ENTITY/EVENT-LEVEL SENTIMENT DETECTION AND INFERENCE

by

Lingjia Deng

Bachelor of Engineering

Beijing University of Posts and Telecommunications

2011

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of

Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Lingjia Deng

It was defended on

August 25th 2016

and approved by

Dr. Janyce Wiebe, Department of Computer Science, University of Pittsburgh

Dr. Rebecca Hwa, Department of Computer Science, University of Pittsburgh

Dr. Yuru Lin, School of Information Science, University of Pittsburgh

Dr. William Cohen, Machine Learning Department, Carnegie Mellon University

Dissertation Advisors: Dr. Janyce Wiebe, Department of Computer Science, University of Pittsburgh,

Dr. Rebecca Hwa, Department of Computer Science, University of Pittsburgh

ENTITY/EVENT-LEVEL SENTIMENT DETECTION AND INFERENCE

Lingjia Deng, PhD

University of Pittsburgh, 2016

Sentiment analysis aims at recognizing and understanding opinions expressed in languages. Previous work in sentiment analysis focused on extracting explicit opinions, which are directly expressed via sentiment words. However, opinions may be expressed implicitly via inferences over explicit sentiments. For example, in the sentence *It is great that he was promoted.* versus *It is great that he was fired*, there is an explicitly positive sentiment in both sentences because of the positive sentiment word *great*. Previous work may stop here. However, the sentiment toward *he* in the former sentence is positive, while the sentiment toward *he* in the later sentence is negative. The sentiments toward *he* in both sentences are implicit since there is no sentiment word directly modifying *he*. The implicit opinions are indicated in the text, and they are important for a sentiment analysis system to fully understand the documents. While previous work cannot recognize such implicit sentiment, this thesis contributes to developing an entity/event-level sentiment analysis system to recognize both explicit and implicit sentiments expressed from entities toward entities and events.

Specifically, we first give the definitions of the entity/event-level sentiment analysis task. Since this is a new task, we develop two corpora serving as resources for this task. The implicit sentiments cannot be recognized merely relying on sentiment lexicons since the implicit sentiments are not directly associated with sentiment words. Inference rules are needed to recognize the implicit sentiments. Instead of developing a rule-based system to automatically infer implicit opinions, we develop computational models which use the inference rules as soft constraints. What's more important, the models take into account the information not only from sentiment analysis tasks, but also from other Natural Language

Processing tasks including information extraction and semantic role labeling. The models jointly solve different NLP tasks in one single model and improve the performances of the tasks. We also contribute to improving recognizing sources of opinions in this thesis. Finally, we conduct an analysis study showing that the idea of sentiment inference defined in this thesis can be applied to Chinese text as well.

TABLE OF CONTENTS

PREFACE xiv

1.0 INTRODUCTION 1

1.1 Motivation for sentiment inference 2

1.2 Research Overview 4

1.2.1 Research Hypothesis 5

1.2.2 Research Methodology 5

1.2.3 Research Summary 7

1.3 Main Contributions 9

1.4 Outline 10

2.0 BACKGROUND 11

2.1 Works on different granularities of sentiment analysis 11

2.2 Works on sentiment inference 14

2.3 Works on joint prediction models and applications 15

2.4 Summary 16

3.0 REPRESENTATIONS OF ENTITY/EVENT-LEVEL SENTIMENT 17

3.1 Representation of sentiments 18

3.2 Representation of +/-Effect Events 20

3.2.1 Definition of +/-effect event 20

3.2.1.1 +/-Effect Event. 21

3.2.1.2 Agent and Theme. 22

3.2.1.3 Influencer. 22

3.2.2 Representation of +/-Effect Events 23

3.3 Related Work . . . 23
3.4 Summary . . . 25
4.0 CORPORA OF ENTITY/EVENT-LEVEL SENTIMENT . . . 27
4.1 +/-Effect Event Sentiment Corpus . . . 28
4.1.1 Annotation Scheme . . . 29
4.1.2 Agreement Study . . . 30
4.1.2.1 Data and Agreement Study Design . . . 30
4.1.2.2 Agreement Study Evaluation . . . 31
4.1.2.3 Agreement Study Results . . . 33
4.1.2.4 Consensus Analysis . . . 34
4.1.3 Statistics and Examples . . . 34
4.2 MPQA 3.0: Entity/Event-level Sentiment Corpus . . . 36
4.2.1 Annotation Scheme: From MPQA 2.0 to MPQA 3.0 . . . 37
4.2.1.1 Examples . . . 39
4.2.1.2 MPQA 3.0 Annotation Scheme . . . 42
4.2.2 Agreement Study . . . 42
4.2.2.1 Disagreement Analysis . . . 43
4.2.3 Statistics and Examples . . . 44
4.3 Related Work . . . 45
4.4 Summary . . . 47
5.0 RULES OF ENTITY/EVENT-LEVEL SENTIMENT INFERENCE . . . 48
5.1 +/-Effect Event Inference Rules . . . 49
5.1.1 +/-Effect Event Inference Rules . . . 50
5.1.2 Validating the Rules: A Graph-based Propagation Model . . . 52
5.1.2.1 Definition of the Entity Graph. . . . 53
5.1.2.2 Sentiment Inference via LBP . . . 54
5.1.2.3 Performance of Graph Model. . . . 56
5.2 Sentiment Inference Rules . . . 57
5.3 Related Work . . . 59
5.4 Summary . . . 60

6.0 COMPUTATIONAL MODELS OF ENTITY/EVENT-LEVEL SENTIMENT DETECTION AND INFERENCE . . . 61
6.1 +/-Effect Event Sentiment Inference . . . 64
6.1.1 Integer Linear Programming model . . . 65
6.1.1.1 Co-reference In the Model . . . 69
6.1.2 Local Systems . . . 70
6.1.3 Experiment and Result . . . 73
6.1.3.1 Experiment Data . . . 73
6.1.3.2 Baseline Methods and Evaluation Metrics . . . 73
6.1.3.3 Results . . . 75
6.1.4 Examples . . . 76
6.2 Entity/Event-level Sentiment Inference . . . 77
6.2.1 Probabilistic Soft Logic . . . 78
6.2.1.1 PSL for Entity-Event-level Sentiment Analysis. . . . 80
6.2.2 Local Systems . . . 80
6.2.2.1 Explicit Sentiments. . . . 80
6.2.2.2 +/-Effect Events. . . . 83
6.2.3 Experiment and Results . . . 84
6.2.3.1 Baselines. . . . 84
6.2.3.2 Evaluations. . . . 85
6.3 Related Work . . . 89
6.4 Summary . . . 89
7.0 RECOGNIZING SOURCES OF OPINIONS . . . 91
7.1 Definitions of Two Types of Opinions . . . 93
7.2 Model . . . 94
7.2.1 Classifying Two Types of Opinions. . . . 94
7.2.2 Recognizing Sources of Two Types of Opinions . . . 97
7.3 Experiments and Results . . . 100
7.3.1 Performance of Recognizing Sources. . . . 101
7.3.2 Contribution of Transductive SVM . . . 103

7.3.3 Discussion of Trained PSL . . . 104
7.4 Related Work . . . 104
7.5 Summary . . . 105
8.0 SENTIMENT INFERENCE IN CHINESE . . . 106
8.1 Chinese Implicit Opinions Corpus . . . 107
8.1.1 Agreement Study . . . 107
8.1.1.1 Data. . . . 107
8.1.1.2 Evaluation and Result. . . . 108
8.1.2 +/-Effect Event Triggered by Chinese Syntax . . . 110
8.2 Inference Rules For Chinese . . . 112
8.2.1 Blocking the Inference . . . 114
8.3 Computational Model in Chinese . . . 116
8.3.1 Chinese +/-Effect Event Lexicon . . . 116
8.3.2 Chinese Reversers . . . 118
8.3.3 Syntax of Agent/theme in Chinese . . . 119
8.3.4 Chinese Explicit Sentiment Analysis . . . 121
8.4 Related Work . . . 121
8.5 Summary . . . 122
9.0 FUTURE DIRECTIONS . . . 123
9.1 Rules Using Intra-Document Knowledge . . . 125
9.1.1 Co-reference Resolution . . . 125
9.1.2 Agree . . . 125
9.1.3 Opinion-oriented Discourse Models . . . 126
9.1.4 Aspect-Based Sentiment Analysis . . . 126
9.1.5 (Non-)Reinforcing Sentiment Analysis. . . . 127
9.2 Rules Using Extra-Document Knowledge . . . 128
9.2.1 Entity Linking. . . . 128
9.2.2 Ideology. . . . 128
9.3 Summary . . . 129
10.0 CONCLUSIONS . . . 130

APPENDIX A. SENTIMENT REPRESENTATION RULES 135
APPENDIX B. SENTIMENT INFERENCE RULES W.R.T. +/-EFFECT EVENT 136
APPENDIX C. SENTIMENT INFERENCE RULES W.R.T. SENTIMENT-TOWARD-SENTIMENT STRUCTURE 137
BIBLIOGRAPHY 138

LIST OF TABLES

2.1 Expected outputs for (Ex1.1) in Chapter 1 from different granularity sentiment analysis systems. . . . 12
3.1 Representations of entity/event-level sentiment . . . 20
3.2 Representations of +/-effect event . . . 23
4.1 Span overlapping agreement $agr(A, B)$ in agreement study and consensus study. . . . 33
4.2 κ for label agreement. . . . 33
5.1 Combinations of rules inferring sentiments toward +/-effect event and the entities. . . . 52
5.2 Loopy Belief Propagation . . . 54
5.3 Performance of graph model itself. . . . 57
6.1 Truth table of being reversed or not (k is omitted) . . . 69
6.2 Performances of sentiment detection . . . 75
6.3 Precision@N of most important entity/event-target. . . . 86
6.4 Accuracy comparing PSL models (ET3 used for all) . . . 87
6.5 F-measure comparing PSL models (ET3 used for all) . . . 88
6.6 Comparison of entity/event-target selection methods (PSL3 used for all) . . . 88
7.1 Examples of Opinions, Sources of opinions, and Categories of opinions . . . 91
7.2 Rules in PSL . . . 97
7.3 Performances of Recognizing Sources . . . 102
7.4 Comparisons to State-of-the-art Models . . . 103
8.1 Results for Agreement Study Analysis. . . . 109

8.2 κ for Agreement Study Analysis. 111
8.3 Performance of Graph-Based Model in Chinese. 114
8.4 Counts of Chinese-English Corresponds 117
A1 Sentiment Representation Rules. 135
B1 Sentiment Inference Rules w.r.t +/-Effect Event 136
C1 Sentiment Inference Rules w.r.t. Sentiment-Toward-Sentiment Structure. . . . 137

LIST OF FIGURES

4.1 Structure in MPQA 3.0. 38
6.1 Overview of Subtasks. 62

PREFACE

The past five years of PhD study have been an incredible journey. The original purpose of pursuing a PhD was investigating what Natural Language Processing is and what I can contribute to this field of research. Luckily the purpose has been retained during my PhD and it will not end after I graduate. Teaching a computer to automatically extract the information humans need from texts, which is one of the goals of Natural Language Processing, is a complicated task. The most fascinating aspect of this task is that it presents the giant variety and incredible creativity of human languages. There is never a fixed algorithm to model all the language phenomenon. I feel really proud to be born as a human being who is able to express and understand ideas in languages. I feel even more proud to participate in developing systems to automate this process. As we advance in "inventing" machines, we progress in "discovering" the world as well.

This thesis would not have been possible without the support of many people. The guidances and help from my major advisor, Janyce Wiebe, cannot be summarized in a few sentences. During my PhD study, Jan has lightened the directions of my research, gave me the courage of paving the unknown world, and accompanying with me to face both disappointments and joys. I really enjoy the hours we spent in discussing the problems, coming up with the ideas, and working on the writings. I have also been lucky to learn from incredible mentors, including my committee members, Rebecca Hwa, Yuru Lin and William Cohen. Their valuable suggestions and comments on this thesis have polished it from a summary of my publications to an organized piece of work. I have also learnt a lot from so many wonderful teachers in the graduate school, such as Diane Litman and Milos Hauskrecht. The knowledge I gained from both their courses and daily talking have contributed to my foundations of knowledge and inspiration of research. This thesis contains

much work of annotations, which are resources of the work in this thesis and resources of future work in the field. The human-labelled annotations cannot be accomplished without the help from my advisor Jan Wiebe, my peer Yoonjung Choi, Changsheng Liu and Fan Zhang. Not limited to the annotations, the interesting observations of languages we have found during the annotating process contributed to my thesis as well. I have also gained much knowledge and support from many researchers outside the University of Pittsburgh. I would like to gratefully thank Claire Cardie from the Cornell University. Though my PhD, the grant have supported me in trying out new ideas and working on solid experiments, especially in the last summer the grant enabled me to work on writing my thesis. I also appreciated the inspiring discussions Claire and Jan had during the grant meetings and academic conferences, together with Rada Mihalcea and Carmen Banea from the University of Michigan.

I am thankful for the kind people I have come across in my life, including my parents who supported me without any condition, my roommates who created a sweet atmosphere in our apartment, my friends in Pittsburgh who walked me through every corner of this beautiful city and every corner of my PhD life. If I were a tiny twinkle star in the universe, your accompany made us as shiny as the milky way.

1.0 INTRODUCTION

Subjectivity analysis is an important task in the field of Natural Language Processing (NLP). As defined in (Wiebe et al., 1999, 2004), subjectivity refers to the linguistic expressions of speculations, evaluations, sentiments, belief, etc (i.e., private states). **Sentiment** is a particular subtype of subjectivity. It refers to the linguistic expressions of feelings and emotions. The other subtypes of subjectivity include arguing, belief, agreement, etc. In this thesis, we work on analyzing sentiments in the text.

There are many opinions expressed in various genres, including reviews, newswire, editorial, blogs, etc. The works in sentiment analysis and opinion mining are continuously moving forward. Early works in opinion mining focus on document-level analysis, e.g., judging the writer's attitude toward a product or a movie by analyzing the writer's review (Pang et al., 2002; Turney, 2002). To fully understand and utilize the opinions, many works in sentiment analysis and opinion mining in recent years begin to focus on more fined-grained levels, including sentence-level sentiment analysis which detects the sentiment expressed by a sentence (Yu and Hatzivassiloglou, 2003; McDonald et al., 2007), phrase-level sentiment analysis which recognizes the sentiment expressed by a phrase expressed as a text span (Yang and Cardie, 2013a; Johansson and Moschitti, 2013a), and aspect-based sentiment analysis which recognizes the sentiment expressed toward an aspect of a certain product (e.g., the battery quality of a computer) (Hu and Liu, 2004; Titov and McDonald, 2008), etc. A fine-grained sentiment analysis gives us an opportunity to better understand different opinions the writer expresses throughout the document by extracting the components of the **opinions frames** defined in (Wiebe et al., 2005; Deng and Wiebe, 2015b): the **source** (whose sentiment is it), the **polarity** (positive or negative), and the **target** (what is the sentiment toward).

1.1 MOTIVATION FOR SENTIMENT INFERENCE

Most of the previous works make the assumption that the opinion frames are revealed by certain opinion expressions. Consider two example sentences. In the sentence "*It is great that he was promoted.*" versus "*It is great that he was fired.*", there is an explicitly positive sentiment in both sentences because of the positive sentiment word great. Most of the previous works stop here. However, the sentiment toward him in the former sentence is positive, while the sentiment toward him in the later sentence is negative. The sentiments toward him in both sentences are implicit, since there is no sentiment word directly modifying him. Previous works only recognize the positive sentiment indicated by the word great, while the goal in this thesis aims at recognizing the sentiments expressed toward him as well, though there is no sentiment word modifying that person in the sentences.

In this thesis, we define the sentiment indicated by the word great as **explicit sentiments** where the sentiments are expressed via sentiment words or phrases. Different from them, the sentiments toward him are defined as **implicit sentiments** where the sentiments are not directly expressed toward the target, but it requires **inference** to understand the opinions. Conducting such inferences is easy for humans but difficult for automatic systems. Let's consider more example sentences below.

(Ex1.1) It is great that Hillary Clinton defeated Donald Trump.
(Ex1.2) It is disappointing that Hillary Clinton defeated Donald Trump.
(Ex1.3) It is great that Mike Pence stands by Donald Trump.
(Ex1.4) It is disappointing that Mike Pence stands by Donald Trump.

In each example, the phrases highlighted in red are negative sentiment expressions and the phrases highlighted in green are positive sentiment expressions. The phrases in red are the entities that the speaker is negative toward, and the phrases in green are the entities that the speaker is positive toward. The sentiments revealed by the highlighted phrases are explicit sentiments. In all the examples, the sources of the explicit sentiments are the writer, and the targets of the explicit sentiments are the clauses. Besides the explicit sentiments, the writer also has opinions toward the people in the sentences. Let's step through each example below.

In (Ex1.1), as revealed by the explicit sentiment, the writer is positive toward the defeating event. It is indicated that the writer is negative toward Trump since being defeated is a bad thing for Trump, and the writer is happy to see something bad happening to him. Further, the writer may be positive toward Clinton because she is the one who accomplished the defeating event. On the contrary, in (Ex1.2), the writer is negative toward the defeating event, as revealed by the explicit sentiment word, disappointing. Since the writer is not happy to see something bad happening to Trump, this indicates a positive sentiment toward Trump, in contrast to (Ex1.1). Analogously, the sentiment toward Clinton is different from the sentiment toward Clinton in (Ex1.1). In both (Ex1.1) and (Ex1.2), two persons (Clinton and Trump) are involved in a defeating event. An explicit sentiment is expressed toward the event in both sentences. Comparing (Ex1.1) and (Ex1.2), we find that different sentiments towards the same event indicate different sentiments toward the same entity, and the source (writer) have different sentiments toward different entities in the same event. What's more important, although the source (writer) does not use any sentiment word modifying Clinton or Trump, the sentiments are actually indicated in the sentence.

We see similar contradictions when comparing (Ex1.3) and (Ex1.4). In (Ex1.3), the writer is positive toward the standing by event. It is indicated that the writer is positive toward Trump since someone standing by him is a good thing for him, and the writer is happy to see something good happening to him. Further, the writer may be positive toward Pence as well because it is he who supports Trump. On the contrary, in (Ex1.4) the writer is negative toward the standing by event. It is indicated that the writer may be negative toward both Trump and Pence.

When we analyze how the implicit sentiments are indicated in the sentences, we find that the inferences arise from interactions between explicit sentiments and events such as fire, defeat, which negatively affect the themes (**-effect events**), and events such as promote, stand by, which positively affect the themes (**+effect events**). In this thesis, we are particular interested in analyzing how to infer implicit sentiments based on explicit sentiments and the information of such +/-effect events.

In the examples above, the same source (i.e., the writer) may have the same sentiment or different sentiments toward different entities in the same event, and the sentiments to-

ward the entities are implicit. The sentiments toward entities, which are implicit, can be hardly recognized by the state-of-the-art works. This indicates that we need a more fine-grained sentiment analysis system than the state-of-the-art tasks, motivating us to define a new sentiment analysis task and contributes methods to this new task, the **entity/event-level sentiment analysis** task where the source and the target of an opinion frame are either entities or events. As we have seen in (Ex1.1)-(Ex1.4), many sentiments expressed toward entities (e.g., persons) are implicit. Thus, a good system developed for solving the entity/event-level sentiment analysis task should be able to recognize both explicit sentiments and implicit sentiments. In this task, a positive entity/event-level sentiment is named PositivePair(s,t), representing there are positive sentiments expressed from the source s toward the target t. Similarly a negative entity/event-level sentiment is named NegativePair(s,t). For example, in (Ex1.1) above, the positive pairs include PosivePair(writer, defeated) and PositivePair(writer, Hillary Clinton). The negative pairs include NegativePair(writer, Donald Trump).

In summary, this work contributes to detecting and inferring both explicit and implicit PositivePairs and NegativePairs in the sentences.

1.2 RESEARCH OVERVIEW

The ultimate goal of this thesis is to utilize the +/-effect event information, the explicit sentiments, and the inference rules to improve the recognition of both explicit and implicit sentiments for the entity/event-level sentiment analysis task. To accomplish this goal, this thesis defines representations of this new task, develops annotated corpora as resources of the task, presents the sentiment inference rules and investigates joint prediction models integrating various clues in the sentences to automatically recognize both explicit and implicit sentiments expressed among entities and events in the text. Note that the inference rules in this thesis can be blocked in certain contexts. Thus, the joint prediction models we have developed use the inference rules as soft constraints instead of hard constraints. Though the focus of this thesis is on the investigation of recognizing implicit sentiments, it

also contributes to improving recognizing explicit sentiments by developing new models for recognizing sources of opinions.

1.2.1 Research Hypothesis

As we work toward the goal described above, we explore the following six hypotheses:

Hypothesis 1. Annotation schemes can be developed to guide annotators to reliably annotate expressions of +/-effect events, the agents and the themes, and their attributes.

Hypothesis 2. Annotation schemes can be developed to guide annotators to reliably annotate expressions of both entity/event-level explicit and implicit sentiments and their attributes.

Hypothesis 3. Inferences to perceive implicit sentiments can be represented to build automatic systems.

Hypothesis 4. +/-Effect event information is conductive to sentiment inference.

Hypothesis 5. Joint prediction models can be developed and can improve automatically recognizing entity/event-level sentiments.

Hypothesis 6. Categorizing opinions according to whether the source is a participant of the opinion or not can improve the recognition of opinion sources.

Hypothesis 7. Inferences are not limited to English text only.

1.2.2 Research Methodology

To test the first and second hypotheses, we follow the steps of previous works in developing corpora. As stated in (Wilson, 2008), a corpus linguistics approach involves: 1) developing a conceptual representation for the linguistic phenomena of interest, 2) developing coding schema and manual annotation instructions for the conceptual representation, 3) training annotators and conducting inter-annotator agreement studies, 4) producing the annotated corpus, and 5) analyzing the corpus to gain insight into how the linguistic phenomena of interest are expressed in context. Therefore, we first define the representations of the tasks. Then we develop manual annotation schemes in iterations. Inter-annotator agreement studies

are used to validate the reliability of the annotation schemes. If the +/-effect event information or the entity/event-level sentiments and their attributes can be reliably annotated, trained annotators will be able to achieve acceptable levels of agreement in an annotation study, as measured by standard agreement metrics used in the previous works, such as average F-measure (Wilson and Wiebe, 2003; Johansson and Moschitti, 2013b) and κ value . Following the commonly used convention in the field of NLP Wilson (2008), an agreement value of 0.80 allows for firm conclusions to be made, and a value of at least 0.67 is sufficient for drawing tentative conclusions. An agreement value of 1 indicates perfect agreement. Thus, we consider an agreement value of 0.67 or more as evidence supporting the first and the second hypothesis, and a higher agreement value indicates stronger evidence.

