+BIS+PCI+PIE+SC	46.0%	66.7%	54.4%
SGP+CC+BIS+PCI+PIE+SC+ODP	50.1%	62.9%	55.8%
CRW	36.4%	65.7%	46.8%
MRW	40.1%	67.7%	50.4%
MRW+PCI	42.4%	67.2%	52.0%
MRW+PCI+LCP	45.3%	64.1%	53.1%
MRW+PCI+LCP+OGI	50.7%	59.9%	54.9%

Experiment Set 15:

Model	P	R	F
BL	31.6%	53.9%	39.8%
BL+CC	38.4%	51.7%	44.1%
SGP	30.1%	64.6%	41.1%
SGP+CC	38.5%	65.1%	48.4%
SGP+CC+BIS	40.2%	65.2%	49.7%
SGP+CC+BIS+PCI	41.4%	66.7%	51.1%
SGP+CC+BIS+PCI+PIE	45.8%	63.2%	53.1%
SGP+CC+BIS+PCI+PIE+SC	45.1%	68.6%	54.4%
SGP+CC+BIS+PCI+PIE+SC+ODP	47.7%	67.6%	55.9%
CRW	39.1%	69.3%	50.0%
MRW	42.2%	73.7%	53.7%
MRW+PCI	43.1%	71.9%	53.9%
MRW+PCI+LCP	47.0%	65.3%	54.7%
MRW+PCI+LCP+OGI	49.8%	62.7%	55.5%

Experiment Set 16:

Model	P	R	F
BL	30.8%	54.7%	39.4%
BL+CC	35.8%	52.2%	42.5%
SGP	29.7%	64.6%	40.7%
SGP+CC	33.7%	65.1%	44.4%
SGP+CC+BIS	37.8%	63.9%	47.5%
SGP+CC+BIS+PCI	37.0%	65.1%	47.2%
SGP+CC+BIS+PCI+PIE	40.2%	63.4%	49.2%
SGP+CC+BIS+PCI+PIE+SC	42.1%	64.8%	51.0%
SGP+CC+BIS+PCI+PIE+SC+ODP	44.9%	62.2%	52.2%
CRW	31.4%	60.8%	41.4%
MRW	37.2%	61.7%	46.4%
MRW+PCI	38.4%	62.1%	47.5%
MRW+PCI+LCP	42.6%	60.5%	50.0%
MRW+PCI+LCP+OGI	46.3%	58.8%	51.8%

Experiment Set 17:

Model	P	R	F
BL	30.3%	55.2%	39.1%
BL+CC	36.2%	53.1%	43.1%
SGP	31.1%	64.6%	42.0%
SGP+CC	37.9%	62.8%	47.3%
SGP+CC+BIS	39.5%	63.7%	48.8%
SGP+CC+BIS+PCI	40.8%	65.1%	50.2%
SGP+CC+BIS+PCI+PIE	42.9%	64.4%	51.5%
SGP+CC+BIS+PCI+PIE+SC	45.1%	65.7%	53.5%
SGP+CC+BIS+PCI+PIE+SC+ODP	46.4%	63.0%	53.4%
CRW	38.2%	66.6%	48.6%
MRW	43.1%	66.2%	52.2%
MRW+PCI	42.3%	65.7%	51.5%
MRW+PCI+LCP	45.8%	62.3%	52.8%
MRW+PCI+LCP+OGI	49.0%	60.4%	54.1%

Experiment Set 18:

Model	P	R	F
BL	32.0%	54.2%	40.2%
BL+CC	39.1%	49.7%	43.8%
SGP	31.1%	64.6%	42.0%
SGP+CC	35.8%	63.1%	45.7%
SGP+CC+BIS	39.4%	62.2%	48.2%
SGP+CC+BIS+PCI	40.7%	63.1%	49.5%
SGP+CC+BIS+PCI+PIE	45.8%	61.2%	52.4%
SGP+CC+BIS+PCI+PIE+SC	46.3%	63.9%	53.7%
SGP+CC+BIS+PCI+PIE+SC+ODP	48.5%	62.2%	54.5%
CRW	40.2%	67.4%	50.4%
MRW	42.9%	68.3%	52.7%
MRW+PCI	43.1%	66.7%	52.4%
MRW+PCI+LCP	46.0%	63.1%	53.2%
MRW+PCI+LCP+OGI	49.3%	61.4%	54.7%

Experiment Set 19:

Model	P	R	F
BL	32.2%	53.4%	40.2%
BL+CC	40.6%	51.0%	45.2%
SGP	31.7%	65.3%	42.7%
SGP+CC	37.1%	63.0%	46.7%
SGP+CC+BIS	40.6%	60.9%	48.7%
SGP+CC+BIS+PCI	39.1%	64.1%	48.6%
SGP+CC+BIS+PCI+PIE	46.0%	60.0%	52.1%
SGP+CC+BIS+PCI+PIE+SC	47.8%	63.3%	54.5%
SGP+CC+BIS+PCI+PIE+SC+ODP	49.1%	62.8%	55.1%
CRW	39.8%	66.7%	49.9%
MRW	44.7%	68.2%	54.0%
MRW+PCI	46.4%	69.1%	55.5%
MRW+PCI+LCP	49.3%	65.4%	56.2%
MRW+PCI+LCP+OGI	53.1%	63.7%	57.9%

Experiment Set 20:

Model	P	R	F
BL	31.3%	54.0%	39.6%
BL+CC	37.4%	52.1%	43.5%
SGP	29.9%	66.4%	41.2%
SGP+CC	34.8%	65.3%	45.4%
SGP+CC+BIS	39.0%	63.9%	48.4%
SGP+CC+BIS+PCI	37.1%	64.2%	47.0%
SGP+CC+BIS+PCI+PIE	41.8%	60.3%	49.4%
SGP+CC+BIS+PCI+PIE+SC	40.4%	63.7%	49.4%
SGP+CC+BIS+PCI+PIE+SC+ODP	44.4%	62.1%	51.8%
CRW	33.8%	63.4%	44.1%
MRW	36.1%	64.2%	46.2%
MRW+PCI	37.0%	61.5%	46.2%
MRW+PCI+LCP	42.1%	59.8%	49.4%
MRW+PCI+LCP+OGI	48.0%	56.1%	51.7%

Appendix C2: List of p-Values for Student's paired t-Tests

In Appendix Section C2, we will show the p-Values when conducting Student's paired t-tests for statistical significance.

System 1	System 2	p-Value
BL	BL+CC	3.35848E-13
BL	SGP	6.7595E-09
BL+CC	SGP+CC	1.06443E-06
SGP	SGP+CC	5.8811E-13
SGP+CC	SGP+CC+BIS	2.172E-11
SGP+CC+BIS	SGP+CC+BIS+PCI	0.06366406
SGP+CC+BIS+PCI	SGP+CC+BIS+PCI+PIE	2.9307E-12
SGP+CC+BIS+PCI+PIE	SGP+CC+BIS+PCI+PIE+SC	2.6835E-10
SGP+CC+BIS+PCI+PIE+SC	SGP+CC+BIS+PCI+PIE+SC+ODP	2.8176E-08
BL+CC	CRW	8.2254E-10
CRW	MRW	9.7817E-14
MRW	MRW+PCI	0.05945349
MRW+PCI	MRW+PCI+LCP	1.0073E-07
MRW+PCI+LCP	MRW+PCI+LCP+OGI	5.41261E-11
SGP+CC+BIS+PCI+PIE	MRW+PCI+LCP	0.055852875
MRW+PCI+LCP+OGI	SGP+CC+BIS+PCI+PIE+SC+ODP	0.142003431