To test the third hypothesis, we use propositional logics to express the inference rules introduced in (Wiebe and Deng, 2014). We express the rules using the representations we have defined from the first hypothesis. Since there are many ways to present the rules and it is consistent for us to use the representations we have defined, we do not evaluate the form of the rules. However, we evaluate whether the rules can help sentiment analysis in a later hypothesis.

To test the fourth hypothesis, we use an experimental paradigm in which only the output sentiments are automatically computed while all the inputs are taken from the gold standard. We use the inference rules to define the model, and the experiments are run on the corpus we have developed. Since it is an unsupervised system, we do not separate the corpus as training set and testing set, but we still randomly select several documents as development set to understand the data. We carry out an intrinsic evaluation of the inference ability of the rules. We address the extrinsic evaluation in the fifth hypothesis.

To test the fifth hypothesis, we use an experimental paradigm in which we use the inference rules as constraints of the unsupervised systems developed for detecting sentiments. We use the state-of-the-art works as baselines. Such baselines not only represent state-of-the-art performances in sentiment analysis which do not use inferences, but also represent the pipeline paradigm compared to the joint paradigm in this thesis. We measure the performance of each system using standard metrics, including F-measure, recall, precision, and accuracy. To evaluate whether a given system performs better than the baseline for a

particular task, we use statistical significance tests. Following Wilson (2008), we consider a statistical significance test with a p-value < 0.05 to be evidence that a given automatic system performs better than the baseline.

To test the sixth hypothesis, we use an experimental paradigm that first a supervised system is trained to classify the opinions according to a new categorization of opinion types first developed in this thesis, then a joint model is used to recognize the sources of opinions. We develop several baselines in the experiment, and we also compare to the state-of-the-art works. We measure the performances of systems using standard metrics, including F-measure, recall, and precision. The paired t-test is used in the statistical significance test.

To test the final hypothesis, we follow the same experimental paradigms used to test the hypothesis mentioned above, but this time, the experiments are carried out on Chinese text. We also analyze the different experiment results or different factors leading to the different experiment results.

Taken together, the methodologies and approaches described above follow those of an ongoing research in the field of Natural Language Processing. First of all, the representations of the problem are defined. Then, the corpora of the problem are developed. Finally, the computational models are developed and evaluated.

1.2.3 Research Summary

First of all, we define the representations of the new task in this thesis, the **entity/event-level sentiment analysis**. We define the representations of the sentiments that this task aims to identify. There are two elements that are well defined for the first time in this thesis. (1) The first element includes the definitions of explicit sentiment and implicit sentiment, and the differences between them. Briefly speaking, an explicit sentiment is always associated with an opinion expression linking the source to the target of the sentiment, while an implicit sentiment is not directly associated with an opinion expression. We have developed representations to depict both explicit sentiments and implicit sentiments. The representations are fundamental to the new task. (2) The second element includes the definitions of +/-effect events. The inferences arise from interactions between explicit sentiment expres-

sions and events such as defeat in (Ex1.1) in Chapter 1, which negatively affect the themes of the events (-effect events), and events such as stand by, which positively affect the themes of the events (+effect events). Though a few works in sentiment analysis have utilized similar events as clues to find more sentiments (Zhang and Liu, 2011; Anand and Reschke, 2010; Reschke and Anand, 2011; Goyal et al., 2012), we are the first to give a full definition of such events in the thesis.

Next, we introduce the corpora we have developed to serve as the resources for this thesis and further research. The first corpus, *+/-Effect Event Sentiment Corpus*, is annotated with the writer's sentiments toward the agents and themes of +/-effect events (Deng et al., 2013). The second corpus, *MPQA 3.0*, is annotated with the entity/event-level sentiments that are expressed by the writer or an entity in the text toward the other entities or events in the text (Deng and Wiebe, 2015b). We will introduce the annotation scheme, the agreement study and results, the corpus statistics and several examples of both corpora.

We present the inference rules used as guidances for an automatic system to recognize implicit sentiments. Two sets of rules are defined based on the representations. The first set of rules infers the sentiment expressed toward the +/-effect events and the entities participating the events (i.e., the agents and the themes). We also develop a graph-based model to demonstrate that the rules can give correct inferences in 89% in the +/-Effect Event Sentiment Corpus. The second set of rules infers the sentiments expressed from any entity toward any entity or event, which are not limited to +/-effect events. The second set of rules especially handles a new case in the field of sentiment analysis, which is that the target of a sentiment is another sentiment (i.e., *sentiment toward sentiment* structure).

Based on the representations and inference rules, we introduce computational models developed to automatically recognize sentiments from the sentences. Two models are developed. The first model, as a pilot study, recognizes the writer's sentiments toward the agents and themes of the +/-effect events in the +/-Effect Event Sentiment Corpus (Deng et al., 2014). The second model recognizes the entity/event-level sentiments in the MPQA 3.0 corpus (Deng and Wiebe, 2015a).

Not only limited to the joint prediction models, we also contribute to improving individual components used in the joint prediction models. In the works of recognizing sources

of opinions, previous works mainly classify whether the source of an opinion is the writer or a noun phrase in the sentence. Different from them, we introduce the novel idea that an opinion should be classified as for whether the source of it is a participant in the event that triggers the opinion, or the source is not a participant. Based on this new categorization, we develop models in improving recognizing sources of opinions (Deng and Wiebe, 2016b).

Last but not the least, we have carried out a study demonstrating that the inference rules can also be applied to Chinese (Deng and Wiebe, 2014a).

We also discuss some future directions pertaining to the development of new rules that can connect sentiment analysis to the vast amount of knowledge as future directions (Deng and Wiebe, 2016a).

1.3 MAIN CONTRIBUTIONS

The research in this thesis contributes to an on-going line of research in subjectivity analysis. The main contribution of this thesis is the investigation of joint prediction models that integrate knowledge from Natural Language Processing (NLP) tasks other than sentiment analysis to improve recognizing sentiments from texts. We summarize the main contributions that the thesis devotes to the field of sentiment analysis and opinion mining.

- Defining +/-effect events and the components of such events.
- Defining the new sentiment analysis task, the entity/event-level sentiment analysis.
- Developing the +/-Effect Event Sentiment Corpus annotated with the writer's sentiments toward the agents and the themes of the +/-effect events in the text.
- Augmenting the original MPQA 2.0 corpus as the developed MPQA 3.0 corpus that is annotated with entity/event-level sentiments expressed among entities and events.
- Presenting sentiment inference rules to infer more sentiments in propositional logics.
- Building a graph-based model to validate that the inference rules can give correct inferences of sentiments.
- Building joint prediction models to automatically detect and infer entity/event-level sentiments in the text.

- Defining a new categorization of opinions based on whether the source is a participant of the event that triggers the opinion or the source is not a participant, which improves recognizing sources of opinions.
- Investigating whether the sentiment inferences can be applied to Chinese as well.

1.4 OUTLINE

The representations of the +/-effect events and the entity/event-level sentiments will be given in Chapter 3. The definitions will be introduced first, followed by the representations. Next, in Chapter 4 we will introduce the corpora we have developed to serve as the resources for this thesis and further research. Two corpora have been developed. We will introduce the annotation scheme, the agreement study and results, the corpus statistics and several examples of both corpora.

In Chapter 5, we present the inference rules used as guidances for an automatic system to recognize implicit sentiments. Two sets of rules are defined based on the representations given in Chapter 3.

In Chapter 6, we introduce the computational models developed to automatically recognize sentiments from the sentences. Two models are developed and evaluated.

In Chapter 7, we talk about our new categorization of opinions, based on which we develop the model to improve recognizing sources of opinions.

Last but not the least, we have carried out a study demonstrating that the inference rules are also applied to Chinese in Chapter 8 and we discuss the idea of new rules that can connect sentiment analysis to the vast amount of knowledge as future directions in Chapter 9.

Finally we give the conclusions in Chapter 10.

The rules used in different tasks in this thesis are listed as appendices.

2.0 BACKGROUND

In this chapter, we give the background of this thesis. We will introduce the background works from three perspectives: (1) works on different granularities of sentiment analysis (2) works on sentiment inference (3) works on joint prediction models and applications.

2.1 WORKS ON DIFFERENT GRANULARITIES OF SENTIMENT ANALYSIS

The definitions of subjective and objective were first given in (Wiebe et al., 2005; Wilson, 2008). Subjective expressions are words and phrases being used to express mental and emotional states, such as speculations, evaluations, feelings, emotions, stances and beliefs. A general covering term for such states is private state (Quirk et al., 1985), an internal state that cannot be directly observed or verified by others (Wiebe et al., 2005; Wilson, 2008). In this thesis, we focus on recognizing one type of the subjectivities, sentiment. As defined in (Wiebe et al., 2005; Wilson, 2008), a sentiment has a source and one or more than one targets. The source is to whom the opinion is attributed, and the target is toward whom or what the opinion is expressed. For example, it is also annotated in the subjectivity corpus such as MPQA (Wiebe et al., 2005) that each opinion has an opinion expression, the phrase which reveals the sentiment.

(Ex1.1) It is great that Hillary Clinton defeated Donald Trump.

Different works in sentiment analysis differ on the granularities of sentiments defined in their task. We summarize different sentiment analysis works recognizing the opinions

Granularity	Output
document-level	This article is positive.
sentence-level	This sentence has positive opinions.
phrase-level	⟨writer, *great*(positive), *Hillary Clinton defeated Trump*⟩
entity/event-level	PositivePair(writer, *defeated*)
	PositivePair(writer, *Hillary Clinton*)
	NegativePair(writer, *Donald Trump*)

Table 2.1: Expected outputs for (Ex1.1) in Chapter 1 from different granularity sentiment analysis systems.

in different granularities in Table 2.1, using (Ex1.1) in Chapter 1 as an example which is repeated below.

At the beginning in the field of sentiment analysis, researchers focused on judging whether a document (especially a review document) is a praising document or a criticizing document (Pang et al., 2002; Turney, 2002). Later some people proceeded to judge whether a sentence is a praising sentence or a criticizing sentence (Yu and Hatzivassiloglou, 2003; McDonald et al., 2007). The source of the sentiment in their works is the writer of the document (Yu and Hatzivassiloglou, 2003; McDonald et al., 2007). The target is usually the topic of the document, though not specified. We call these works document-level sentiment analysis or sentence-level sentiment analysis since they judge the sentiment polarity of a document or a sentence as a whole. For example, in Table 2.1, a document-level or sentence-level sentiment analysis system outputs whether the overall text presents a positive sentiment or a negative sentiment.

Instead of giving an overall sentiment label of a document or a sentence, many works focus on recognizing the specific components of each individual opinion, namely the sources, the opinion expressions, the targets, and the polarities (Johansson and Moschitti, 2013a; Yang and Cardie, 2013a). In their works, it is very important that the source is not limited to the writer, and that the source can be a noun phrase in the text. We call these works

phrase-level since the sources, the targets, and the opinion expressions are annotated as phrases in the commonly used resources. The phrase-level works usually analyze news, editorials, etc since the opinions in those genres are not always the writer, and the targets are much more general than the aspects of a certain product. In Table 2.1, a phrase-level system is expected to recognize three text spans, where the source is the writer, the opinion expression is *great* and it is a positive sentiment, the target is the phrase *Hillary Clinton defeated Trump*. However, the target span contains more than one entities and the source (i.e., the writer) has opposite sentiments toward the two entities. This indicates that we need a more fine-grained sentiment analysis system than a phrase-level system.

Our task is different from the works above in that our goal aims at recognizing the opinions expressed by the writer or an entity and toward an entity or event in the sentence. Our task is defined as entity/event-level sentiment analysis. This is a more fine-grained task compared to the document-level, sentence-level, and phrase-level. Note that our task is different from the phrase-level task because a phrase annotated as a target span may contain more than one entity or event (Deng and Wiebe, 2015b). Moreover, we do not organize the targets as aspects since we focus on the news and editorials genre. The features of a certain product such as hotel or computer are much fewer than the features of a news event.

Note that in the review genres, there are many works specifying the target of a sentiment and find relations of the targets. In their works, the sources are usually assumed to be the writers of the reviews, but the targets are phrases that represent the "features" of the product. For example, in a hotel review, the sentences "*The hotel is clean*" and "*The room is clean*" contain opinions whose targets are *The hotel* and *The room*. The two targets refer to the same feature of the hotel, *cleanness*. Their goal aims at recognizing the writer's sentiment toward the features of the hotels. The feature of a product is denoted as "aspect" and such works are called as aspect-based sentiment analysis (Hu and Liu, 2004; Titov and McDonald, 2008; Liu, 2012). In recent years the aspect-based sentiment analysis (ABSA) was first introduced as a task in the SemEval 2014 and has been hosted as an independent task until the current year (Pontiki and Manandhar, 2014; Pontiki et al., 2015, 2016). Most of the aspect-based systems are trained to find the aspects of certain products such as hotels, laptops, cameras, etc which have a pre-defined set of possible aspects. However, there is no

pre-defined set of aspects about a defeating event in Table 2.1, and it is very difficult to abstract an aspect of a general news event. Our task differs from the aspect-based sentiment analysis in that we do not recognize the relations of opinion targets. This is because we focus on the news and editorial genres where the targets are much more various than the targets in the review genre. However, the works including co-reference resolution, event resolution, and entity linking can be used to find whether two targets refer to the same concept. But that goal is out of the scope of this thesis.

2.2 WORKS ON SENTIMENT INFERENCE

Most works mentioned above in sentiment analysis focuses on recognizing explicit sentiments and extracting explicit opinion expressions, sources, and targets. There are some works investigating features that directly indicate implicit sentiments (Zhang and Liu, 2011; Feng et al., 2013). However, identifying terms that imply opinions is a different task than sentiment inference between entities. Dasigi et al. (2012) search for implicit sentiments shared between authors, while we address inferences within a single text. Several papers apply compositional semantics to determine polarity (Moilanen and Pulman, 2007a; Choi and Cardie, 2008; Moilanen et al., 2010, etc.). The goal of such works is to determine one overall polarity of an expression or sentence. In contrast, this thesis commits to a source having sentiments toward various events and entities in the sentence, possibly of different polarities.

Specifically, in this thesis, we work on +/-effect events to recognize more sentiments. The idea of +/-effect events in sentiment analysis is not entirely new. For example, two papers mentioned above (Zhang and Liu, 2011; Choi and Cardie, 2008) include linguistic patterns for the tasks that they address that include +/-effect events, but they don't define general inference rules relating sentiments and +/-effect events, agents, and objects as we do. Recently, in linguistics, Anand and Reschke (2010) identify classes of +/-effect event terms, and carry out studies involving artificially constructed +/-effect events and corpus examples matching fixed linguistic templates (Reschke and Anand, 2011). Our works focus on +/-effect event triples in naturally-occurring data and use generalized implicature rules.

Goyal et al. (2012) generate a lexicon of *patient polarity verbs*, which correspond to +/-effect events whose spans are verbs. Riloff et al. (2013a) investigate sarcasm where the writer holds a positive sentiment toward a negative situation. However, neither of these works performs sentiment inference. Previously, Wiebe and Deng (2014) also propose a set of sentiment inference rules and develop a rule-based system to infer sentiments. However, the rule-based system requires *all* information regarding explicit sentiments and +/-effect event information to be provided as oracle information by manual annotations.

In this thesis, we first develop a full set of inference rules instead of linguistic templates and develop automatic systems where the rules are used as soft constraints instead of hard constraints. We will evaluate the rules and the systems on real-world data instead of manually crafted examples.

2.3 WORKS ON JOINT PREDICTION MODELS AND APPLICATIONS

Different from pipeline architectures, where each step is computed independently, a joint prediction model has often achieved better results. Roth and Yih (2004) formulated the task of information extraction using Integer Linear Programming (ILP). Since then, ILP has been widely used in various tasks in NLP, including semantic role labeling (Punyakanok et al., 2004, 2008; Das et al., 2012), joint extraction of opinion entities and relations (Choi et al., 2006; Yang and Cardie, 2013a), co-reference resolution (Denis and Baldridge, 2007), and summarization (Martins and Smith, 2009). The most similar ILP model to ours is (Somasundaran and Wiebe, 2009), which improves opinion polarity classification using discourse constraints in an ILP model. However, their works address discourse relations among explicit opinions in different sentences. Probabilistic Soft Logic (PSL) is a new statistical relational learning method that has been applied to many NLP and other machine learning tasks in recent years (Beltagy et al., 2014; London et al., 2013; Pujara et al., 2013; Bach et al., 2013; Huang et al., 2012, 2013; Memory et al., 2012). Previously, PSL has not been applied to entity/event-level sentiment analysis.

2.4 SUMMARY

In this thesis, we devote to the entity/event-level sentiment analysis task which has not been formally defined and fully studied before. This is a much more fine-grained sentiment analysis task, compared to the tasks in the previous works. As shown by the examples in Chapter 1, the sentiments toward an entity may not be directly associated with a sentiment expression. Instead, it needs inference to recognize that opinion. Though previous works have used linguistic templates or lexicons to conduct sentiment inference, they did not propose specific rules to conduct sentiment inference. Different from previous works, this thesis presents the inference rules in propositional logics and develop computational models to use the rules as soft constraints in order to recognize the sentiments toward entities and events in the sentences. As a downstream application, the sentiment analysis tasks needs input from several individual NLP tasks including semantic role labeling, word sense disambiguation, etc. Instead of building a pipeline system which selects the answer to each individual task separately, this thesis develops joint prediction models to solve each individual task after considering all the answers of all the tasks. Furthermore, this thesis pioneers in building dependencies between the sentiment analysis task and other NLP tasks to improve sentiment analysis.

3.0 REPRESENTATIONS OF ENTITY/EVENT-LEVEL SENTIMENT

Before we talk about how to conduct sentiment inferences, we first give the representations of sentiments. Previously the representations of sentiments in various works of sentiment analysis vary a lot. For example, in document-level or sentence-level sentiment analysis works, the representations are sentiment labels including positive, negative or neutral. Later, for more fine-grained sentiment analysis works, more specific representations are developed to address the targeted goals of the task. For example, most works on aspect-based sentiment analysis focus on recognizing the sentiment toward the aspect (i.e., feature) of a certain product. Different text spans may refer to the same aspect. (In a review, two sentences, (1) The hotel is clean and (2) The room is clean, are opinions expressed toward the same aspect of the hotel, cleanness.) The source is assumed to be the writer. Thus, their representations are sentiment labels toward various aspects where each "abstract" aspect may correspond to more than one specific text span. The works on phrase-level sentiment analysis focus on recognizing the text spans of sources and targets. The text spans are usually phrases. The representations are triples of spans, each of which is ⟨source span, opinion expression span, target span⟩. The representations listed above do not apply to our task. First, the document-level or sentence-level sentiment labels are too coarse-grained. Second, the focus of this thesis is to recognize sentiments expressed or indicated in the sentences, instead of organizing the recognized sentiments as knowledge bases. The aspect-based sentiment aims at the latter goal in the review genre. Third, as defined in Chapter 1, only explicit sentiments are associated with opinion expressions, while implicit sentiments do not have corresponding opinion expressions. Thus, in this thesis we develop our representations of entity/event-level sentiment analysis. Though it is different from previous representations, the new representation aims at representing the sentiments in sentences. What's more important, the

representation is able to represent the sentiments defined in the previous tasks since we also always specify different sources and targets in different tasks using this representation.

In this chapter, we will introduce our representations for entity/event-level sentiment analysis in this thesis. The representations include entity/event-level sentiment representations, which will be introduced in Section 3.1. Not limited to sentiments, the representations also include entity and event representations, especially about a particular type of event: +/-effect event. We particularly define representations of this type of event is because many implicit sentiments are expressed the +/-effect events. As will be shown later in the rest of this thesis, modelling the +/-effect events is very important in sentiment inference. The representation of +/-effect events will be introduced in Section 3.2.

3.1 REPRESENTATION OF SENTIMENTS

Though the representation of sentiments differs as the granularity of sentiments differs, generally speaking, a sentiment is defined to have three components. The **source** of a sentiment is someone who holds the opinion. The **target** of a sentiment is toward whom or which the sentiment is expressed. The **polarity** of the sentiment is positive, negative or neutral. The three components of opinions are the same across different levels of granularities. In this section, we define our representations and terminologies to represent the granularity of the task in this thesis, the entity/event-level sentiment analysis. The representations of entity/event-level sentiments will be given by going through examples.

Let's begin with the example sentence below.

(Ex3.1) The reform will lower the ridiculous healthcare cost.

There is a negative sentiment, revealed by an opinion expression (ridiculous). We define -SENTIMENT(m) to represent this negative sentiment, where m refers to the phrase ridiculous. Similarly, we define +SENTIMENT(m) to represent a positive sentiment, where m refers to a positive opinion expression. Each sentiment has a corresponding source (i.e., holder), who the sentiment is attributed to. In (Ex3.1), the source is the writer of the sentence. We

define SOURCE(m, s) to represent that the entity s is the source of the sentiment m, where in (Ex3.1) s is the writer. Each sentiment has a corresponding target, whom or which the sentiment is expressed toward. In (Ex3.1), the target is the healthcare cost. We define TARGET(m, t) to represent that the entity or event t is the target of the sentiment m. In summary, the negative sentiment can be represented as

-SENTIMENT(ridiculous) $\wedge$ SOURCE(ridiculous, writer) $\wedge$ TARGET(ridiculous, cost).

There is also a positive sentiment in the sentence. The writer is positive toward the reform since the writer believes the reform could lower the cost which the writer does not like. However, there is no opinion expression describing the reform. In other words, the positive sentiment does not have a corresponding opinion expression, even though it is indicated in the sentence. In order to represent this positive sentiment, we define POSITIVEPAIR(s, t) to represent this positive sentiment, representing that the entity s is positive toward the entity or event t. In (Ex3.1), s refers to the writer and t refers to the reform. Similarly, we define NEGATIVEPAIR(s, t) to represent a negative sentiment. Note that, the negative sentiment revealed by the word ridiculous can also be represented as NEGATIVEPAIR(writer, cost).

In this thesis, a sentiment such as the negative sentiment in (Ex3.1) is defined as an **explicit sentiment**. An explicit sentiment is associated with an opinion expression. It can be represented as +SENTIMENT(m) or -SENTIMENT(m). A sentiment such as the positive sentiment in (Ex3.1) is defined as an **implicit sentiment**. An implicit sentiment is not associated with an opinion expression but it is indicated in the text. Both explicit and implicit sentiment can be represented as POSITIVEPAIR(s, t) or NEGATIVEPAIR(s, t). In the example above, the negative sentiment revealed by the opinion expression can be represented in two ways, either by -SENTIMENT(ridiculous) or by NEGATIVEPAIR(writer, cost). We also define two rules to build connections from the different representations of explicit sentiment, listed below.

+SENTIMENT(y) $\wedge$ SOURCE(y,s) $\wedge$ TARGET(y,t) $\Rightarrow$ POSITIVEPAIR(s,t)
-SENTIMENT(y) $\wedge$ SOURCE(y,s) $\wedge$ TARGET(y,t) $\Rightarrow$ NEGATIVEPAIR(s,t)

By the two rules above, we can see that the representations on the left hand side of the rules are coherent to the definitions of opinions and they are applicable to different

granularities of explicit sentiments. The representations on the right hand side of the rules are very suited for the new task in this thesis, the entity/event-level sentiment analysis. The s and t in the literals refer to entities or events, and the positive and negative pairs show that the focus of the entity/event-level sentiment analysis task aims at ultimately identifying and aggregating opinions expressed among people and things in the sentences without rigidly adhering to concrete text spans **directly** linking from the source to the target.

A summary of entity/event-level sentiment representations is presented in Table 3.1.

+SENTIMENT(m)	m is a positive opinion expression
-SENTIMENT(m)	m is a negative opinion expression
SOURCE(m, s)	s is the source of the sentiment m
TARGET(m, t)	t is the target of the sentiment m
POSITIVEPAIR(s, t)	s is positive toward t
NEGATIVEPAIR(s, t)	s is negative toward t

Table 3.1: Representations of entity/event-level sentiment

3.2 REPRESENTATION OF +/-EFFECT EVENTS

Recall (Ex3.1) in the previous section. The reason why we can perceive the positive sentiment toward the reform is because the writer believes the reform is reducing the cost, and lowering the cost is an event which is bad for the cost. In this section, we give the definitions of events such as lower which is bad for the theme of it in Section 3.2.1, and we give the representations of this type of event in Section 3.2.2.

3.2.1 Definition of +/-effect event

A +effect event is defined as an event which has positive effect on the theme of the event, including creation events (as in *bake a cake*, which creates the cake and subsequently has

positive effect on the cake), gaining events (as in *increasing costs*, which contributes to the costs), and benefiting events (as in *comforted the child*, which is good for the child), etc (Deng et al., 2013; Anand and Reschke, 2010). A -effect event is defined as an event which has negative effect on the theme of the event, including destroying events (as in *tear down the building*, which is bad for the building), decreasing events (as in *decreasing the costs*, which leads to loss of the costs, and harming events (as in *infuriating him*, which is harmful to him), etc (Deng et al., 2013; Anand and Reschke, 2010). A +/-effect event has four components: the event itself, including the effect of the event; the agent who conducts the event; the theme whom the event affects; and the influencer which retains or reverses the effect of the event. We will introduce each component below.

3.2.1.1 +/-Effect Event. As introduced above, a +/-effect event either benefits or harms the theme. Here we target clear cases of +/-effect events. The event must be representable as a triple of contiguous text spans, ⟨agent,event,theme⟩. The effect of the event should be perceptible by looking only at the spans in the triple. If for example, another argument of the event is needed to perceive the relationship, that event is out of scope of this study. Consider the examples below.

(Ex3.2) His uncle **left** him ***a massive amount of debt***.
(Ex3.3) His uncle **left** him ***a treasure***.

There is no way to break these sentences into triples that follow our definitions. ⟨His uncle, left, him⟩ doesn't work because we cannot perceive the effect of the event by looking only at the triple; whether it is +effect or -effect depends on what his uncle left him. ⟨His uncle, left him, a massive amount of debt⟩ isn't correct: the event is not bad for the debt, it is bad for *him*. Finally, ⟨His uncle, left him a massive amount of debt, Null⟩ isn't correct, since no theme is identified. Thus, we do not define or annotate the cases such as (Ex3.2) and (Ex3.3) as +/-effect events in this thesis.

An event may be beneficial or harmful to the agent (Choi and Wiebe, 2014). However, in this thesis we only focus on events which affect the theme.

3.2.1.2 Agent and Theme. The **agent** of a +/-effect event is someone who conducts the event. It is often a noun phrase, but sometimes it can be *implicit* (as in *the constituent will be destroyed*, the agent of the destroying event is implicit). The **theme** of a +/-effect event is someone or something that the event effects. It is always a noun phrase. In our definition, the +/-effect event always has a theme.

3.2.1.3 Influencer. Another component of a +/-effect event is the **influencer**, a word whose effect is to either retain or reverse the effect of a +/-effect event. Let's see the examples below.

(Ex3.4) Luckily Bill *didn't* **kill** him.
(Ex3.5) The reform *prevented* companies from **hurting** patients.
(Ex3.6) John *helped* Mary to **save** Bill.

In (Ex3.4) and (Ex3.5), the words, didn't and prevented, respectively, reverse the effect of the event from -effect to +effect (killing itself is bad for Bill, but not killing Bill is good for Bill; hurting patients is bad for the patients, but preventing companies from hurting patients is good for the patients). We define the phrase reversing the effect as the **reverser**, and the phrase retaining the effect as the **retainer**. In (Ex3.6), the word helped is an influencer which retains the +effect (i.e., helping Mary to save Bill is good for Bill, as saving Bill is good for Bill).

An influencer has an agent and a theme. Similar to the agent of the +/-effect event, the agent of an influencer is a noun phrase or *implicit*. Examples (Ex3.5) and (Ex3.6) illustrate the case where an influencer introduces an additional agent that is different from the agent of the +/-effect event the (the reform in (Ex3.5) and John in (Ex3.6)). The theme of an influencer must be another influencer or a +/-effect event.

Note that, semantically, an influencer can be seen as a +/-effect event for its theme. A reverser influencer makes its theme irrealis (i.e., not happen). Thus, it has negative effect on it. In (Ex3.5), for example, *prevent* is bad for the *hurting* event. A retainer influencer maintains its theme, and thus has positive effect on it. In (Ex3.6), for example, *helped* maintains the *saving* event.

3.2.2 Representation of +/-Effect Events

We define +EFFECT(m) to represent a +effect event, and -EFFECT(m) to represent a -effect event. Though a +/-effect event may have an influencer, we do not define the influencers in our representations. As discussed above in Section 3.2.1 an influencer can be treated as a +/-effect event. It is not necessary to define a representation of influencer. Instead, we generalize both +/-effect evens and those influencers that are semantically equivalent to +/-effect events using one definition: +EFFECT(m) and -EFFECT(m). For example, in (Ex3.5), it should be represented as -EFFECT(hurting) and -EFFECT(prevent). We define AGENT(m, a) to represent a is the agent who conducts the event, and THEME(m, h) to represent h is the theme which or whom the event affects.

The representations of +/-effect event information are summarized in Table 3.2.

+EFFECT(m)	m is a +effect event
-EFFECT(m)	m is a -effect event
AGENT(m, a)	a is the agent of m
THEME(m, h)	h is the theme of m

Table 3.2: Representations of +/-effect event

3.3 RELATED WORK

The representation of a sentiment task depends on the goal of the task. The works in analyzing a document's overall polarity of a sentence's overall polarity seldom develop a particular presentation suited for the task since the problem can be modelled as a classification problem (Pang et al., 2002; Turney, 2002; Yu and Hatzivassiloglou, 2003; McDonald et al., 2007). In their works, the source is assumed to be the writer, and the target is not specified. If it is a movie review document, then the target is the movie and the polarity of the document represents the writer's opinion about the movie. We call these works as document-level sen-

timent analysis or sentence-level sentiment analysis since they classify the sentiment polarity of a document or a sentence as a whole.

Later as researchers sought a deeper understanding of the sentiments expressed in the text, components of an opinion were defined. Wilson (2008) defined an opinion to have a source and a target. The source is whom the opinion is attributed to, and the target is toward whom or what the opinion is expressed. It is also annotated in the subjectivity corpus such as MPQA (Wiebe et al., 2005) that each opinion has an opinion expression, the phrase which reveals the sentiment. Thus, the representation of a sentiment in a task can be defined in various ways depending on the goal of the task. The representation of a sentiment can be a pair, including the opinion expression and the polarity. The works using the pair representation aim at recognizing the opinion expressions, which are words and phrases, and the corresponding polarity, which includes positive, negative or neutral (Yang and Cardie, 2014). The representation of a sentiment can also be a triple, including the opinion source, the opinion expression, and the opinion target. The works using the triple representation aim at recognizing three phrases corresponding to each component in the triple set (Yang and Cardie, 2013a). Of course, the representation of a sentiment can also be a four tuple, including all the components of a sentiment. The works using the quad set representation usually combine the pair representation and the triple representation (Johansson and Moschitti, 2013b). We call these works as phrase-level sentiment analysis since the sources, the targets, and the opinion expressions are annotated as phrases in the commonly used resources.

The targets of opinions annotated in the fine-grained sentiment corpus are phrases, each of which is an individual phrase and independent from the others. Many works in sentiment analysis try to find relations of the targets. Such works focus on the review genre and analyze the product reviews. In the works, the sources are usually assumed to be the writers of the reviews, but the targets are phrases that represent the "features" of the product. For example, in a hotel review, two sentences, (1) The hotel is clean and (2) The room is clean, contain opinions whose targets are the hotel and the room. The two targets refer to the same feature of the hotel, cleanness. Their goal aims at recognizing the writer's sentiment toward the features of the hotels. Thus, the representation of their works includes the opinion

expression phrase, the polarity and the feature that the opinion comments. For example, the first sentence mentioned above (The hotel is clean) is expected to be annotated with ⟨clean, positive, cleanness⟩. The feature of a product is denoted as "aspect" and such works are named as aspect-based sentiment analysis (Liu, 2012). In recent years aspect-based sentiment analysis (ABSA) was first introduced as a task in the SemEval 2014 and has been hosted as an independent task until current year (Pontiki and Manandhar, 2014; Pontiki et al., 2015, 2016).

Our task is different from the works above in that our goal aims at recognizing the opinions expressed by the writer or an entity and toward an entity or event in the sentence. Our task is named as entity/event-level sentiment analysis. This is a more fine-grained task compared to the document-level, sentence-level and phrase-level. Note that our task is different from the phrase-level task because a phrase annotated as a target span may contain more than one entities or event (Deng and Wiebe, 2015b). Moreover, we do not organize the targets as aspects, since we focus on the news and editorials genre. The features of a certain product such as hotel or computer are much less than the features of a news event.

3.4 SUMMARY

In this chapter, we have introduced the representation of sentiments for our task. The new representations are able to represent the components of explicit sentiments and the components of implicit sentiments as well. Previously, there have been few works that focus on recognizing implicit sentiments, and that give a formal definition of implicit sentiments. The distinction between explicit sentiments and implicit sentiments has not been discussed. The representation introduced in Section 3.1 includes two sets of representations. One set can represent both explicit and implicit sentiments, and the other set only represents the explicit sentiments. From the representation, it is clearly shown that the difference between explicit sentiments and implicit sentiments is whether a sentiment expression is directly present in the sentence. Further, since there are two sets of representations defined for the explicit sentiments, we define rules mapping from one set to the other so that the two sets are not

isolated from each other. Based on our observation, many implicit sentiments are triggered by a particular event, the +/-effect events. We give the definition of +/-effect events and develop the representation of it in Section 3.2. In the next chapter, we will talk more about the +/-effect events as we develop the corpus annotated with that information. Further, in the later chapters, we will see that the +/-effect information contributes to improving entity/event-level sentiment analysis.

4.0 CORPORA OF ENTITY/EVENT-LEVEL SENTIMENT

Since previous corpora are mainly annotated with explicit sentiments, we have developed the corpora annotated with both explicit and implicit sentiments, which can serve as resources of entity/event-level sentiment analysis.

Developing a corpus from scratch is not an easy task because it requires a very detailed manual and specially trained annotators. At the beginning of annotating entity/event-level sentiments, we try to focus on a small but clear subtask. We annotate the writer's sentiments toward the agents and themes of +/-effect events in the text. Compared to any entity's sentiments in the text, the writer's sentiments are more intuitive to detect and easy to guide the annotators. Annotating this type of sentiments is a clear annotation task. Further, as shown in the examples in the previous chapters, +/-effect events are good indicators of implicit sentiments. Thus, annotating writer's sentiments toward agents and themes of +/-effect events is a good start point for the whole task. To annotate such sentiments, we need to annotate the +/-effect event spans in the sentences as well. The annotations of +/-effect event spans have contributed to the study of +/-effect lexicon development and the +effect or -effect word sense disambiguation by other researchers (Choi and Wiebe, 2014, etc). This corpus is referred as **+/-Effect Event Sentiment Corpus** later in the thesis. The development of this corpus will be introduced in Section 4.1.

Though the annotations of the +/-Effect Event Corpus are limited and do not cover all the cases of the ultimate task, the annotations still provide resources for us to study the simplified subtask as a pilot study of the whole task. The good agreement study provides evidence that it is feasible to develop an annotation scheme to guide the annotators to annotate both explicit and implicit sentiments expressed toward the agents and themes of +/-effect events. Next, we try to fully utilize the existing resources of annotated sentiment

corpora.

Then we move on to annotating the sentiments expressed from any entity toward another entity or event. Compared to the first one, the second corpus is annotated with various entity/event-level sentiments, where the sources are not only the writer but can be the writer or any entity in the text, and the targets are not limited to the agent and themes of +/-effect events but can be any entity or event in the text. The second corpus is more complicated and more difficult in annotation than the first one. In order to both reduce the annotation effort and utilize existing resources, we choose to build upon annotations in the selected previous corpus. Previously annotated corpus is developed to serve a different task from ours, the annotations provide sound resources of sentiments defined in the previous task. Building upon the previous annotations largely reduces the workload of annotating resources for our task. Since our goal is detecting entity/event-level sentiment, the most similar corpus to ours is a phrase-level sentiment annotated corpus, where each source and target is a phrase which may contain more than one entity or event. To develop the entity/event-level sentiment resource, we choose to build upon MPQA 2.0 (Wiebe et al., 2005; Wilson, 2008) and add entity/event-level annotations onto the phrase-level annotations. As an extension to MPQA 2.0, the second corpus is referred as **MPQA 3.0**. The development of the second corpus will be introduced in Section 4.2.

We will give the annotation scheme, the agreement study design and the result, the corpus statistics and several examples when introducing each corpus. The new corpora promise to be valuable new resources for developing systems for entity/event-level sentiment analysis.

4.1 +/-EFFECT EVENT SENTIMENT CORPUS

In this section, we present a focused study of annotating +/-effect events and the sentiments expressed toward the entities participating in the +/-effect events. The annotated corpus will support further research on how implicit opinions are expressed via +/-effect events. Based on the definitions of +/-effect events in Section 3.2, we develop the annotation scheme for annotating sentiments expressed toward the agents and themes of +/-effect events (Section

4.1.1). Then we will carry out the agreement study which achieves a good result (Section 4.1.2). We also analyze the disagreement of annotations as a follow up to the agreement study. Finally, we will present the statistics of the annotated corpus and give some examples in the corpus (Section 4.1.3).

4.1.1 Annotation Scheme

There are four types of annotations: +/-effect event, influencer, agent, and theme. To annotate a +/-effect event, it is required to annotate the text span of the event and how it affects the theme (+effect or -effect). To annotate an influencer, it is required to annotate the text span of the influencer and how it affects the theme of it (retain or reverse). To annotate an agent, it is required to annotate the text span of the agent and the writer's sentiment toward it (positive, negative or none). To annotate a theme, it is required to annotate the text span of the theme and the writer's sentiment toward it (positive, negative or none). Each annotation is assigned a unique ID in the document. It is also required to link agents and themes to their +/-effect events or influencer annotations via explicit IDs. When an agent is not mentioned explicitly, the annotator should indicate that it is *implicit*. For any span the annotator is not certain about, he or she can set the *uncertain* option to be true.

For example:

(Ex4.1) GOP **Attack on** Reform Is a **Fight Against** Justice.
(Ex4.2) **Jettison** any reference to end-of-life counselling.

In (Ex4.1), the annotators are expected to identify two -effect events: ⟨GOP, Attack on, Reform⟩ and ⟨GOP Attack on Reform, Fight Against, Justice⟩. The effects of both events are -effect. The writer's sentiments toward both agents are negative, and the writer's sentiments toward both themes are positive. In (Ex4.2), the annotator is expected to identify the -effect event: ⟨implicit, Jettison, any reference to end-of-life counselling⟩. The writer conveys a negative attitude toward the end-of-life counselling. The annotation manual instructs the annotators to consider whether an attitude of the writer is communicated or revealed in the particular sentence which contains the +/-effect event.

Though it often is, the +/-effect event span need not be a verb or verb phrase. We saw an example above, namely (Ex4.1). Even though the events (attack on and fight against) are not verbs, we still mark them because they represent events that have negative effect on the theme.

Turning to influencers, there may be chains of them, where the ultimate effect and agent must be determined compositionally. For example, the structure of *Jack stopped Mary from trying to kill Bill* is a reverser influencer (stopped) whose theme is a retainer influencer (trying) whose theme is, in turn, a -effect event (kill). The ultimate effect of this event is +effect and the "highest level" agent is Jack. In our scheme, all such chains of length N are treated as $N-1$ influencers followed by a single +/-effect event. It will be up to an automatic system to calculate the ultimate effect and agent using rules such as those presented in (Moilanen and Pulman, 2007b; Neviarouskaya et al., 2010, etc).

To save some effort, the annotators are not asked to mark retainer influencers which do not introduce new agents. For example, for *Jack stopped trying to kill Bill*, there is no need to mark the word trying. Of course, all reverser influencers must be marked.

4.1.2 Agreement Study

To validate the reliability of the annotation scheme, we conduct an agreement study. In this section we introduce how we design the agreement study, present the evaluation method and give the agreement results. Additionally, we conduct a second-step consensus study to further analyze the disagreement.

4.1.2.1 Data and Agreement Study Design For this study, we want to use data that is rich in opinions and implicatures. Thus we used the corpus from (Conrad et al., 2012), which consists of 134 documents from blogs and editorials about a controversial topic, "the Affordable Care Act".

To measure agreement on various aspects of the annotation scheme, two annotators participate in the agreement study; one of the two isn't involved in developing the scheme. The new annotator first reads the annotation manual and discusses it with the first annotator.

Then, the annotators label 6 documents and discuss their disagreements to reconcile their differences. For the formal agreement study, we randomly select 15 documents, which have a total of 725 sentences. These documents do not contain any examples in the manual, and they are different from the documents discussed during training. The annotators then independently annotate the 15 selected documents.

4.1.2.2 Agreement Study Evaluation We annotate four types of items (+/-effect event, influencer, agent, and theme) and their corresponding labels. As noted previously in Section 3.2, influencers can also be viewed as +/-effect events. Also, the two may be combined together in chains. Thus, we measure agreement for +/-effect and influencer spans together, treating them as one type. Then we choose the subset of +/-effect and influencer annotations that both annotators identified, and measure agreement on the corresponding agents and themes.

Sometimes the annotations differ even though the annotators recognize the same +/-effect event. Consider the following sentence:

(Ex4.3) Obama **helped** reform **curb** costs.

Suppose the annotations given by the annotators were:

Ann 1. ⟨Obama, helped, curb⟩
⟨reform, curb, costs⟩
Ann 2. ⟨Obama, helped, reform⟩

The two annotators do agree on the ⟨Obama, helped, reform⟩ triple, the first one marking the word helped as a retainer and the other marking it as a +effect event. To take such cases into consideration in our evaluation of agreement, if two spans overlap and one is marked as +/-effect event and the other as influencer, we use the following rules to match up their agents and themes:

- for a +/-effect event, consider its agent and theme as annotated;
- for an influencer, assign the agent of the influencer's theme to be the influencer's theme, and consider its agent as annotated and the newly-assigned theme. In (Ex4,3), Ann 2's

annotations remain the same and Ann 1's become ⟨Obama, helped, reform⟩ and ⟨reform, curb, costs⟩.

We use the same measurement for agreement for all types of spans. Suppose A is a set of annotations of a particular type and B is the set of annotations of the same type from the other annotator. For any text span $a \in A$ and $b \in B$, the span coverage c measures the overlap between a and b. Two measures of c are adopted here.

Binary: As in (Wilson and Wiebe, 2003), if two spans a and b overlap, the pair is counted as 1, otherwise 0.

$$c_1(a, b) = 1 \quad if \quad |a \cap b| > 0$$

Numerical: Johansson and Moschitti (2013b) propose, for the pairs that are counted as 1 by c_1, a measure of the percentage of overlapping tokens,

$$c_2(a, b) = \frac{|a \cap b|}{|b|}$$

where $|a|$ is the number of tokens in span a, and $\cap$ gives the tokens that two spans have in common. As Breck et al. (2007) point out, c_2 avoids the problem of c_1, namely that c_1 does not penalize a span covering the whole sentence, so it potentially inflates the results.

Following (Wilson and Wiebe, 2003), treating each set A and B in turn as the gold-standard, we calculate the average F-measure, denoted $agr(A, B)$. $agr(A, B)$ is calculated twice, once with $c = c_1$ and once with $c = c_2$.

$$match(A, B) = \sum_{\substack{a \in A, b \in B, \\ |a \cap b| > 0}} c(a, b)$$

$$agr(A||B) = \frac{match(A, B)}{|B|}$$

$$agr(A, B) = \frac{agr(A||B) + agr(B||A)}{2}$$

Now that we have the sets of annotations on which the annotators agree, we use κ (Artstein and Poesio, 2008) to measure agreement for the labels. We report two κ values: one for the effects of the +/-effect events, together with the effects of the influencers, and one for the writer's sentiment toward the agents and themes. Note that, as in Example (9),

		+/-effect & influencer	agent	theme
all annotations	c_1	0.70	0.92	1.00
	c_2	0.69	0.87	0.97
only certain	c_1	0.75	0.92	1.00
	c_2	0.72	0.87	0.98
consensus study	c_1	0.85	0.93	0.99
	c_2	0.81	0.88	0.98

Table 4.1: Span overlapping agreement $agr(A, B)$ in agreement study and consensus study.

sometimes one annotator marks a span as +/-effect and the other marks it as an influencer; in such cases we regard the labels, retain and +effect, as the same label and the labels, reverse and -effect, as the same label. Table 4.1.2.3 gives the *agr* values and Table 4.2 gives the κ values.

4.1.2.3 Agreement Study Results Recall that the annotator could choose whether (s)he is certain about the annotation. Thus, we evaluate two sets: all annotations and only those annotations that both annotators are certain about. The results are shown in the top four rows in Table 4.1.2.3 .

The results for agents and themes in Table 4.1.2.3 are all quite good, indicating that,

	polarity & effect	attitude
all	0.97	0.89
certain	0.97	0.89

Table 4.2: κ for label agreement.

given a +/-effect or influencer, the annotators are able to correctly identify the agent and theme.

Table 4.1.2.3 also shows that results are not significantly worse when measured using c_2 rather than c_1. This suggests that, in general, the annotators have a good agreement concerning the boundaries of spans.

Table 4.2 shows that the κ values are high for both sets of labels.

4.1.2.4 Consensus Analysis Following (Medlock and Briscoe, 2007), we examine what percentage of disagreement is due to negligence on behalf of one or the other annotator (i.e., cases of clear +/-effects events or influencers that were missed), though we conduct our consensus study in a more independent manner than face-to-face discussion between the annotators. For annotator $Ann1$, we highlight sentences for which only $Ann2$ marked a +/-effect event, and give $Ann1$'s annotations back to him or her with the highlights added on top. For $Ann2$ we do the same thing. The annotators reconsider their highlighted sentences, making any changes they feel they should, without communicating with each other. There could be more than one annotation in a highlighted sentence; the annotators are not told the specific number.

After re-annotating the highlighted sentences, we calculate the agreement score for all the annotations. As shown in the last two rows in Table 4.1.2.3, the agreement for +/-effect and influencer annotations increases quite a bit. Similar to the claim in (Medlock and Briscoe, 2007), it is reasonable to conclude that the actual agreement is approximately lower bounded by the initial values and upper bounded by the consensus values, though, compared to face-to-face consensus, we provide a tighter upper bound.

4.1.3 Statistics and Examples

As mentioned in Section 4.1.2 that we use the corpus from (Conrad et al., 2012), which consists of 134 documents with a total of 8,069 sentences from blogs and editorials about "the Affordable Care Act". There are 1,762 +/-effect and influencer annotations. On average, more than 20 percent of the sentences contain a +/-effect event or an influencer. Out of all

+/-effect and influencer annotations, 40 percent are annotated as +effect or retain and 60 percent are annotated as -effect or reverse. For agents and themes, 52 percent are annotated as positive and 47 percent as negative. Only 1 percent are annotated as none, showing that almost all the sentences (in this corpus of editorials and blogs) which contain +/-effect annotations are subjective. The annotated corpus is available online[1].

To illustrate various aspects of the annotation scheme, in this section we give several examples from the corpus. In the examples below, words in square brackets are agents or themes, words in italics are influencers, and words in boldface are +/-effect events.

1. And [it] will *enable* [Obama and the Democrats] - who run Washington - to get back to **creating** [jobs].

   (a) The creating event has +effect on the theme, the jobs; the agent is Obama and the Democrats.

   (b) The phrase, to get back to, is a retainer influencer. But, the agent span is also Obama and the Democrats, as the same with the +effect event, so we don't have to give an annotation for it.

   (c) The word, enable, is a retainer influencer. Since its agent span is different (it), we do create an annotation for it.

2. [**Repealing** [the Affordable Care Act]] would **hurt** [families, businesses, and our economy].

   (a) Repealing is a -effect event since it deprives the theme, the Affordable Care Act, of its existence. In this case the agent is *implicit*.

   (b) The agent of the hurting event is the whole phrase, Repealing the Affordable Care Act. Note that the agent span is, in fact, a noun phrase (even though it refers to an event). Thus, it doesn't break the rule that all agent +/-effect spans should be noun phrases.

3. It is a moral obligation to *end* this indefensible **neglect of** [hard-working Americans].

   (a) This example illustrates a +/-effect event that is anchored to a noun (neglect) rather

[1] http://mpqa.cs.pitt.edu/

than to a verb.

(b) It also illustrates the case when two words can be seen as +/-effect events: both events, end and neglect of, can be seen as -effect events. Following our specification, they are annotated as a chain ending in a single +/-effect event: the ending event is an influencer that reverses the effect of the -effect event, neglect of.

4.2 MPQA 3.0: ENTITY/EVENT-LEVEL SENTIMENT CORPUS

An entity/event-level sentiment corpus consists of opinions expressed from entities toward entities or events. That is to say, both the source and the target of an annotated sentiment is an entity or an event. Different from review data, where the opinion targets include the product and the aspects of the product (Hu and Liu, 2004; Titov and McDonald, 2008), the opinion targets in the news genre are more various. Though current sentiment corpora in news genre are span based, they have already provided what opinions are expressed. Thus, we choose to add entity/event-target annotations into existing span based sentiment corpus.

In this section, we present the development of the new entity/event-level sentiment corpus, MPQA 3.0, based on the annotations in the MPQA 2.0 corpus (Wiebe et al., 2005; Wilson, 2008).[2] The MPQA opinion annotated corpus is entirely span-based, and contains no entity/event-target annotations. However, it provides an infrastructure for sentiment annotation that is not provided by other sentiment NLP corpora, and is much more varied in topic, genre, and publication source. In this section, we talk about the annotation scheme guiding the annotators to add the annotations required in MPQA 3.0 based on the annotations in the existing MPQA 2.0 (Section 4.2.1). We carry out an agreement study, which achieves good results and analyze the disagreement (Section 4.2.2). Finally we present several examples we have annotated in MPQA 3.0 (Section 4.2.3).

[2]Available at http://mpqa.cs.pitt.edu

4.2.1 Annotation Scheme: From MPQA 2.0 to MPQA 3.0

The MPQA 2.0 (Wiebe et al., 2005; Wilson, 2008) is a rich opinion resource. It includes editorials, reviews, news reports, and scripts of interviews from different news agencies, and covers a wide range of topics. To create MPQA 3.0, entity/event-target annotations are added to the MPQA 2.0 annotations.[3] An entity/event-target is an entity or an event. The MPQA annotations consist of **private states**, which are states of sources holding attitudes toward targets.

In the MPQA 2.0 annotations, the top-level annotations are **direct subjective** (*DS*) and **objective speech event** annotations. DS annotations are for private states, and objective speech event annotations are for objective statements attributed to a source. An important property of sources is that they are nested, reflecting the fact that private states and speech events are often embedded in one another.

As shown in Figure 4.1, one DS may contain links to multiple **attitude** annotations, meaning that all of the attitudes share the same nested source. The attitudes differ from one another in their attitude types, polarities, and/or targets. There are several types of attitudes included in MPQA 2.0 (Wilson, 2008; Stoyanov et al., 2005), including sentiment and arguing. This work focuses on sentiments, which are defined in (Wilson, 2008) as positive and negative evaluations, emotions, and judgements. Later in this section, we use the word **sentiment** to refer to the attitude annotations whose types are sentiments in MPQA 2.0. In the future, entity-targets or event-targets may be added to private states with other types of sentiments.

MPQA 2.0 also contains **expressive subjective element** (*ESE*) annotations, which pinpoint specific expressions used to express subjectivity (Wiebe et al., 2005). An ESE also has a nested-source annotation. Since we focus on sentiments, we only consider ESEs whose polarity is positive or negative (excluding those marked neutral).

The **span-target** annotations in MPQA 2.0 are linked to from the sentiments. The target annotations in MPQA 2.0 are usually spans that contain more than one word. What's more important, the span-targets in MPQA 2.0 may contain more than one entity or events,

[3] Available at http://mpqa.cs.pitt.edu

Figure 4.1: Structure in MPQA 3.0.

toward which the sentiments may be different. Later we will show this limitation with the real examples in MPQA 2.0. More than one span-target may be linked to from a sentiment, but most sentiments have only one span-target. The MPQA 2.0 annotators identified the main/most important target(s) they perceive in the sentence. If there is no target, the span-target annotation is *none*. However, there are many other targets to be identified. First, while ESE annotations have nested sources, they do not have any target annotations. Second, there are many more targets that may be marked than the major ones identified in MPQA 2.0. In Figure 4.1, the entity/event-targets are what we add in MPQA 3.0. We identify the blue (orange) entity/event-targets that are in the span of a blue (orange) span-target in MPQA 2.0. We also identify the green entity/event-targets that are not in the scope of any target.

Since our priority was to add entity/event-targets to sentiments, no entity/event-targets have yet been added to objective speech events, as shown in Figure 4.1.

To create MPQA 3.0, the corpus is first parsed, and potential entity/event-target annotations are automatically created from the heads of NPs and VPs. The annotators then consider each sentiment and each polar ESE, and decide for each which entity or event to add as the correct target of the sentiment. By adding such entity/event-targets to the existing annotations, the information in MPQA 2.0 is retained. Before presenting the scheme, we first give some examples.

4.2.1.1 Examples For each example, a subset of the annotations is shown. The phrase in blue is a sentiment span, the phrase in red is a target span, the tokens in yellow are the entity/event-targets which are newly annotated in MPQA 3.0. The underlined phrases are ESE spans. Each example is followed by the MPQA structure of the annotations.

In (Ex4.4), a negative sentiment is shown, which is issued the fatwa against. The source is the Imam. The target is the insulting event (Rushdie insulting the Prophet). However, the assertion that the Imam is negative toward the insult event is within the scope of this article. This is captured by an objective speech event annotation (not shown) whose target span includes the insult event, and whose source is the writer (w). In other articles, there may be opposite viewpoints. Strictly speaking, the writer of this article **thinks** that Imam is negative toward it. Thus, the complete interpretation of this negative sentiment is, according to the writer, the Imam is negative toward the insult event. In MPQA, the complete source is represented in a nested structure (i.e., nested-source) and the annotated nested-source is (writer, imam).

(Ex4.4) When the Imam issued the fatwa against Salman Rushdie for insulting the Prophet ...

DS: issued the fatwa
  nested-source: w, imam
  sentiment: issued the fatwa against
    sentiment-type: sentiment-negative
    span-target: Salman Rushdie for insulting the Prophet
      entity/event-target: Rushdie, insulting

We find two entity/event-targets in the span-target: Rushdie himself plus his act of insulting.

In the same sentence, there is another negative sentiment, insulting, as shown in (Ex4.5). The source is Salman Rushdie and the target is the Prophet. Note that the span covering this event is the target span of the sentiment in (Ex4.4): the private state of (Ex4.5) is nested in the private state of (Ex4.4). Thus, the complete interpretation of the negative sentiment in (Ex4.5) is: according to the writer, the Imam is negative toward Rushdie insulting the Prophet. The nested source is annotated as (w, Imam, Rushdie).

(Ex4.5) When the Imam issued the fatwa against Salman Rushdie for insulting the Prophet

...

DS: insulting
nested-source: w, imam, rushdie
sentiment: insulting
sentiment-type: sentiment-negative
span-target: the Prophet
entity/event-target: Prophet

We add an entity/event-target for the Prophet, anchored to the head Prophet. Interestingly, Prophet is an entity/event-target for the nested source (w,Iman,Rushdie) (i.e., Rushdie is negative toward the Prophet), but not for the nested source (w,Imam) (i.e., the Imam is not negative toward the Prophet). This shows that different entities (different sources) have different sentiments toward the same entity (same target).

In the following example, the target span is short.

(Ex4.6) He is therefore planning to trigger wars ...

DS: (entire sentence)
nested-source: w
sentiment: planning to trigger wars
sentiment-type: sentiment-negative
span-target: He
entity/event-target: He
entity/event-target: planning, trigger, wars

He is George W. Bush; this article appeared in the early 2000s. The writer is negative toward Bush because (the writer claims) he is planning to trigger wars. As shown in the example, the MPQA 2.0 target span is only the word, He, for which we do create an entity/event-target. But there are three additional entity/event-targets, which are not included in the target span. The writer is negative toward Bush planning to trigger wars; we make sense of this by inferring that the writer is negative toward the idea of triggering wars and thus toward war itself. As this example illustrates, all entities and events toward which the sentiment holds should be entity/event-targets.

(Ex4.7) Three leading international organisations warned jointly Thursday that the international fight against terrorism should not be a pretext for the violation of human rights.

DS: warned
nested-source: w, threeint

sentiment: warned
sentiment-type: sentiment-negative
span-target: the international fight against terrorism should not be a pretext for the violation of human rights
entity/event-target:be, pretext, violation
ESE: pretext
nested-source: w, threeint
polarity: negative
entity/event-target: pretext

The viewpoints in the article of (Ex4.7) are not against fighting terrorism (another sentence begins "*While we recognize that the threat of terrorism requires specific measures ...*") but against doing so in certain ways. Here the three organizations are against the fight being used as a pretext for civil rights violations. Thus, three heads, be, pretext, and violation, are entity/event-targets, but the other two heads, fight and terrorism are not. We mark the head, be, as an entity/event-target because the source is negative toward the **state** of the fight being a pretext for the violation of human rights. This makes sense with the source also being negative toward two heads, pretext, and violation. The fact that pretext is identified as a negative ESE annotation in the MPQA 2.0 supports this as well.

There is a difference between ESE and sentiment entity/event-target annotations. Since ESE annotations pinpoint specific wording used to express subjectivity, ESE entity/event-targets are annotated more narrowly than sentiment entity/event-targets. For ESEs, the entity/event-targets are the entities and events that are directly evaluated by the expression, while, for sentiments, the entity/event-targets include all entities and events toward which the sentiment holds (as we saw in the examples above). For example:

(Ex4.8) ... because the hard-line wing in the US administration comprising Vice President Dick Cheney ...

DS: (entire sentence)
nested-source: w
sentiment: hard-line
sentiment-type: sentiment-negative
target: wing in the US administration comprising Vice President Dick Cheney
entity/event-target: wing, Cheney
ESE: hard-line wing
nested-source: w
polarity: negative

entity/event-target: wing

The ESE has only one entity/event-target (wing) while the sentiment has two entity/event targets (wing and Cheney).

4.2.1.2 MPQA 3.0 Annotation Scheme

An **entity/event-target** is an entity or event that is the target of a sentiment (identified in MPQA 2.0 by a sentiment sentiment or polar ESE span). The entity/event-target annotation is anchored to the head word of the NP or VP that refers to the entity or event, and has three slots: id (unique within the document), isNegated (yes or no), and type (entity or event; note that the event type includes both states and events). The *isNegated = yes* option is for the case where the entity/event-target is the negation of the event referred to by the head word, for example, when the source is positive toward someone *not* doing something.

An **sentiment** has one or more span-target annotations in MPQA 2.0. We provide two slots for the k^{th} target annotation. The k-targetSpan slot shows the k^{th} target span. The k-eTarget-link slot[4] is to be filled with a list of ids of entity/event-targets whose text anchors are within the k^{th} target span. An additional slot new-eTarget-link[5] is to be filled with a list of ids of other entity/event-targets.

Each entity/event-target of a **ESE** has two slots, one for the entity/event-target id, and one for an attribute, isReferedInSpan (yes, or no). The value is yes if the entity/event-target is referred to in the ESE span.

4.2.2 Agreement Study

We develop the manual via iterative annotation, discussion, and revision. Once the manual is developed, we participate in an agreement study.

For the formal agreement study, one document was randomly selected from each of the four topics of the OPQA subset (Stoyanov et al., 2005) of the MPQA corpus. They were

[4] An entity/event-target is written as eTarget in the annotation manual and annotation interface.
[5] An entity/event-target is written as eTarget in the annotation manual and annotation interface.

not any of the documents used to develop the manual. We then independently annotated the four documents. There are 292 entity/event-targets in the four documents in total.

To evaluate the results, the same agreement measure is used for both sentiment and ESE entity/event-targets. Given a sentiment or ESE, let set A be the set of entity/event-targets annotated by annotator X, and set B be the set of entity/event-targets annotated by annotator Y. Following (Wilson and Wiebe, 2003), which treat each set A and B in turn as the gold-standard, and we calculate the average F-measure, denoted $agr(A, B)$. The $agr(A, B)$ is 0.82 on average over the four documents, showing good agreement: $agr(A, B) = (|A \cap B|/|B| + |A \cap B|/|A|)/2$. The measurement is the same as the measurement used in the agreement study of annotating the writer's sentiments toward agents and themes of +/-effect events in Section 4.1.2.

4.2.2.1 Disagreement Analysis One issue is whether a sentiment toward an entity or event is indeed communicated in the sentence. Consider this sentence: "*President* ***Mugabe's reelection*** *has been praised by OAU.*" The OAU is positive toward reelection, which is an entity/event-target both annotators mark. The question is whether it is also communicated in this sentence that the OAU is also positive toward Mugabe. X did not mark Mugabe as an entity/event-target, whereas Y did. During the subsequent discussion, X now agrees that it should be marked. In general, X was using what we now consider to be a too conservative policy. Overall, 29% of all disagreements are of this type of borderline case.

8% of the disagreements arise when there are multiple sentiments with overlapping spans, the same source, the same polarity, but different targets and intensities[6]. When there are new entity/event-targets which are not in any target span, annotator X splits the new entity/event-targets into different sentiments based on the intensity, while annotator Y adds the new entity/event-targets to all the sentiments regardless of intensity. Later the annotators discuss to decide which sentiment each new entity/event-target should be linked to in the final version.

31% of the disagreements are caused by negligence, meaning an annotator realized, during

[6]In MPQA 2.0, a sentiment annotation is marked with an intensity (low, medium, or high) representing the intensity.

later discussion, that she should have included an entity/event-target when she saw that the other annotator had included it.

The other disagreements are caused by annotator mistakes. Consider the following sentence: "*We'd like to see further efforts on the part of the U.S.*". We are positive toward the entity/event-target efforts. In addition, one annotator also marked the word see, which is not an entity/event-target, but is instead part of the direct subjectivity (i.e. by saying *would like to see*, we express a positive sentiment toward the entity/event-target efforts). Confusing entity/event-target with direct subjectivity leads to 9% of the disagreements. Another 8% are caused by annotator violations of the manual, and 6% are caused by filling in wrong ids by mistake. The last 8% arise from one annotator not understanding the sentence, given that the annotator is a non-native English speaker.

4.2.3 Statistics and Examples

The current corpus consists of 70 documents, including the subset of the documents in MPQA 2.0 that come from English-language sources (i.e., that are not translations) and a subset of the OPQA subset in MPQA 2.0. A subset contains consensus annotations of X and Y and the rest were annotated by Y. The 70 documents have 1,029 ESEs, 1,287 sentiments, and 1,213 target spans of sentiments (excluding the target span that are marked as *none*) from MPQA 2.0; they have 4,459 entity/event-targets in total. We added 1,366 entity/event-targets to the ESEs and 1,608 entity/event-targets to the target spans. We added 1,485 entity/event-targets which are not in any target span.

In this section, we present an example from the OPQA subset (Stoyanov et al., 2005) to demonstrate how entity/event-targets could help to automatically answer a question. There are opinion and fact questions for each document in the OPQA subset. The sentence below is annotated in MPQA 2.0 to answer the question, "*Is the US Annual Human Rights Report received with universal approval around the world?*" Here the writer is negative toward the report.

> It is due to this hegemony, which the United States wants to maintain, that its State Department makes an assessment of the human rights situation in different countries and

prepares a report on their violations all over the world.

The annotations in MPQA 2.0:
S1: ⟨writer-US, positive, hegemony⟩
S2: ⟨writer, negative, the United States⟩
ESE1: ⟨writer, negative, N/A⟩

First, it is possible for a state-of-the-art system to be trained to recognize the sentiment S1, by the phrase, maintain, and syntax information. But it would be difficult to find S2. There is no direct sentiment modifying the US, nor is there any sentiment or ESE annotation toward the maintaining event or hegemony in MPQA 2.0. Now, in MPQA 3.0, we add the entity/event-target of the ESE1, so that it becomes ⟨writer, negative, hegemony⟩. This is a critical step, because the complete ESE bridges the two sentiments together.

Second, even though we have the two sentiments and the ESE, there is still a gap between the United States in the sentence and the report in the question. One of the entity/event-targets we add is the word report. It is more feasible for a co-reference system to recognize the report in both the sentence and the question as the same thing, than to recognizing that the United States and the report refer to the same concept.

Third, in this sentence, according to the newly added entity/event-targets, the system knows the writer is negative toward both the United States and State Department. When building a knowledge base about the human rights report, this reveals that the two entities have the same stance toward this topic, even without any world knowledge.

4.3 RELATED WORK

Many sentiment corpora that previous works of document-level sentiment analysis task do not need annotations. For example, a movie review from IMDB has a star rating, which can be directly used as the positive or negative class label (Pang et al., 2002; Turney, 2002). As researchers proceed to a finer-grained sentiment analysis, we need annotated corpora.

Annotated corpora of reviews (Hu and Liu, 2004; McDonald et al., 2007) that are widely used in NLP often include target annotations. Such targets are often aspects or features

of products or services. These features are somewhat limited. For example, as stated in SemEval-2014: "We annotate only aspect terms naming particular aspects." Though there are works focusing on identifying which feature the opinion is expressed toward even the feature is not pre-defined, the features in the reviews of a particular product are limited. However, various types of news happen everyday and it is very difficult to identify a fixed set of features describing any news event. For example, car accidents include many elements including both drivers, passengers on both cars, the address, the speed, the details of the vehicles, any injury, the police decision, the insurance, etc.

In most previously developed sentiment corpora, the words or phrases in the text are annotated. On the contrary, to create the Sentiment Treebank (Socher et al., 2013), researchers crowdsourced annotations of movie review data and then overlaid the annotations onto syntax trees. Thus, the targets are not limited to aspects of products/services or text spans. The targets are nodes on the syntax trees which correspond to parts of the sentences. One of the benefits of such annotation is that it makes sure the target is a strictly syntactic unit. The models trained on such dataset can learn the relations of sentiments and mappings to the syntax tree and the context on the syntax tree. However, annotators are asked to annotate small and then increasingly larger segments of the sentence. Thus, the annotations are mixed in the degree to which context was considered when making the judgements.

Different from the corpora mentioned above, our task targets at the news and editorials genre. In such genre, the opinions are not only attributed to the writers, but may be attributed to anyone in the documents. However, the corpora mentioned above assume the sources of opinions are the writers so that it is not necessary to annotate the source. Though our +/-Effect Event Sentiment Corpus, as a pilot study, is annotated with writer's sentiments only, the later developed MPQA is annotated with opinions that are attributed to the writers or the entities in the documents. The MPQA opinion annotated corpus (Wiebe et al., 2005; Wilson, 2008) is entirely span-based, and does not contain any entity/event-target annotation. However, it provides an infrastructure for sentiment annotation that is not provided by other sentiment NLP corpora, and is much more varied in topic, genre, and publication source.

4.4 SUMMARY

Corpora have been annotated in the past for explicit sentiment expressions. In this chapter, we have developed corpora annotated for both explicit and implicit sentiments. We have observed that sentiment inferences arise from interactions between sentiment expressions and +/-effect events. Thus, in Section 4.1 we have developed the +/-Effect Event Sentiment Corpus (Deng et al., 2013) annotated with the +/-effect information and the sentiments toward the agents and the themes. It fills in a gap by presenting an annotation scheme for +/-effect events and the writer's sentiments toward the agents and themes of those events. We have conducted an agreement study, the results of which are positive. Further, we have carried out consensus study providing a better estimation of how many disagreements were caused by negligence.

As a further step as shown in Section 4.2, we have developed the MPQA 3.0 corpus (Deng and Wiebe, 2015b) in which entity and event target annotations are added to the existing MPQA 2.0 corpus. Building upon the existing well-annotated corpus saves us much time in annotating opinions from scratch, and it preserves the original annotation structures in the existing corpus. We have designed a good annotation scheme as a transition from phrase-level sentiment corpora to entity/event-level sentiment corpora. Similarly, we have conducted an agreement study, the results of which are positive.

Both corpora are annotated with both explicit and implicit sentiments. The corpora are promising to provide good resources for the research focusing on identifying implicit sentiments.

5.0 RULES OF ENTITY/EVENT-LEVEL SENTIMENT INFERENCE

As defined in Chapter 3, an explicit opinion is associated with an opinion expression, while an implicit opinion is not associated with any opinion expression. In order to recognize the implicit sentiments, an automatic system needs to know the guidelines of inferences. The focus of this chapter is formally representing the inferences as a set of rules so that an automatic system can use them to infer sentiments. Basically, we manually craft rules that "guide" the systems to know what to infer based on the existing evidence recognized by state-of-the-art sentiment analysis systems and other auxiliary NLP systems.

Wiebe and Deng (2014) have developed inference rules to infer implicit opinions. One of the rules applied to (Ex1.1) "*It is great that Clinton defeated Trump*" is:

writer +sentiment toward defeated
-effect Trump →
writer -sentiment toward Trump

where "+sentiment" represents positive sentiment, "-sentiment" represents negative sentiment, and "-effect" represents a -effect event happening to someone. There are two pieces of evidence supporting this inference. First, the writer is positive toward the defeating event. Second, the defeating event has a negative effect on Trump. Then the rule infers that the writer is negative toward Trump. The rules in (Wiebe and Deng, 2014) are expressed using natural languages as the rule above. We can use various methods to present the natural language rules in the format that an automatic system can use them to infer sentiments.

In this chapter, we present the rules in (Wiebe and Deng, 2014) as propositional logic using the definitions of entity/event-level sentiment defined in Chapter 3. Using propositional logic has several benefits. First of all, it is natural to represent inference as the derivatives of propositional logic. Second, the representations of entity/event-level sentiment defined in

Chapter 3 can be seen as propositional literals in the propositional logic. Later in Chapter 6 we will show that each propositional literal defined by a representation corresponds to a Natural Language Processing (NLP) task. The value of a propositional literal clearly corresponds to an output from an individual NLP task. Such correspondence gives us the big benefit that different NLP tasks can be bridged together and can be analyzed simultaneously. Third, the propositional logic is intuitive to understand and flexible to revise as well. If we want to add more conditions or relax conditions of to a rule in the future, we can add or delete propositional literals in the propositional logic without changing to another set of representations.

Two sets of rules will be presented in this chapter. We first introduce the inference rules applied to +/-effect events in Section 5.1. The evidence for inferences of this set of rules are sentiments expressed toward +/-effect events and the entities involved in the +/-effect events (i.e., agents and themes). Then we present the inference rules that can be used to infer sentiments expressed toward events not limited to +/-effect events in Section 5.2.

5.1 +/-EFFECT EVENT INFERENCE RULES

As introduced in Chapter 1, many implicit sentiments are expressed via +/-effect events. It's is natural for humans to perceive the implicit sentiments but it is difficult for automatic systems to recognize them. An automatic system should be guided to infer implicit sentiments. In this section, we first introduce the "guidelines" expressed as propositional rules applied to +/-effect events, in order to teach automatic systems how to infer sentiments in Section 5.1.1. Furthermore, it is important to demonstrate the consistency of the rules. In other words, we need to check whether the rules actual work in the various context before the rules are used in the automatic systems. Thus, we have developed a graph-based model to validate the consistency of the rules. The experiments have shown that the inference rules are able to correctly infer sentiments in 89% context in the dataset. The graph-based model and the corresponding experiment are introduced in Section 5.1.2

5.1.1 +/-Effect Event Inference Rules

Since we define the rules to teach an automatic system about how humans think, let's recall how we perceive the sentiments in an example. Here we begin with the example (Ex1.1), shown again below, from Chapter 1. Here we present the rules applied to the sentiments perceived in (Ex1.1), using the representations introduced in Chapter 3.

(Ex1.1) It is great that Hillary Clinton defeated Donald Trump.

The explicit sentiment is positive toward the defeating event, and the implicit sentiments are positive toward Clinton and negative toward Trump.

The rules applied to (Ex1.1) are:

PositivePair(writer, defeat) ∧ -Effect(defeat) ∧ Theme(defeat, Trump)
→ NegativePair(writer, Trump)
PositivePair(writer, defeat) ∧ -Effect(defeat) ∧ Agent(defeat, Clinton)
→ PositivePair(writer, Clinton)

In the first rule, it is inferred that the writer is negative toward Trump since the writer is positive toward a -effect event happening to him. In the second rules, it is inferred that the writer is positive toward Clinton since the writer is positive toward a -effect event that Clinton conducts. The inferences according to the rules are consistent with the descriptions in Chapter 1 when we first introduce these examples. The rules listed above are instantiations applied to (Ex1.1).

PositivePair(S, T) ∧ -Effect(T) ∧ Theme(T, H) → NegativePair(S, H)
PositivePair(S, T) ∧ -Effect(T) ∧ Agent(T, A) → PositivePair(S, A)

According to the first rule, if the entity S has a positive sentiment toward T, and T is a -effect event whose theme is H, then it is inferred that S is negative toward H. According to the second generalization rule, if the entity S has a positive sentiment toward T, and T is a -effect event whose agent is A, then it is inferred that S is positive toward A.

In total we have 16 generalization rules inferring sentiments toward +/-effect events and the entities involved in the events. All the rules are listed in Appendix B. Compared to the rule expressed in natural language at the beginning of this section, the generalization

rules are clearer as they decompose the inferences into separate literals in the propositions. Later in Chapter 6, we will present that each literal (w.g., PositivePair(S,T), -Effect(T), Theme(T, H), etc) corresponds to a natural language processing task.

Let's step through a more complicated example. Here we do not repeat the rules listed in Appendix B. We try to explain the rules in natural languages and just refer to the rules in the appendix.

> (Ex5.1) Why would [President Obama] **support** [health care reform]? Because [reform] could **lower** [*skyrocketing* health care costs], and **prohibit** [private insurance companies] from **overcharging** [patients].

Suppose a sentiment analysis system recognizes only one explicit sentiment expression, (skyrocketing). According to the annotations, there are several +/-effect events. Each is listed below in the form ⟨agent, +/-effect event, theme⟩.

E_1: ⟨reform, lower, costs⟩
E_2: ⟨reform, prohibit, E_3 ⟩
E_3: ⟨companies, overcharge, patients⟩
E_4: ⟨Obama, support, reform⟩

In E_1, from the negative sentiment (skyrocketing, i.e., the writer is negative toward the costs because they are too high), and the fact that the costs are the theme of a -effect event (lower), The rule (Rule2.12) infers **a positive sentiment toward** E_1 since the lowering event decreases the cost.

Now, (Rule2.2) applies. We infer the writer is **positive toward the reform**, since it is the agent of E_1 and it initiates E_1, toward which the writer is positive.

E_2 illustrates the case where the theme is an event. Specifically, the theme of E_2 is E_3, a -effect event (overcharging). As we can see, E_2 keeps E_3 from happening. Events such as E_2 are reversers. Recall that reversers may be seen as -effect events as defined in Section 3.2.1, because they make their themes irrealis (i.e., not happen).

Above, we have inferred that the writer is positive toward the reform, the agent of E_2. By (Rule2.10), we can infer that the writer is **positive toward** E_2 since the writer likes the reform and subsequently likes what the reforms does. Then by (Rule2.4) the writer is **negative toward** E_3, the theme of E_2 since the writer likes E_2 being harmful to E_3.

m is a +/-effect event			a is the agent	h is the theme
+EFFECT(m)	POSITIVEPAIR(s,m)	→	POSITIVEPAIR(s,a)	POSITIVEPAIR(s,h)
-EFFECT(m)	POSITIVEPAIR(s,m)	→	POSITIVEPAIR(s,a)	POSITIVEPAIR(s,h)
-EFFECT(m)	POSITIVEPAIR(s,m)	→	POSITIVEPAIR(s,a)	NEGATIVEPAIR(s,h)
-EFFECT(m)	NEGATIVEPAIR(s,m)	→	NEGATIVEPAIR(s,a)	NEGATIVEPAIR(s,h)

Table 5.1: Combinations of rules inferring sentiments toward +/-effect event and the entities.

For E_3, we know the writer is **positive toward patients** by (Rule2.8) and the writer is **negative toward companies** by (Rule2.6).

Turning to E_4, supporting the health care reform is good for the reform. We already inferred the writer is positive toward the reform. (Rule2.11) infers that the writer is **positive toward** E_4. (Rule2.1) then infers that the writer is **positive toward the agent of** E_4**, Obama**.

In summary, we infer that the writer is positive toward E_1, health care reform, E_2, patients, E_4, and Obama, and negative toward E_3 and private insurance companies.

In addition to the rules listed in the appendix , we have observed an interesting phenomenon of the rules (Deng and Wiebe, 2014b). The combinations of a subset of rules are presented in Table 5.1, where each line the sentiments toward a +/-effect event (m) infers the sentiment toward the agent of it (a) and the sentiment toward the theme of it (h). From Table 5.1, we have observed that, regardless of the sentiment of the source s toward the event, **if the event is +effect, then the sentiments of the source** s **toward the agent and theme are the same, while if the event is -effect, the sentiments of the source** s **toward the agent and theme are opposite**. Later we will use such relations as constraints in the experiments.

5.1.2 Validating the Rules: A Graph-based Propagation Model

We develop a graph-based model consisting of the +/-effect events and the agents and the themes. The graph classifies the sentiments toward entities (i.e., positive or negative) by propagation, defined by the inference rules in Section 5.1.1. In this section, we introduce the

definition of the graph in Section 5.1.2.1 and the propagation algorithm in Section 5.1.2.2. We carry out an experiment to examine the performance of inferring implicit sentiments based on the rules in Section 5.1.2.3.

5.1.2.1 Definition of the Entity Graph. We define a graph $EG = \{N, E\}$. Each node in the node set N represents an annotated noun phrase agent or theme span. The label of each node is positive or negative, representing the writer's sentiment toward it. Each edge in the edge set E links two nodes if they co-occur in a +/-effect event. The label of each edge is +effect or -effect, representing the event that the two nodes participate in. If there are more than two entities involved in an event, we define that pairwise edges are created connecting each pair of nodes in the event. Let's review (Ex5.1) below.

(Ex5.1) Why would President Obama support health care reform? Because the reform could lower skyrocketing health care costs, and prohibit private insurance companies from overcharging patients.

E_1: ⟨reform, lower, **skyrocketing** costs⟩
E_2: ⟨reform, prohibit, E_3 ⟩
E_3: ⟨companies, overcharge, patients⟩
E_4: ⟨Obama, support, reform⟩

According to the annotations, the node of reform is linked to the node of costs via E_1 and it is linked to the node of Obama via E_4. Note that, the two +/-effect events E_2 and E_3 are **linked in a chain**: ⟨reform, prohibit, ⟨companies, overcharge, patients⟩ ⟩. The three nodes, reform, companies and patients, participate in this chain; thus, pairwise edges exist among them. The edge linking the nodes companies and patients is -effect (i.e., overcharging). The edge linking the nodes reform and companies is also a -effect event since we treat a reverser as a -effect event. The edge linking the nodes reform and patients encodes two -effect events (i.e., prohibit and overcharge); computationally we say two -effect events result in a +effect event, so the edge linking the two is +effect. Also, two +effect events result in a +effect event; a combination of +effect event and a -effect event results in a -effect event.

Given a text, we get the spans of +/-effect events and their agents and themes plus the effect of the events (i.e., +effect/-effect) from the gold standard annotations, and then build the graph upon them. In other words, the structure of the graph and the label of the edges

initialize all $m_{i\to j}(pos) = m_{i\to j}(neg) = 1$

repeat

  foreach $n_i \in N$ **do**

    foreach $n_j \in Neighbor(n_i)$ **do**

      foreach $y \in pos, neg$ **do**

        calculate $m_{i\to j}(y)$

        normalize $m_{i\to j}(pos) + m_{i\to j}(neg) = 1$

until all $m_{i\to j}$ stop changing;

for each $n_i \in N$ **assign its polarity as**

  $\underset{y\in pos,neg}{\text{argmax}} \Phi_i(y) * \prod_{n_k \in Neighbor(n_i)} m_{\ k\to i}(y)$

  neutral, in case of a tie

Table 5.2: Loopy Belief Propagation

are taken from gold standard annotations. However, the manual annotations of the writer's sentiments toward the agents and themes are used as the gold standard for evaluation.

5.1.2.2 Sentiment Inference via LBP The goal is to classify the writer's sentiments toward each node (i.e., each entity) on the graph. With graph EG containing cycles and no apparent structure, we utilize an approximate collective classification algorithm, *loopy belief propagation (LBP)* (Pearl, 1982; Yedidia et al., 2005), to classify nodes through belief message passing. The algorithm is shown in Table 5.2.

In LBP, each node has a score, $\Phi_i(y)$, and each edge has a score, $\Psi_{ij}(y_i, y_j)$. In our case, $\Phi_i(y)$ represents the writer's sentiment toward n_i. $\Psi_{ij}(y_i, y_j)$ is the score on edge e_{ij}, representing the likelihood that the sentiment toward the node n_i is the polarity y_i and the sentiment toward the node n_j is the polarity y_j. Since the inference rules define the relations of sentiment polarities between agents and themes, **we encode inference rules as definitions of the edge score** $\Psi_{ij}(y_i, y_j)$. Recall that in Table 5.1 the sentiments toward

the agent and the theme of a +effect event are the same. Therefore, we define $\Psi_{ij}(pos, pos)$ and $\Psi_{ij}(neg, neg)$ to be 1 if the two nodes i and j are linked by a +effect edge; otherwise, it is 0. Similarly, the sentiments toward the agent and the theme are opposite. Thus we define $\Psi_{ij}(neg, pos)$ and $\Psi_{ij}(pos, neg)$ to be 1 if the two nodes are linked by a -effect edge; otherwise, it is 0.

LBP is an iterative message passing algorithm. A message from n_i to n_j over edge e_{ij} has two values: $m_{i\to j}(pos)$ is how much information from node n_i indicates node n_j is positive, and $m_{i\to j}(neg)$ is how much information from node n_i indicates node n_j is negative. In each iteration, the two are normalized such that $m_{i\to j}(pos) + m_{i\to j}(neg) = 1$. The message from n_i to its neighbour n_j is computed as:

$$\begin{aligned} m_{i\to j}(pos) = \Psi_{ij}(pos, pos) * \Phi_i(pos) * \prod_{n_k \in \text{Neighbor}(n_i)/n_j} m_{k\to i}(pos) \\ + \Psi_{ij}(neg, pos) * \Phi_i(neg) * \prod_{n_k \in \text{Neighbor}(n_i)/n_j} m_{k\to i}(neg) \end{aligned} \tag{5.1}$$

$$\begin{aligned} m_{i\to j}(neg) = \Psi_{ij}(neg, neg) * \Phi_i(neg) * \prod_{n_k \in \text{Neighbor}(n_i)/n_j} m_{k\to i}(neg) \\ + \Psi_{ij}(pos, neg) * \Phi_i(pos) * \prod_{n_k \in \text{Neighbor}(n_i)/n_j} m_{k\to i}(pos) \end{aligned} \tag{5.2}$$

For example, the first part of Equation (5.1) means that the positive message n_i conveys to n_j (i.e., $m_{i\to j}(pos)$) comes from n_i being positive itself ($\Phi_i(pos)$), the likelihood of edge e_{ij} with its nodes n_i being positive and n_j being positive ($\Psi_{ij}(pos, pos)$), and the positive message n_i's neighbors (besides n_j) convey to it ($\prod_{k \in Neighbor(n_i)/n_j} m_{k\to i}(pos)$).

After convergence, the polarity of each node is determined by its explicit sentiment and the messages its neighbors convey to it, as shown at the end of the algorithm in Table 5.2. By this method, we take into account both sentiments and the interactions between entities via +/-effect events in order to discover implicit attitudes.

Note that the node scores $\Phi_i(y)$ and edge scores $\Psi_{ij}(y_i, y_j)$ are determined initially and do not change. Only $m_{i\to j}$ changes from iteration to iteration.

5.1.2.3 Performance of Graph Model.

In this section we examine whether the graph model is able to correctly propagate sentiments. We perform an experiment to assess the chance of a node being correctly classified via the graph.

In each subgraph (connected component), we assign one of the nodes in the subgraph with its gold standard label (the writer's sentiment toward it). Then we run LBP on the subgraph and record whether the other nodes in the subgraph are classified correctly or not. The experimented is designed to demonstrate the inference ability of the graph-based model. Thus as noted in Section 5.1.2.1 the structure of the graph and the labels of the edges are taken from the gold standard annotations. We also assign one of the nodes with its gold standard label (the writer's sentiment toward it).

The experiment is run on the subgraph $|S|$ times, where $|S|$ is the number of nodes in the subgraph, so that each node is assigned its gold-standard polarity exactly once. Each node is given a propagated value $|S| - 1$ times, as each of the other nodes in its subgraph receives its gold-standard polarity.

To evaluate the chance of a node given a correct propagated label, we use the equations (5.3) and (5.4).

$$correct(a|b) = \begin{cases} 1 & a \text{ is correct} \\ 0 & otherwise \end{cases} \tag{5.3}$$

$$correctness(a) = \frac{\sum_{b \in S_a, b \neq a} correct(a|b)}{|S_a| - 1} \tag{5.4}$$

where S_a is the set of nodes in a's subgraph. Given b being assigned its gold-standard polarity, if a is classified correctly, then $correct(a|b)$ is 1; otherwise 0. $|S_a|$ is the number of nodes in a's subgraph. $correctness(a)$ is the percentage of assignments to a that are correct. If it is 1, then a is correctly classified given the correct classification of any single node in its subgraph.

For example, suppose there are three nodes in a subgraph, A, B and C. For A we (1) assign B its gold label and carry out propagation on the subgraph, (2) assign C its gold label and carry out propagation again, then (3) calculate $correctness(A)$. Then the same process is repeated for B and C.

Dataset	# subgraph	correctness
all subgraphs	983	0.8874
multi-node subgraphs	169	0.9030

Table 5.3: Performance of graph model itself.

Some subgraphs contain only two nodes, the agent and the theme. In this case, graph propagation corresponds to single applications of two implicature rules. Other subgraphs contain more nodes. Two results are shown in Table 5.3. One is the result on the whole experiment data, the other is the result for all nodes whose subgraphs have more than two nodes.

As we can see, a node has an 89% chance of being correct if there is one correct explicit subjectivity node in its subgraph. If we only consider subgraphs with more than two nodes, the correctness chance is higher. The results indicate that, if given correct sentiments, the graph model will assign the unknown nodes with correct labels 89% of the time. Further, the results indicate that the inference rules defined in Section 5.1.1 are consistent most of the time across the corpus.

5.2 SENTIMENT INFERENCE RULES

The rules introduced in the previous section are applied to only +/-effect events. However, not all the events are +/-effect events. Various implicit sentiments are expressed via various types of events. Compare the example (Ex1.3) previously mentioned in Chapter 1 and the example below.

(Ex1.3) It is great that Mike Pence **stands by** Donald Trump.
(Ex5.2) It is great that Mike Pence is **in favour of** Donald Trump.

In both examples, it is indicated in the sentence that the writer is positive toward Trump and Pence. There is an explicitly positive sentiment (great), in both examples. The difference

comes from the target of the positive sentiment. In (Ex1.3), the target is a +effect event, stand by. By rules introduced in the previous section, we can infer the positive sentiment toward Trump and Pence. In (Ex5.2), the target is the phrase, Mike Pence is in favour of Donald Trump, which represents a positive sentiment expressed from Pence toward Trump. In this case, the target of the explicitly positive sentiment is another explicitly positive sentiment. Thus, though it is not a +/-effect event, the implicit sentiments are indeed indicated in the text. In (Wiebe and Deng, 2014), inference rules are defined to infer more opinions when the target of an opinion is another opinion. We represent the rules using the representations defined in Chapter 3. For example, the instantiated rules applied to (Ex5.2) are:

PositivePair(writer, in favor of) $\wedge$ +Sentiment(in favour of) $\wedge$ Target(in favor of, Trump) $\rightarrow$ PositivePair(writer, Trump)
PositivePair(writer, in favor of) $\wedge$ +Sentiment(in favour of) $\wedge$ Source(in favor of, Pence) $\rightarrow$ PositivePair(writer, Pence)

The inference rules are novel in that the target of a sentiment may be another sentiment (i.e., **sentiment toward sentiment** structure). The inference rules link sentiments to sentiments and, transitively, link entities to entities (e.g., from writer to Trump and to Pence). A full list of the rules is in Appendix C.

Ultimately, each rule in this section and the rules about inferring sentiments toward +/-effect events and the entities in the events in Section 5.1 are implicatures (Wiebe and Deng, 2014). Implicatures are defeasible, meaning that the inference may be blocked given outside evidence to the contrary (Greene and Resnik, 2009) and the inference should not be carried out. In some circumstances the state-of-the-art works may recognize the sentiment toward an entity/event toward which the rules also infer the sentiment. The inferred sentiment may be contradictory to the sentiment that is recognized by state-of-the-art works. There is no hard rule to say when a conflict exists which result is more confident. Therefore, instead of building a rule-based system where each rule is used as hard constraints, we devote to developing joint prediction models where the rules are used as soft constraints in the model. Eventually, the model pursues a trade-off between trusting the recognized sentiments by state-of-the-art systems and sticking to the rules.

5.3 RELATED WORK

Most works in NLP address explicit sentiment, but some address implicit sentiment. For example, Zhang and Liu (2011) identify noun product features that imply opinions, and Feng et al. (2013) identify objective words that have positive or negative connotations. However, identifying terms that imply opinions is a different task than sentiment propagation between entities. Dasigi et al. (2012) search for implicit sentiments shared between authors, while we address inferences within a single text.

Several papers apply compositional semantics to determine polarity (Moilanen and Pulman, 2007b; Choi and Cardie, 2008; Moilanen et al., 2010; Liu, 2012, etc.). The goal of such works is to determine one overall polarity of an expression or sentence. In contrast, our works commit to a holder having sentiments toward various events and entities in the sentence, possibly of different polarities.

The idea of +/-effect events in sentiment analysis is not entirely new. For example, two papers mentioned above (Zhang and Liu, 2011; Choi and Cardie, 2008) include linguistic patterns for the tasks that they address that include +/-effect events, but they don't define general sentiment inference rules relating sentiments and +/-effect events, agents, and objects as we do. Recently, in linguistics, Anand and Reschke (2010); Reschke and Anand (2011) identify classes of +/-effect terms, and carry out studies involving artificially constructed +/-effect triples and corpus examples matching fixed linguistic templates. Our works focus on +/-effect triples in naturally-occurring data and uses generalized implicature rules. Goyal et al. (2012) generate a lexicon of *patient polarity verbs*, which correspond to +/-effect events whose spans are verbs. Riloff et al. (2013b) investigate sarcasm where the writer holds a positive sentiment toward a negative situation. However, neither of these works performs sentiment inference.

Wiebe and Deng (2014) introduce the the inference in the *sentiment toward sentiment* structure. The inferences are mixtures of sentiments and beliefs. In this chapter, the inference rules we define are simplifications which are only applied to sentiments. It is assumed that the sources of sentiments believe the opinions exist. The analysis of sentiment and belief is outside the scope of this thesis.

5.4 SUMMARY

Two sets of sentiment inference rules have been defined in this chapter. The rules are expressed in propositional logic. The first set of rules, defined in Section 5.1, fills in the gap of the +/-effect event information and the sentiments expressed toward them. Basically, if a sentiment is expressed toward one component of the ⟨agent, +/-effect event, theme⟩, then the sentiments toward the other two components can be inferred. The sentiment inference rules are novel in that they show dependencies between the outputs from the information extraction field which are entities, events and relations and the outputs from the sentiment analysis field. As we will present in the next chapter, such dependencies are important to jointly analyze different tasks together to improve performances in each task. We have also built a graph-based model defined by the sentiment inference rules to infer the sentiments expressed toward the agents and themes in Section 5.1.2. we find it has an 89% chance of propagating sentiments correctly. This is a good indicator that the rules give the correct inference in most contexts. Based on the sentiment inference rules about sentiments toward +/-effect event information, we further define the second set of rules in Section 5.2. The second set of rules infers more sentiments in the *sentiment toward sentiment* structure. Though the superficial form of the second set of rules is similar to the first set of rules, the two sets are from different perspectives. The first set of rules combines different NLP tasks, while the second set of rules dives deeper into the sentiment analysis outputs. Later we will show that both sets of rules are useful in improving recognizing entity/event-level sentiments.

6.0 COMPUTATIONAL MODELS OF ENTITY/EVENT-LEVEL SENTIMENT DETECTION AND INFERENCE

The ultimate goal of the task in this thesis is to utilize +/-effect event information and inference rules to improve detecting entity/event-level sentiments in the sentences. We decompose this task into several subtasks, as shown in Figure 6.1. There are ambiguities in each subtask. In this chapter, we start with illustrating the ambiguities in each subtask, then introduce models aiming at automatically solving the ambiguities in different subtasks, including solving the ambiguities of sentiments. We also carried out the experiments and the results have shown that the models achieve better performances than state-of-the-art works in identifying entity/event-level sentiments.

Let's first look at the subtasks illustrated by Figure 6.1.

(1) The region in the blue circle in Figure 6.1 represents the +/-effect events and the agents and themes to be identified. The ambiguities come from: (1.1) Which spans are +/-effect events? (1.2) Which noun phrases are the agents, which are the themes? (1.3) What is the effect of the +/-effect event? (1.4) Is the effect reversed?

(2) The region in the red circle represents sentiments we need to extract from the document. The ambiguities are: (2.1) Is there any explicit sentiment? (2.2) What are the sources, targets and polarities of the explicit sentiments? (2.3) Is there any implicit sentiment inferred? (2.4) What are the sources, targets and polarities of the implicit sentiments?

(3) The region in the green circle represents all types of subjectivities of the writer, including sentiments, beliefs and arguing . The ambiguities are similar to those in the red circle: (3.1) Is there any subjectivity of the writer? (3.2) What are the targets and polarities of the subjectivity?

Though there are many ambiguities, they are interdependent. Inference rules

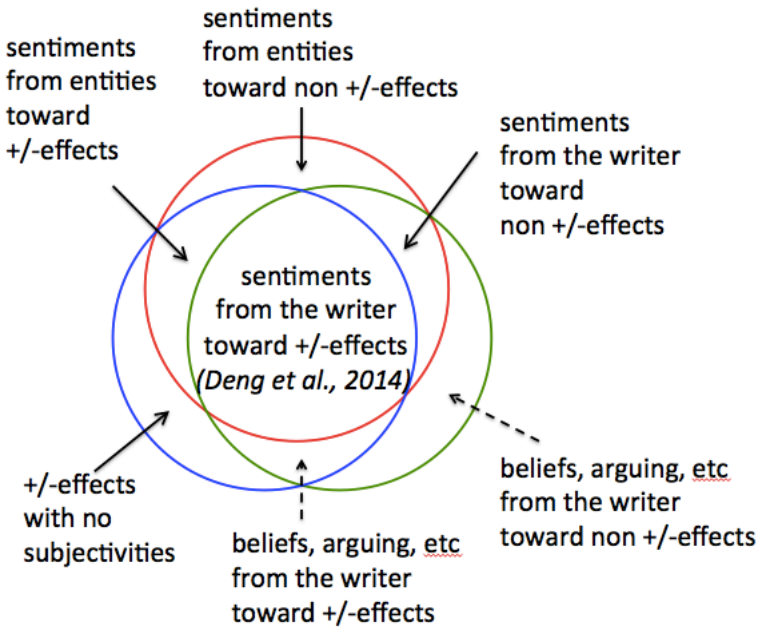

Figure 6.1: Overview of Subtasks.

define dependencies among these ambiguities. Let's look at one of the inference rules to see how inference rules build connections among these tasks.

$$\textsc{PositivePair}(S, T) \wedge +\textsc{Effect}(T) \wedge \textsc{Theme}(T, H) \rightarrow \textsc{PositivePair}(S, H)$$

Each propositional literal in the propositional logics (i.e., PositivePair(S, T), +Effect(T)) corresponds to a Natural Language Processing (NLP) task. For example, PositivePair(S, T) is the goal of a sentiment analysis task. Theme(T, H) is the goal of a semantic role labeling task. If we replace the literal (e.g., PositivePair(S, T)) with grounded variables (e.g., PositivePair(writer, Trump)), the grounded literal can be assigned with values representing whether the grounding is true or false. The values of the grounded literals can be drawn from an automatic system of the corresponding task, or they could be inferred by the logics. Recall that in Section 5.1.2, we build a graph-based model to demonstrate the validity of the rules represented by the logics. Each node of the graph corresponds to an agent or theme of a +/-effect event, and each edge connects two nodes together if the two nodes participate in the same event. In that experiment, the values of

the grounded propositional literals +Effect and Theme are taken from the gold standard annotations. The value of the grounded propositional literal of PositivePair in the body (left hand side) of logics are also taken from the gold standard annotations. The model only outputs the values of the grounded propositional literal PositivePair in the head (right hand side) of logics. Further, in the experiment the sources of all sentiments are assumed to be the writer.

Different from the graph-based model in Section 5.1.2, the models in this chapter aim at automatically assigning values to all the grounded propositional variables in the logics. Previously in Section 5.1.2 the only ambiguity is sentiment: whether a PositivePair or a NegativePair is true or false. In this chapter, the ambiguities include each propositional literal in the logics. With these extensions, there are many more ambiguities in the model. Fortunately the inference rules define constraints among different ambiguities and they are used as constraints in the models in this chapter. Using the inference rules as constraints has several benefits. First, the increased complexity added by the new ambiguities in the extended regions will be mitigated by more constraints defined by the inference rules. Second, the rules connect different NLP tasks together. Assigning true or false to PositivePairs and NegativePairs is the task of sentiment analysis. Assigning true or false to +Effects and -Effects is the task of sense disambiguation. Assigning true or false to Agent and Theme is the task of semantic role labeling. The rules build a bridge connecting different NLP tasks together, while previous works treat these tasks as separate tasks. Third, the rules force the model to choose an optimal set of **all** labels assigned to all the propositional literals in all the propositional logics instead of selecting the optimal label of individual values separately. Later we will see that such constraints may even correct some errors made by individual systems that are designed for individual literal.

In this chapter, we first begin with an easier task. In Section 6.1 we develop an Integer Linear Programming (ILP) model to recognize writer's sentiments toward the agents and themes in the +/-effect events. The rules in Section 5.1 are used as constraints. The experiments have shown that the ILP model is able to improve recognizing sentiments over state-of-the-art works. The task in Section 6.1 using ILP corresponds to the intersection of the three regions in Figure 6.1. Though the ILP task is only part of the whole task, the

success of it encourages us to extend our works from the intersection to all the regions with *solid lines* pointed to: the sources of sentiments are not limited to only the writer but all entities, and the targets of sentiments are not only the agents and themes of +/-effect events, but are all entities and events, including other sentiments.

In Section 6.2 we develop a Probabilistic Soft Logic (PSL) model to recognize entity/event-level sentiments, each of which is any entity's sentiment (including the writer) toward any entity or event in the sentences. The rules in Section 5.1 and Section 5.2 are used as constraints. The experiments also show that the PSL model is able to improve recognizing entity/event-level sentiments over state-of-the-art works. Note that though there are multiple tasks in the models, the model disambiguates all the ambiguities and assign labels to all the tasks simultaneously.

The two regions with dashed lines pointed to are analyzing writer's subjectivities other than sentiments, including belief, arguing, etc. Although that topic is out of scope in this thesis, the joint models presented in this chapter promise to jointly analyze different types of subjectivities in the future.

6.1 +/-EFFECT EVENT SENTIMENT INFERENCE

In this section, we will talk about an optimization model which only needs manual annotations of +/-effect event spans. The model is able to **jointly** resolve ambiguities of (1) the effect of events, (2) whether the effect is reversed ,(3) the agents and themes of the events, and (4) the writer's sentiments toward agents and themes. In the model, we first run local systems to extract candidates for each ambiguity of (1)-(4), then an Integer Linear Programming (ILP) model chooses one from the candidates of each ambiguity, which is an optimal subset of all the candidates. Later the experiments show that the joint prediction achieves better performance than local systems. For example, recall (Ex1.1) in the Chapter 1.

(Ex1.1) It is great that Hillary Clinton defeated Donald Trump.

The input to the model is the sentence and a text span, defeated, representing there

is a +/-effect event in the sentence. The reason why the model does not automatically recognize +/-effect event spans is that the gold standard we use for evaluation contains sentiment annotations only toward the agents and themes of +/-effect events. We are only able to evaluate true hits of +/-effect events. Thus,the input to the system is the set of the text spans marked as +/-effect events in the corpus. But the system is not given whether the event is +effect or -effect. The output of the model working on (Ex1.1) is expected to include: (1) it is a -effect event, (2) it is not reversed, (3) the agent is Hillary Clinton and the theme is Donald Trump, (4) the writer is positive toward the agent and the writer is negative toward the theme.

The ILP joint prediction is performed over two sets of variables. The first set is *EffectEvent*, containing a variable for each +/-effect event in the document. The other set is *Entity*, containing a variable for each agent or theme candidate. Each variable k in *EffectEvent* has its corresponding agent and theme variables, i and j, in *Entity*. The three form a triple unit, $\langle i, k, j \rangle$. The set *Triple* consists of each $\langle i, k, j \rangle$, recording the correspondence between variables in *EffectEvent* and *Entity*. The goal of the model is to assign optimal labels to variables in *EffectEvent* and *Entity*.

In this section, we first introduce the ILP applied to this task model in Section 6.1.1, and we talk briefly about various local systems developed to assign local scores to the labels of variables in Section 6.1.2. Finally we will present the experiment result in Section 6.1.3.

6.1.1 Integer Linear Programming model

We extract two agent candidates and two theme candidates for each +/-effect event (one each will ultimately be chosen by the ILP model).[1] We use syntax, and the output of the SENNA (Collobert et al., 2011) semantic role labeling tool. SENNA labels the A0 (subject), A1 (object), and A2 (indirect object) spans for each predicate, if possible. To extract the *semantic agent* candidate: If SENNA labels a span as A0 of the +/-effect event, we consider it as the semantic agent; if there is no A0 but A1 is labeled, we consider A1; if there is no A0 or A1 but A2 is labeled, we consider A2. To extract the *syntactic agent* candidate, we

[1]This model is able to handle any number of candidates. The methods we tried using more candidates did not perform as well - the gain in recall was offset by larger losses in precision.

find the nearest noun in front of the +/-effect span, and then extract any other word that depends on the noun according to the dependency parse. Similarly, to extract the *semantic theme* candidate, we consider A1, A2, A0 in order. To extract the *syntactic theme* candidate, the same procedure is conducted as for the syntactic agent, but the nearest noun should be after the +/-effect. If there is no A0, A1 or A2, then there is only one agent candidate, *implicit* and only one theme candidate, *null*. We treat a *null* theme as an incorrect span in the later evaluations, since each +/-effect event must have at least one theme according to the definition in Section 3.2.1. If the two agent (theme) candidate spans are the same, there is only one candidate.

We use Integer Linear Programming (ILP) to assign labels to variables. Variables in *Entity* will be assigned *positive* or *negative*, representing the writer's sentiments toward them. We may have two candidate agents for a +/-effect and that we will choose between them. Thus, only one agent is assigned a *positive* or *negative* label; the other is considered to be an incorrect agent of the +/-effect (similarly for the theme candidates). Each variable in *Event* will be assigned the label *+effect* or *-effect*. Optionally, it may also be assigned the label *reversed*. Label *+effect* or *-effect* is the effect of the event; *reversed* is assigned if the effect is reversed (e.g., for "not harmed", the labels are *-effect* and *reversed*).

The objective function of the ILP is:

$$\min_{u_{1pos}, u_{1neg} \ldots} \left(-1 * \sum_{i \in \text{Event} \cup \text{Entity}} \sum_{c \in L_i} p_{ic} u_{ic} \right) + \sum_{\langle i,k,j \rangle \in \text{Triple}} \xi_{ikj} + \sum_{\langle i,k,j \rangle \in \text{Triple}} \delta_{ikj} \tag{6.1}$$

subject to

$$u_{ic} \in \{0, 1\}, \forall i, c \tag{6.2}$$

$$\xi_{ikj}, \delta_{ikj} \in \{0, 1\}, \forall \langle i, k, j \rangle \in Triple \tag{6.3}$$

where L_i is the set of labels given to $\forall i \in Event \cup Entity$. If $i \in Event$, L_i is {+effect, -effect, reversed} ({*+effect, -effect, r*}, for short). If $i \in Entity$, L_i is {positive, negative} ({*pos, neg*}, for short). u_{ic} is a binary indicator representing whether the label c is assigned to

the variable i. When an indicator variable is 1, the corresponding label is selected. p_{ic} is the score given by local detectors, introduced in the following sections. Variables ξ_{ikj} and δ_{ikj} are binary slack variables that correspond to the constraints of $\langle i,k,j \rangle$, defined by the sentiment inference rules. When a given slack variable is 1, the corresponding triple violates the constraints. Minimizing the objective function could achieve two goals at the same time. The first part $(-1 * \sum_i \sum_c p_{ic} u_{ic})$ tries to select a set of labels that maximize the scores given by the local detectors. The second part $(\sum_{ikj} \xi_{ikj} + \sum_{ikj} \delta_{ikj})$ aims at minimizing the cases where +/-effect implicature constraints are violated. Here we do not force each triple to obey the sentiment inference rules, but to minimize the violating cases. For each variable, we have defined constraints:

$$\sum_{c \in L_{Event'}} u_{kc} = 1, \forall k \in Event \tag{6.4}$$

$$\sum_{\substack{i \in Entity \\ \langle i,k,j \rangle \in Triple}} \sum_{c \in L_{Entity}} u_{ic} = 1, \forall k \in Event \tag{6.5}$$

$$\sum_{\substack{j \in Entity, \\ \langle i,k,j \rangle \in Triple}} \sum_{c \in L_{Entity}} u_{jc} = 1, \forall k \in Event \tag{6.6}$$

where $L_{Event'}$ in Equation (6.4) is a subset of L_{Event}, consisting of $\{+effect, -effect\}$. Equation (6.4) means a +/-effect must be either +effect or -effect. But the ILP is free to choose whether it is being reversed. Recall that we have two agent candidates ($a1$,$a2$) for a +/-effect. Thus we have four agent indicators in Equation (6.5): $u_{a1,pos}$, $u_{a1,neg}$, $u_{a2,pos}$ and $u_{a2,neg}$. Equation (6.5) ensures that three of them are 0 and one of them is 1. For instance, $u_{a1,pos}$ assigned 1 means that candidate $a1$ is selected to be the agent span and *pos* is selected to be its polarity. In this way, the model disambiguates the agent span and sentiment polarity simultaneously. Similar comments apply for the theme candidates in Equation (6.6).

According to the sentiment inference rules in Section 5.1, the writer has the same sentiment toward entities in a +effect event. Thus, for each triple unit $\langle i,k,j \rangle$, this constraint is

applied via the following:

$$|\sum_{i,\langle i,k,j\rangle} u_{i,pos} - \sum_{j,\langle i,k,j\rangle} u_{j,pos}| + |u_{k,+effect} - u_{k,r}| <= 1 + \xi_{ikj}, \forall k \in Event \tag{6.7}$$

$$|\sum_{i,\langle i,k,j\rangle} u_{i,neg} - \sum_{j,\langle i,k,j\rangle} u_{j,neg}| + |u_{k,+effect} - u_{k,r}| <= 1 + \xi_{ikj}, \forall k \in Event \tag{6.8}$$

We use $|u_{k,+effect} - u_{k,r}|$ to represent whether this triple is +effect. In Equation (6.7), if this value is 1, then the triple should follow the +effect constraints. In that case, $\xi_{ikj} = 0$ means that the triple doesn't violate the +effect constraints, and $|\sum_i u_{i,pos} - \sum_j u_{j,pos}|$ must be 0. Further, in this case, $\sum_i u_{i,pos}$ and $\sum_j u_{j,pos}$ are constrained to be of the same value (both 1 or 0) – that is, entities i and j must be both positive or both not positive. However, if $\xi_{ikj} = 1$, Equation (6.7) does not constrain the values of the variables at all. If $|u_{k,+\text{effect}} - u_{k,r}|$ is 0, representing that the triple is not +effect, then Equation (6.7) does not constrain the values of the variables. Similar comments apply to Equation (6.8).

In contrast, as pointed out in Section 5.1, the writer has opposite sentiments toward entities in a -effect event.

$$|\sum_{i,\langle i,k,j\rangle} u_{i,pos} + \sum_{j,\langle i,k,j\rangle} u_{j,pos} - 1| + |u_{k,-effect} - u_{k,r}| <= 1 + \delta_{ikj}, \forall k \in Event \tag{6.9}$$

$$|\sum_{i,\langle i,k,j\rangle} u_{i,neg} + \sum_{j,\langle i,k,j\rangle} u_{j,neg} - 1| + |u_{k,-effect} - u_{k,r}| <= 1 + \delta_{ikj}, \forall k \in Event \tag{6.10}$$

We use $|u_{k,\text{-effect}} - u_{k,r}|$ to represent whether this triple is -effect. In Equation (6.9), if a triple is -effect and the constraints are not violated, then $|\sum_i u_{i,pos} + \sum_j u_{j,pos} - 1|$ must be 0. Further, in this case, $\sum_i u_{i,pos}$ and $\sum_j u_{j,pos}$ are constrained to be of the opposite value – that is, if entity i is positive then entity j must not be positive. Similar comments apply to Equation (6.10).

Note that above we use $|u_{k,+\text{effect}} - u_{k,r}|$ and $|u_{k,\text{-effect}} - u_{k,r}|$ to represent whether a triple is +effect or -effect. In Table 6.1, we show that they always take opposite values and that they are consistent with the actual effects. In Table 6.1, Case A means the triple is +effect

and Case B means the triple is -effect but it is reversed. In both cases, $|u_{+\text{effect}} - u_r| = 1$, indicating that the triple should follow the +effect constraints. Similarly for Case C and Case D to follow the -effect constraints.

	$u_{+\text{effect}}$	$u_{\text{-effect}}$	u_r	$\lvert u_{+\text{effect}} - u_r \rvert$	$\lvert u_{\text{-effect}} - u_r \rvert$
A	1	0	0	1	0
B	0	1	1	1	0
	$u_{+\text{effect}}$	$u_{\text{-effect}}$	u_r	$\lvert u_{+\text{effect}} - u_r \rvert$	$\lvert u_{\text{-effect}} - u_r \rvert$
C	0	1	0	0	1
D	1	0	1	0	1

Table 6.1: Truth table of being reversed or not (k is omitted)

6.1.1.1 Co-reference In the Model So far the constraints in the model are within a +/-effect triple. In order to bridge connections between different triples, we assume that if two entities in the same document co-refer to each other, the writer's sentiments toward the two entities should be the same. In addition, Consider the following example:

> (Ex6.1) **The reform** will decrease the healthcare costs and improve the medical qualify as expected.

The two +/-effect events, decrease and improve, have the same agent, reform. Thus, if there is more than one +/-effect in a sentence, and the path between the two +/-effects in dependency parse contains only *conj* or *xcomp* relations, and there is no other noun between the latter +/-effect event and the conjunction, we assume the two agents are the same and the sentiments toward them should be the same. Thus, for any $i, j \in \mathit{Entity}$, if i, j co-refer[2], or they are the same agent as described above, $\text{Coref}(i, j) = 1$ (otherwise 0). We add two more constraints, similar to the +effect constraints in Equations (6.7) and (6.8), as shown in Equation (6.11) and (6.12). where ν_{ij} is a slack variable, $e(i)$ is the set of agent/theme candidates linked to the same +/-effect as i is. If $Coref(i, j) = 0$, Equations (6.11) and

[2]We use the co-reference resolution system from (Stoyanov et al., 2010).

(6.12) do not constrain the variables. The objective function in Equation (6.13) is updated to incorporate these new constraints.

$$|\sum_{e(i)} u_{i,pos} - \sum_{e(j)} u_{j,pos}| + Coref(i,j) <= 1 + \nu_{ij}, \forall i,j \in Entity \tag{6.11}$$

$$|\sum_{e(i)} u_{i,neg} - \sum_{e(j)} u_{j,neg}| + Coref(i,j) <= 1 + \nu_{ij}, \forall i,j \in Entity \tag{6.12}$$

$$\min_{u_{1pos}, u_{1neg} \ldots} \left(-1 * \sum_{i \in Event \cup Entity} \sum_{c \in L_i} p_{ic} u_{ic} \right) + \sum_{\langle i,k,j \rangle \in Triple} \xi_{ikj} + \sum_{\langle i,k,j \rangle \in Triple} \delta_{ikj} + \sum_{i,j \in Entity} \nu_{ij} \tag{6.13}$$

6.1.2 Local Systems

We utilize various state-of-the-art systems and resources to extract candidates of each ambiguity and assign scores to each candidate, based on the local context and features.

+/-Effect Score $p_{k,\text{+effect}}, p_{k,\text{-effect}}$. We utilize a sense-level +/-effect lexicon by Choi et al. (2014). In total there are 6,622 +effect senses and 3,290 -effect senses. The +effect lexicon covers 64% of the +effect words in the corpus and the -effect lexicon covers 42% of the -effect words. We then look up the +/-effect span k in the +/-effect lexicon. If k only appears in the +effect lexicon, then $p_{k,\text{+effect}} = 1 - \epsilon$ and $p_{k,\text{-effect}} = \epsilon$. Here $\epsilon = 0.0001$, to prevent there being any 0 scores in our computation. If k only appears in the -effect lexicon, then $p_{k,\text{-effect}} = 1 - \epsilon$ and $p_{i,\text{+effect}} = \epsilon$. If k appears in both the +effect and -effect lexicon, and there are a senses in the +effect lexicon and b senses in the -effect lexicon, then $p_{k,\text{+effect}} = a/(a+b)$ and $p_{k,\text{-effect}} = b/(a+b)$. If k is not in either lexicon, then $p_{k,\text{+effect}} = p_{k,\text{-effect}} = \epsilon$. If there is more than one word in the +/-effect span, we take the maximum score.

Local Reversed Score $p_{k,r}$. As defined in Section 3.2.1, a reverser changes the effect of a +/-effect event. First, we build reverser lexicons from Wilson's shifter lexicon (Wilson, 2008), namely the entries labeled as *genshifter*, *negation*, and *shiftneg*. We create two lexicons: one with the verbs and the other with the non-verb entries, excluding nouns, adjectives,

and adverbs, since most non-verb reversers are prepositions or subordinating conjunctions. There are 219 reversers in the entire corpus; 134 (61.19%) are instances of words in one of the two lexicons. Based on the lexicon, we categorize reversers into three classes. Examples are shown below.

(Ex6.2) They will **not** be able to water down your coverage.
(Ex6.3) ... how a massive new bureaucracy will cut costs **without** hurting the old and the helpless.
(Ex6.4) The new law includes new rules to **prevent** insurance companies from overcharging patients.

Negation. An instance in this category is the word not in (Ex6.2). If any word in the +/-effect span has a *neg* dependency relation according to the Stanford dependency parser, then we consider the +/-effect to be negated (i.e., reversed). In this case the path between the negator and the +/-effect is labeled *neg* and the length of the path is one.

Other Non-Verb. This category consists of words such as without in (Ex6.3) (others are never and few, etc). These words lower the extent of the +/-effect event. We look in the sentence for instances of words in the non-verb reverser lexicon, which are not tagged as noun, verb, adj, or adv. For any found, we examine the path in the dependency parse between the potential reverser and the +/-effect span. If the path has at least one of *advmod*, *pcomp*, *cc*, *xcomp*, *nsubj*, *neg* and the length of the path is less than four (learnt from development set), the event is considered to be reversed.

Verb. In (Ex6.4), the verb (prevent) stops the +/-effect event (overcharging) from happening. We call such words *Verb* reverser (others are prohibit and ban, etc). We look in the sentence for instances of words in the verb reverser lexicon. For any that appear before the +/-effect span in the sentence, if the path has at least one of *xcomp*, *pcomp*, *obj* and the length of the path is less than four, then the event is reversed. For the triple ⟨companies, overcharging, patients⟩ in (Ex6.4), though it is reversed by a -effect event (prevent), the agent of the reverser (law), is different from the agent of the +/-effect (companies), so the -effect within the overcharging event is not reversed. Recall the triple chain we defined previously in Section 3.2.1. Here is a triple chain in (Ex6.4): ⟨law, prevent ⟨companies, overcharging, patients⟩ ⟩. The reverser is changing the effect between law and patients, but it does not change the polarity between companies and patients. Though we extract the *Verb* reversers

to evaluate the performance of recognizing a reverser, in the optimization model, +/-effect events with *Verb* reversers are not considered to be reversed, since almost all *Verb* reversers introduce new agents.

Different from other scores, $p_{k,r}$ could be negative. According to the heuristics above, the probability of a +/-effect event being reversed decreases as the length of the path increases. We define $p_{k,r}$ so it is inversely proportional to the length of the path. Further, to make sense of a +/-effect triple ⟨agent, +/-effect, theme⟩, where, e.g., the local detectors label it ⟨pos, -effect, pos⟩, the model is choosing the smaller one from (a) $-1 * p_{k,r} * u_{k,r}$ (it has a reverser) versus (b) $1 * \xi_{ikj}$ (it is an exception to the rules). The model assigns $u_{k,r} = 0$ and $\xi_{ikj} = 1$ if $-1 * p_{k,r} > 1$. It assigns $u_{k,r} = 1$ and $\xi_{ikj} = 0$ if $-1 * p_{k,r} <= 1$. For +/-effect events which have *Negation* or *Other Non-verb* reversers, since we use length four as a threshold in the heuristics above, we define $p_{k,r} = \frac{1}{d} - \frac{5}{4}$, so that $-1 * p_{k,r} = \frac{5}{4} - \frac{1}{d} > 1$ if $d > 4$. For +/-effect events for which no reverser word appears in the sentence, or those which only have *Verb* reversers, $p_{k,r} = -1 * \frac{5}{4}$ (so $-1 * p_{k,r} > 1$), so that the model chooses case (b) (choosing the +/-effect event to be not reversed). The probability of a +/-effect event being reversed depends on the dependency path. The longer the path, the less likely the effect is reversed.

Local Sentiment Score $p_{i,pos}, p_{i,neg}$**.** In the +/-Effect Event Sentiment Corpus developed in Section 4.1, only the writer's sentiments toward the agents and the themes of +/-effect events are annotated. Thus, since there are many false negatives of sentiments toward entities, the corpus does not support training a classifier. Therefore, we use available resources to detect writer's sentiments toward all agent and theme candidates. We use Opinion Extractor (Johansson and Moschitti, 2013b), opinionFinder (Wilson et al., 2005), MPQA subjectivity lexicon (Wiebe et al., 2005), General Inquirer (Stone et al., 1966) and a connotation lexicon (Feng et al., 2013), to detect writer's sentiments toward all agent and theme candidates, and all +/-effect events. If there is a sentiment toward +/-effect event detected, we use the inference rules to infer from the sentiment toward event to the sentiment toward the theme. Then we conduct a majority voting based on the results. The sentiment scores range from 0.5 to 1.

6.1.3 Experiment and Result

6.1.3.1 Experiment Data We use the +/-Effect Event Sentiment Corpus developed in Section 4.1 (Deng et al., 2013), consisting of 134 online editorials and blogs. In total, there are 1,762 annotated triples, out of which 692 are +effect or retainers and 1,070 are -effect or reversers. From the writer's perspective, 1,495 noun phrases are annotated positive, 1,114 noun phrases are negative and the remaining 8 are neutral. This indicates that there are many opinions in the corpus. Out of 134 documents in the corpus, 3 do not have any annotation. 6 are used as a development set to develop the heuristics in Sections 6.1.2 and 6.1.1.1. We use the remaining 125 for the experiments.

6.1.3.2 Baseline Methods and Evaluation Metrics We compare the output of the global optimization model with the outputs of baseline systems built from the local detectors in Section 6.1.2. For the +/-effect of events and reverser ambiguities, the local detectors directly provide a disambiguation result. For the agent/theme span and sentiment ambiguities, the local sentiment detector assigns positive and negative scores to each candidate. The model chooses among the combined options. For example, we extract two agent candidates a_1 and a_2 of a +/-effect event, the model will assign the value 1 to only one from the four indicator variables $u_{a1,pos}$, $u_{a1,neg}$, $u_{a2,pos}$ and $u_{a2,neg}$. In this way, the model determines both the agent span and the sentiment toward the agent. Thus, for comparison, we build a baseline system that combines the outputs of the local agent/theme candidate detector and the local sentiment detector.

Recall from Section 6.1.2, a variable $k \in$ *Event* has two agent candidates, $a1$ and $a2$ $\in$ *Entity*. Together there are four binary indicator variables: $u_{a1,pos}$, $u_{a1,neg}$, $u_{a2,pos}$ and $u_{a2,neg}$. Among these indicator variables whose corresponding local scores (e.g., $p_{a1,pos}$ is the score of $u_{a1,pos}$) are larger than 0.5, the baseline system (denoted *Local*) chooses the one with the largest local sentiment score. If there is a tie, it prefers the variable representing the semantic candidate. If there is still a tie, it chooses the variable representing the majority polarity (positive). If all the local scores of the four variables are 0.5 (neutral), *Local* fails to recognize any sentiment for that entity, so it assigns 0 to all the indicator variables.

Local+coref takes the maximum local score of the entities if they co-ref, and assigns each entity the maximum score before disambiguation.

Another baseline, *Majority*, always chooses the semantic candidate and the majority polarity.

To evaluate the performance in detecting sentiment, we use precision, recall, and F-measure. We do not take into account any agent or theme manually annotated as neutral (there are only 8).

$$\mathrm{P} = \frac{\#(\text{auto=gold \& gold!=neutral})}{\#\text{auto!=neutral}} \tag{6.14}$$

$$\text{Accuracy} = \mathrm{R} = \frac{\#(\text{auto=gold \& gold!=neutral})}{\#\text{gold!=neutral}} \tag{6.15}$$

$$\mathrm{F} = \frac{2{*}\mathrm{P}{*}\mathrm{R}}{\mathrm{P}+\mathrm{R}} \tag{6.16}$$

In the equations, *auto* is the system's output and *gold* is the gold-standard label from annotations. Since we don't take into account any *neutral* agent or theme, *#gold!=neutral* equals the number of nodes in the experiment set. Thus accuracy is equal to recall. We only report recall here. Here we have two definitions of *auto=gold*: (1) **Strict** evaluation means that, by saying *auto=gold*, the agent/theme must have the same polarity and must be the same NP as the gold standard, and (2) **Relaxed** evaluation means the agent/theme has the same polarity as the gold standard, regardless whether the span is correct or not.

Note that according to the annotations in the corpus, an implicit agent isn't annotated with any sentiment. Thus, for an implicit agent in *gold*, if *auto* outputs the span "implicit", we treat it as a correct span with correct polarity, regardless what sentiment *auto* gives to it. If *auto* outputs any span other than "implicit", we treat it as a wrong span with the wrong polarity, regardless of its sentiment as well. For the theme span, if *auto* outputs a "null" theme candidate, we treat it as a wrong span but we evaluate its sentiment according to *gold*.

To evaluate extracting candidate spans, we use accuracy. The baseline for this task always chooses the semantic candidate. To evaluate +/-effect polarity and reverser, we also

use accuracy.

Note that although we evaluate the performance in different tasks separately, the model resolves all the ambiguities at the same time.

6.1.3.3 Results We report the performance results for **(A) sentiment detection** in Table 6.2, on two sets. One is the subset containing the agents and themes where *auto* has the correct spans with *gold*. The other is the set of all agents and themes. As shown in Table 6.2, *ILP* significantly improves performance, approximately 10-20 points on F-measure over different baselines. Though *Local* has a competitive precision with *ILP*, it has a much lower recall. That means the local sentiment detector cannot recognize implicit sentiments toward most entities. But *ILP* is able to recognize more entities correctly. By adding *coref*, performance improves for both *ILP* and *Local*.

		correct span subset			whole set, strict eval			whole set, relaxed eval		
		P	R	F	P	R	F	P	R	F
1	ILP	0.6421	0.6421	0.6421	0.4401	0.4401	0.4401	0.5939	0.5939	0.5939
2	Local	0.6409	0.3332	0.4384	0.4956	0.2891	0.3652	0.5983	0.3490	0.4408
3	ILP+coref	0.6945	0.6945	0.6945	0.4660	0.4660	0.4660	0.6471	0.6471	0.6471
4	Local+coref	0.6575	0.3631	0.4678	0.5025	0.3103	0.3836	0.6210	0.3834	0.4741
5	Majority	0.5792	0.5792	0.5792	0.3862	0.3862	0.3862	0.5462	0.5462	0.5462

Table 6.2: Performances of sentiment detection

In terms of the other tasks: For **(B) agent/theme span**, the baseline achieves 66.67% in accuracy, compared to 68.54% and 67.10% for *ILP* and *ILP+coref*, respectively. For **(C) +/-effect polarity**, the baseline has an accuracy of 70.68%, whereas *ILP* achieves 77.25% and *ILP+coref* achieves 77.47%, respectively, both 7 points higher. This improvement is interesting because it represents cases in which the optimization model is able to ***infer*** the correct polarity even though the +/-effect span is not recognized by the local detector (i.e., the span isn't in the +/-effect lexicon). For **(D) reverser**, the baseline is 88.07% in accuracy. *ILP* and *ILP+coref* are competitive with the baseline: 89% and 88.07% respectively. Note

that both our local detector and *ILP* surpass the majority class (not reversed) which has an accuracy of 86.60%.

Following Akkaya et al. (2009), since ILP is unsupervised without multiple runs, we adopt McNemar's test to measure statistical significance of our improvements (Dietterich, 1998). In Table 6.2, the improvements in recalls of Line 1 over 2, Line 3 over 4, and Lines 1&3 over 5 are statistically significant at the $p < .001$ level. The improvements of Line 3 over 1 are statistically significant at the $p < .005$ level. For accuracy of +/-effect polarity, the improvement is significant at the $p < .001$ level.

6.1.4 Examples

This section gives simplified examples to illustrate how the model can improve over the local detectors. The explicit sentiment clues referred to in this section are from MPQA lexicon.

(Ex6.5) The reform would curb skyrocketing costs in the long run.

The local sentiment detector assigns a negative label to the costs due to the single sentiment clue, skyrocketing. Since the agent and theme are in a -effect triple, and the writer is negative toward that theme, we can infer the writer is positive toward the agent. This illustrates how we improve recall on sentiments.

(Ex6.6) The supposedly costly reform will curb skyrocketing costs in the long run.

In (Ex6.6), the agent reform is labeled negative because the word costly is a negative clue in the lexicon. (The word supposedly is not in the lexicon.) However, in (Ex6.6), it is actually positive. The agent's negative score is 0.6, and its positive score is 0.5 due to the absence of a positive clue. Since the theme is labelled negative too, by the -effect constraints, we expect to see a positive agent. If we were to assign negative to the agent, the objective function would have -0.6 subjectivity score and +1 in violation penalty, together giving +0.4. If we assign positive, the subjectivity score is -0.5, and there is no violation, resulting in a total score of -0.5. Thus the model correctly chooses the positive label. This shows how we can improve precision on sentiments.

(Ex6.7) The great reform will curb skyrocketing costs in the long run.

In this case, the agent is positive and the theme is negative. If the +/-effect word, curb, is not in the lexicon, we could still infer its effect. Given that the entities in the triple have different sentiments, to not violate the implicature rules, the model will assign it the label -effect, or assign it two labels, +effect along with reversed. However, there is no reverser word in the sentence, so the reversed score $p_r = -\frac{5}{4}$. The model will assign the reverser indicator $u_r = 0$, in order to avoid a gain in the objective function by $-1 * p_r * u_r$. Thus the model assigns the label -effect to curb. This is how the model can improve the accuracy of disambiguating the +/-effect sense.

6.2 ENTITY/EVENT-LEVEL SENTIMENT INFERENCE

At the beginning at this chapter, we decomposed the whole task into three subtasks. In each subtask, there are different ambiguities. In the previous section, we have utilized an Integer Linear Programming model, in which the candidates of each ambiguity are detected by local systems. Then the ILP model simultaneously selects the best candidate of all the ambiguities and jointly resolves all the ambiguities. The experiment has shown that the joint model is better than individual systems. Thus, in this chapter, we follow the same procedure but the task in this chapter is much more complicated. The model in this chapter aims at recognizing the sentiments expressed from any entity or the writer toward any entity or event in the sentences.

In this chapter, we first introduce the Probabilistic Soft Logic (PSL) model in Section 6.2.1. We do not continue using ILP as in the previous section. One limitation of the ILP model is that in the objective function, the constraints are expressed in mathematical equations. Recall Equation (6.7). The constraints need to be specifically designed, and do not have a one-to-one correspondence to the propositional rules. On the contrary, the constraints of PSL are propositional rules. This enables us to use the inference rules directly as constraints which are intuitive to understand. We present the local systems developed to assign local scores in Section 6.2.2. Finally we will talk about the experiment result in Section 6.2.3.

6.2.1 Probabilistic Soft Logic

We are pursing such a model that combines the probabilistic calculation of many ambiguities under the constraints of the dependencies of the data, defined by inference rules in the propositional logic form. Every candidate of every ambiguity is represented as a variable in the joint model. The goal is to find an optimal configuration of all the variables, so the ambiguities are solved. Our previous study in Section 6.1 (Deng et al., 2014) and many previous works in various applications of NLP (Roth and Yih, 2004; Punyakanok et al., 2004, 2008; Das et al., 2012; Choi et al., 2006; Yang and Cardie, 2013a; Denis and Baldridge, 2007; Martins and Smith, 2009; Somasundaran and Wiebe, 2009), have used Integer Linear Programming (ILP) as a joint model to do so, by setting the dependencies as constraints in the ILP framework. However, the constraints in ILP are linear equations and inequations. In order to choose a framework that computes propositional logics more directly, we use the Markov Logic Networks (MLNs) (Richardson and Domingos, 2006) and their variations as an appropriate way to solve our problem.

The MLN is a framework for probabilistic logic that employs weighted formulas in propositional logic to compactly encode complex undirected probabilistic graphical models (i.e., Markov networks) (Beltagy et al., 2014). An MLN model is defined using a set of atoms to be grounded, and a set of weighted if-then rules expressed in first-order logic. For example, we define the atom TARGET(y,t) to represent an opinion y having the entity/event-target t. If y and t are constants, then TARGET(y,t) is a ground atom (e.g., TARGET(insulting, Prophet)). Each ground atom is assigned a score by a local system. MLN takes as input all the local scores as well as the constraints defined by the rules among atoms, so that it is able to jointly resolve all the ambiguities. In the final output, for example, the score TARGET(insulting, Prophet) = 1 means that MLN considers Prophet to be an entity/event-target of the insulting event, while TARGET(insulting, countries) = 0 means that MLN does not consider the entity countries to be an entity/event-target of the insulting event. The goal of the MLN is to find an optimal grounding which maximizes the values of all the satisfied propositional logic formula in the knowledge base (Richardson and Domingos, 2006). Each rule in the MLN is associated with a weight, indicating the importance of this rule in the

whole rule set. The weights can be learnt.

In this chapter, we use a variation of MLN, which is Probabilistic Soft Logics (PSL), A key feature of PSL is that each ground atom a has a soft, continuous truth value in the interval [0, 1], denoted as $I(a)$, rather than a binary truth value as in MLN and most other probabilistic logic frameworks (Beltagy et al., 2014). To compute the soft truth values for logical formulas, Lukasiewicz relaxations (Klir and Yuan, 1995) are used:

$$l_1 \wedge l_2 = \max\{0, I(l_1) + I(l_2) - 1\} \tag{6.17}$$

$$l_1 \vee l_2 = \min\{I(l_1) + I(l_2), 1\} \tag{6.18}$$

$$\neg l_1 = 1 - I(l_1) \tag{6.19}$$

A rule $r \equiv r_{body} \rightarrow r_{head}$ is satisfied (i.e. $I(r) = 1$) iff $I(r_{body}) \leq I(r_{head})$. Otherwise, a distance to satisfaction $d(r)$ is calculated, which defines how far a rule r is from being satisfied: $d(r) = \max\{0, I(r_{body}) - I(r_{head})\}$. Using $d(r)$, PSL defines a probability distribution over all possible interpretations I of all ground atoms:

$$p(I) = \frac{1}{Z} \exp\{-1 \times \sum_{r \in R} \lambda_r (d(r))^g\} \tag{6.20}$$

where Z is the normalization constant, λ_r is the weight of rule r, R is the set of all rules, and g defines the MPE inference (Most Probable Explanation) used in PSL. If $g = 1$, the inference is comparable to inferences in MLN. If $g = 2$, MPE inference can be shown to be a second-order cone program (SOCP) (Beltagy et al., 2014). PSL seeks the interpretation with the minimum distance $d(r)$ and which satisfies all rules to the extent possible.

Though MLN and PSL use propositional logics, they can be applied to the propositional logics in our works, since the proposition logics do not use quantifiers. Previously in the works of PSL (Broecheler et al., 2010) each symbol is denoted as *atom*, while in this thesis we name each symbol as *literal*. In the rest of this chapter, we use the term *literal* instead of the word *atom* in the literature.

6.2.1.1 PSL for Entity-Event-level Sentiment Analysis.

We have built three PSL models for entity/event-level sentiment analysis, depending on the rules used in the models.

PSL1 only uses the rules introduced in Section 5.2 (repeated below) to aggregate various opinions extracted by phrase-level systems into positive pairs and negative pairs. Thus, under our representations, the PSL model not only finds a set of entity/event-targets of an opinion, but also represents the aggregated sentiments among entities and events (positive pairs and negative pairs) in the sentence.

$$+\textsc{Sentiment}(y) \wedge \textsc{Source}(y,s) \wedge \textsc{Target}(y,t) \Rightarrow \textsc{PositivePair}(s,t)$$
$$-\textsc{Sentiment}(y) \wedge \textsc{Source}(y,s) \wedge \textsc{Target}(y,t) \Rightarrow \textsc{NegativePair}(s,t)$$

PSL2 uses the sentiment inference rules defined in Section 5.2, in addition to the rules used in PSL1. PSL2 aims at relying on the *sentiment toward sentiment* structure to infer more sentiments.

PSL3 uses the sentiment inference rules about inferring the sentiments toward +/-effect events and the entities in the events defined in Section 5.1, in addition to the rules used in PSL2. PSL3 uses the full set of rules. It aims at using both the sentiment toward sentiment structure and the +/-effect event information to infer more sentiments.

6.2.2 Local Systems

In this section, we introduce the local systems to extract candidates of ambiguities in each subtask. Generally speaking, there are two tasks. The first task is recognizing the explicit entity/event-level sentiments (Section 6.2.2.1). The second task is recognizing the +/-effect event information (Section 6.2.2.2).

6.2.2.1 Explicit Sentiments.

Instead of building an entity/event-level sentiment system from scratch, we propose to fully utilize off-the-shelf tools and resources for extracting opinions, the sources, and the targets (Yang and Cardie, 2013a, 2014; Johansson and Moschitti, 2013a; Riloff et al., 2013a; Xu et al., 2013; Liu et al., 2014; Tang et al., 2014; Liu et al., 2013; Irsoy and Cardie, 2014; Scholz and Conrad, 2013; Zhou et al., 2013). Some of the resources and tools extract the opinion expressions, the opinion polarities, the opinion

sources, and the opinion targets. Some of the resources and tools only extract the opinion expressions and the polarities. Moreover, the sources and targets extracted by off-the-shelf tools are usually span-based. We take the union of all the explicit opinions and the components of the opinions that state-of-art systems extract and use them as a basis for our sentiment inference.

Specifically, the local systems should assign local scores to the following literals.

+Sentiment(y)
-Sentiment(y)
Source(y,s)
Target(y,t)

+SENTIMENT(y) and -SENTIMENT(y). We build upon three span-based sentiment analysis systems. The first, S1 (Yang and Cardie, 2013a), and the second, S2 (Yang and Cardie, 2014), are both trained on MPQA 2.0, which does not contain any entity/event-target annotations. S1 extracts triples of ⟨source span, opinion span, span-target span⟩, but does not extract opinion polarities. S2 extracts opinion spans and opinion polarities, but it does not extract sources or targets. The third system, S3 (Socher et al., 2013), is trained on movie review data. It extracts opinion spans and polarities. The source is always assumed to be the writer.

We take the union of opinions extracted by S1, S2, and S3. For each opinion y, a ground atom is created, depending on the polarity (+SENTIMENT(y) if y is positive and -SENTIMENT(y) is y is negative). The polarity is determined as follows. If S2 assigns a polarity to y, then that polarity is used. If S3 but not S2 assigns a polarity to y, then S3's polarity is used. In both cases, the score assigned to the ground atom is 1.0. If neither S2 nor S3 assigns a polarity to y, we use the MPQA subjectivity lexicon to determine its polarity. The score assigned to the ground atom is the proportion of the words in the opinion span that are included in the subjectivity lexicon.

TARGET(y,t). Though each entity/event-target is an entity or event, it is difficult to determine which nouns and verbs should be considered. Taking into consideration the trade-off between precision and recall, we experimented with three methods to select entity/event-target candidates. For each opinion y, a ground atom TARGET(y,t) is created for each

entity/event-target candidate t.

ET1 considers all the nouns and verbs in the sentence, to provide high recall of entity/event-targets.

ET2 considers all the nouns and verbs in span-target spans and opinion spans that are automatically extracted by systems S1, S2, and S3. We hypothesized that ET2 would be useful because most of the entity/event-targets in MPQA 3.0 appear within the opinion or the span-target spans of MPQA 2.0.

ET3 considers the heads of the span-target and opinion spans that are automatically extracted by systems S1, S2 and S3,[3] and also considers the heads of the spans that have the same parent node on the syntax tree with the span-target or opinion spans ("siblings"). Among the three methods, ET3 has the lowest recall but the highest precision.

In addition, for the entity/event-target candidate set extracted by ET2, or ET3, we run the Stanford co-reference system (Manning et al., 2014; Recasens et al., 2013; Lee et al., 2013) to expand the set in two ways. First, for each entity/event-target candidate t, the co-reference system extracts the entities that co-refer with t. We add the referring entities into the candidate set. Second, the co-reference system extracts words which the Stanford system judges to be entities, regardless of whether they have any referent or not. We add this set of entities to the candidate set as well.

We train an SVM classifier (Cortes and Vapnik, 1995) to assign a score to the ground atom TARGET(y,t). Syntactic features describing the relations between an entity/event-target and the extracted opinion span and span-target span are considered, including: (1) whether the entity/event-target is in the opinion/span-target span; (2) the unigrams and bigrams on the path from the entity/event-target to the opinion/span-target span in the constituency parse tree; and (3) the unigrams and bigrams on the path from the entity/event-target to the opinion/span-target word in the dependency parse graph. We normalize the SVM scores to the range of a ground atom score, [0,1].

SOURCE(y,s). S1 extracts the source of each opinion, S2 does not extract the source, and S3 assumes the source is always the writer. Thus, for an opinion y, if the source s is assigned

[3]The head of a phrase is extracted by the Collins head finder in the Stanford parser (Manning et al., 2014).

by S1, a ground atom SOURCE(y,s) is created with score 1.0. Otherwise, if S3 extracts opinion y, a ground atom SOURCE(y,writer) is created with score 1.0 (since S3 assumes the source is always the writer). Otherwise, we run the Stanford named entity recognizer (Manning et al., 2014; Finkel et al., 2005) to extract named entities in the sentence. The nearest named entity to the opinion span on the dependency parse graph will be treated as the source. The score is the reciprocal of the length of the path between the opinion span and the source span in the dependency parse.

6.2.2.2 +/-Effect Events. We follow the individual systems developed in Section 6.1 to extract candidates of the ambiguities caused by +/-effect events, corresponding to the blue circle in Figure , including (1.1) Which spans are +/-effect events? (1.2) Which noun phrases are the agents, which are the themes? (1.3) What is the effect of the +/-effect event? (1.4) Is the effect reversed? These questions correspond to assigning local scores to the literals listed below.

+Effect(x)
-Effect(x)
Agent(x, a)
Theme(x, h)

+EFFECT(x) and -EFFECT(x). We use the +/-effect sense-level lexicon (Choi and Wiebe, 2014)[4] to extract the +/-effect events in each sentence. The score of +EFFECT(x) is the fraction of that word's senses that are +effect senses according to the lexicon, and the score of -EFFECT(x) is the fraction of that word's senses that are -effect senses according to the lexicon. If a word does not appear in the lexicon, we do not treat it as a +/-effect event, and thus assign 0 to both +EFFECT(x) and -EFFECT(x).

AGENT(x,a) and THEME(x,h): We consider all nouns in the same or in sibling constituents of a +/-effect event as potential agents or themes. An SVM classifier is run to assign scores to AGENT(x,a), and another SVM classifier is run to assign scores to THEME(x,h). Both SVM classifiers are trained on a separate corpus, the +/-Effect Event Sentiment Corpus (Deng et al., 2013) used in (Deng et al., 2014), which is annotated with +/-effect event,

[4]Available at: http://mpqa.cs.pitt.edu/lexicons/effect_lexicon/

agent, and theme spans. The features we use to train the agent and theme classifier include unigram, bigram, and syntax information.

Finally, to support new inferences, more groundings of TARGET(y,t) are defined in PSL3. For a +/-effect event x (both +effect events and -effect events) whose agent is a, if one of x and a is an entity/event-target candidate of y, the other will be added to the entity/event-target candidate set for y (sentiments toward both +effect and -effect events and their agents have the same polarity according to the rules (Deng et al., 2014)). For a +effect event x (only +effect events) whose theme is h, if one of x and h is an entity/event-target candidate of y, the other is added to the entity/event-target candidate set for y (sentiments toward +effect events and their themes have the same polarity).

6.2.3 Experiment and Results

We carry out experiments on the MPQA 3.0 corpus developed in Section 4.2. Currently, there are 70 documents, 1,634 sentences, and 1,921 DS and ESEs in total. The total number of POSPAIR(s,t) and NEGPAIR(s,t) are 867 and 1,975, respectively. Though the PSL inference does not need supervision and the SVM classifier for agents and themes in Section 6.2.2.2 is trained on a separate corpus, we still have to train the entity/event-target SVM classifier to assign local scores as described in Section 6.2.2.1. Thus, the experiments are carried out using 5-fold cross validation. For each fold test set, the entity/event-target classifier is trained on the other folds. The trained classifier is then run on the test set, and PSL inference is carried out on the test set.

In total, we have three methods for entity/event-target candidate selection (ET1, ET2, ET3) and three models for sentiment analysis (PSL1, PSL2, PSL3).

6.2.3.1 Baselines. Since each noun and verb may be an entity/event-target, the first baseline (All NP/VP) regards all the nouns and verbs as entity/event-targets. The first baseline estimates the difficulty of this task.

The second baseline (SVM) uses the SVM local classification results from Section 6.2.2.1. The score of TARGET(y,t) is assigned by the SVM classifier. Then it is normalized as input

into PSL. Before normalization, if the score assigned by the SVM classifier is above 0, the SVM baseline considers it as a correct entity/event-target.

6.2.3.2 Evaluations. First, we examine the performance of the PSL models on correctly recognizing entity/event-targets of a particular opinion. This evaluation is carried out on a subset of the corpus: we only examine the opinions which are automatically extracted by the span-based systems (S1, S2 and S3). If an opinion expression in the gold standard is not extracted by any span-based system, it is not input into PSL, so PSL cannot possibly find its entity/event-targets.

The second and third evaluations assess performance of the PSL models on correctly extracting all the positive and negative pairs. Note that our sentiment analysis system has the capability, through inference, to recognize positive and negative pairs even if corresponding opinion expressions are not extracted. Thus, the second and third evaluations are carried out on the entire corpus. The second evaluation uses ET3, and compares PSL1, PSL2, and PSL3. The third evaluation uses PSL3 and compares performance using ET1, ET2, and ET3. The results for the other combinations follow the same trends.

Entity/event-targets of An Opinion. According to the gold standard in Section 4.2, each opinion has a set of entity/event-targets. But not all entity/event-targets are equally important. Thus, our first evaluation assesses the performance of extracting the most important entity/event-target. As introduced in Section 4.2, a span-based target annotation of an opinion in MPQA 2.0 captures the most important target this opinion is expressed toward. Thus, the head of the span-target span can be considered to be the most important entity/event-target of an opinion. We model this as a ranking problem to compare models. For an opinion y automatically extracted by a span-based system, both the SVM baseline and PSL assign scores to $\textsc{Target}(y,t)$. We rank the entity/event-targets according to the scores. Because the ALL NP/VP baseline does not assign scores to the nouns and verbs, we do not compare with that baseline in this ranking experiment. We use the Precision@N evaluation metric. If the top N entity/event-targets of an opinion contain the head of a span-target span, we consider it as a correct hit. The results are in Table 6.3.

Table 6.3 shows that the SVM is poor at ranking the most important entity/event-target.

	Prec@1	Prec@3	Prec@5
SVM	0.0370	0.0556	0.0820
PSL1	0.5105	0.6905	0.7831
PSL2	0.5317	0.7486	0.7883
PSL3	0.5503	0.7434	0.8148

Table 6.3: Precision@N of most important entity/event-target.

The PSL models are much better, even PSL1, which does not include any inference rules. This shows that SVM, which only uses local features, cannot distinguish the most important entity/event-target from the others. But the PSL models consider all the opinions, and can recognize a true negative even if it ranks high in the local results. The ability of PSL to rule out true negative candidates will be repeatedly shown in the later evaluations.

We not only evaluate the ability to recognize the most important entity/event-target of a particular opinion, we also evaluate the ability to extract all the entity/event-targets of that opinion. The F-measure of SVM is 0.2043, while the F-measures of PSL1, PSL2 and PSL3 are 0.3135, 0.3239, and 0.3275, respectively. Correctly recognizing all the entity/event-targets is difficult, but all the PSL models are better than the baseline.

Positive Pairs and Negative Pairs. Now we evaluate the performance in a stricter way. We compare automatically extracted sets of sentiment pairs: $P_{\text{auto}} = \{\textsc{PosPair}(s,t) > 0\}$ and $N_{\text{auto}} = \{\textsc{NegPair}(s,t) > 0\}$, against the gold standard sets P_{gold} and N_{gold}. Table 6.2.3.2 shows the accuracies using ET3. Note that higher accuracies can be achieved, as shown later. Here we use ET3 just to show the trend of results.

As shown in Table 6.2.3.2, the low accuracy of the baseline All NP/VP shows that entity/event-level sentiment analysis is a difficult task. Even the SVM baseline does not have good accuracy. Note that the SVM baseline in Table 6.2.3.2 uses ET3. The baseline classifies the heads of span-target spans and opinion spans, which are extracted by state-of-the-art span-based sentiment analysis systems. This shows the results from span-based sentiment analysis systems do not provide enough accurate information for the more fine-

	PositivePair	NegativePair
All NP/VP	0.1280	0.1654
SVM	0.0765	0.0670
PSL1	0.3356	0.3754
PSL2	0.3705	0.3705
PSL3	0.4315	0.3892

Table 6.4: Accuracy comparing PSL models (ET3 used for all)

grained entity/event-level sentiment analysis task. In contrast, PSL1 achieves much higher accuracy than the baselines. PSL2 and PSL3, which add sentiment toward sentiment and +/-effect event inferences, give further improvements. A reason is that SVM uses a hard constraint to cut off many entity/event-target candidates, while the PSL models take the scores as soft constraints.

Another reason is due to the definition of accuracy: (TruePositive+TrueNegative)/All. A significant benefit of using PSL is correctly recognizing true negative entity/event-target candidates and eliminating them from the set. Interestingly, even though both PSL2 and PSL3 introduce more entity/event-target candidates, both are able to recognize more true negatives and improve the accuracy.

Note that F-measure does not count true negatives. Precision is $\frac{TP}{TP+FP}$, and recall is $\frac{TP}{TP+FN}$; neither considers true negatives (TN). As shown in Table 6.2.3.2, the increment of the PSL models over baselines on F-measure is not as large as the increase in accuracy. Comparing PSL2 and PSL3 to PSL1, the inference rules largely increase recall but lower precision. However, the accuracy in Table 6.2.3.2 keeps growing. Thus, the biggest advantage of PSL models is to correctly rule out true negative entity/event-targets. For the baselines, though the SVM baseline has higher precision, it eliminates so many entity/event-target candidates that the F-measure is not high.

Entity/event-target Selection. To assess the methods for entity/event-target selec-

	Precision	Recall	F-measure	Precision	Recall	F-measure
	PositivePair			NegativePair		
All NP/VP	0.1481	0.4857	0.2270	0.1824	0.6408	0.2840
SVM	0.3791	0.0870	0.1415	0.3568	0.0761	0.1254
PSL1	0.2234	0.2687	0.2440	0.2857	0.3872	0.3288
PSL2	0.1666	0.2738	0.2072	0.2772	0.3883	0.3235
PSL3	0.1659	0.3523	0.2256	0.2586	0.4529	0.3292

Table 6.5: F-measure comparing PSL models (ET3 used for all)

tion, we run PSL3 (the fullest PSL model) using each method in turn. The F-measures and accuracies are listed in Table 6.6. The F-measure of ET1 is slightly lower than the F-measures of ET2 and ET3, while the accuracy of ET1 is much better than the accuracies of ET2 and ET3. Again, this is because PSL recognizes true negatives in the entity/event-target candidates. Since ET1 considers more entity/event-target candidates, ET1 gives PSL a greater opportunity to remove true negatives, leading to an overall increase in accuracy.

	PosPair		NegPair	
	F	Acc.	F	Acc.
ET1	0.2192	0.4963	0.3157	0.4461
ET2	0.2374	0.4433	0.3261	0.3969
ET3	0.2256	0.4315	0.3295	0.3892

Table 6.6: Comparison of entity/event-target selection methods (PSL3 used for all)

In summary, the experiments have shown that the PSL model defined by the sentiment inference rules is able to improve entity/event-level sentiment recognition. The PSL model is good at recognizing the most important entity/event-target of an opinion. What's more important, the joint model is able to resolve various ambiguities of different NLP tasks

simultaneously in one model.

6.3 RELATED WORK

Different from pipeline architectures, where each step is computed independently, joint prediction has often achieved better results. Roth and Yih (2004) formulate the task of information extraction using Integer Linear Programming (ILP). Since then, ILP has been widely used in various tasks in NLP, including semantic role labeling (Punyakanok et al., 2004, 2008; Das et al., 2012), joint extraction of opinion entities and relations (Choi et al., 2006; Yang and Cardie, 2013a), co-reference resolution (Denis and Baldridge, 2007), and summarization (Martins and Smith, 2009). The most similar ILP model to ours is (Somasundaran and Wiebe, 2009), which improves opinion polarity classification using discourse constraints in an ILP model. However, their works address discourse relations among explicit opinions in different sentences. In terms of Probabilistic Soft Logic (PSL), it is a new statistical relational learning method that has been applied to many NLP and other machine learning tasks in recent years (Beltagy et al., 2014; London et al., 2013; Pujara et al., 2013; Bach et al., 2013; Huang et al., 2012, 2013; Memory et al., 2012). Previously, PSL has not been applied to entity/event-level sentiment analysis.

6.4 SUMMARY

The ultimate goal of this thesis is to utilize +/-effect event information to improve detection of the sentiments expressed among entities and events mentioned in the text. In this chapter, we have presented the computational models to achieve the goal. The models follow the representation scheme defined in Chapter 3. We use the inference rules defined in Chapter 4, and carry out experiments on the corpora developed in Chapter 5. We mainly investigate computational models for two tasks in this chapter. First, as a pilot study, we focus on recognizing the writer's sentiments expressed toward the agents and themes of +/-effect

events in Section 6.1. The experiments have demonstrated that our model improves over local sentiment recognition by almost 20 points in F-measure and over all sentiment baselines by over 10 points in F-measure. The success of the pilot study encourages us to extend the model. Next in Section 6.2, we focus on a more complicated task that is recognizing the sentiments whose sources can be the writer or any entity in the text and the targets are entities or events. The model builds upon state-of-the-art span-based sentiment analysis systems to perform entity/event-level sentiment analysis covering both explicit and implicit sentiments expressed among entities and events in text. The experiments have shown that the model jointly disambiguate the ambiguities in the opinion components and improve over baseline accuracies in recognizing entity/event-level sentiments.

The computational models in both tasks are joint prediction models. The input to the joint models is the result of individual NLP tasks, and the output of the joint models are answers to each individual task which are globally optimized. For example, the joint model in Section 6.1 infers the effect of the +-effect event, whether or not it are reversed, which candidate noun phrases are the agent and theme, and the writer's sentiments toward them. The joint models are constrained by the inference rules. As experiments have shown, in addition to beating the baselines for sentiment detection, the joint model in Section 6.1 significantly improves the accuracy of +/-effect sense disambiguation without any loss in accuracy of the remaining tasks. The exciting conclusions drawn from this chapter include: 1) the inference rules define useful dependencies among different NLP tasks; 2) the joint prediction models defined by the inference rules promise to jointly resolve the ambiguities in different NLP tasks and improve the performance in more than one task.

7.0 RECOGNIZING SOURCES OF OPINIONS

As discussed in the previous sections, a joint model is able to improve each subtask implemented in the joint model, on the condition that the constraints of the joint model are properly defined to build connections of different subtasks. Though the performances of the joint models are better than the baselines, the performances in entity/event-level sentiment analysis are not good enough. One of the reasons is due to the performances of local systems, which provide input to the joint prediction models. A better local system provides a better basis for the joint model to improve the performances. Thus, we will discuss our works in improving the recognitions of sources of opinions. A better system of recognizing sources of opinions can provide more accurate opinions, so that the joint model is promising to infer more correct opinions.

Previous works tend to recognize the opinion sources in two steps. First, the systems classify whether the opinion is attributed to an noun phrase source, or attributed to the writer. Then the systems identify the noun phrase source based on the classification result in the first step. However, there may be a problem with the classification in the first step. Consider the examples in Table 7.1.

Sentence (Boldfaced Opinion Expressions)	Source	Previous	Ours
(Ex7.1) Tom **criticized** the student.	Tom	noun	participant
(Ex7.2) Mary didn't expect that Tom **criticized** the student.	Tom	noun	participant
(Ex7.3) Mary says that Jack is very **considerate**.	Mary	noun	non-participant
(Ex7.4) Jack is very **considerate**.	writer	writer	non-participant
(Ex7.5) He **embezzled** the pension.	writer	writer	non-participant

Table 7.1: Examples of Opinions, Sources of opinions, and Categories of opinions

In (Ex7.1), there is a negative opinion. The opinion expression is the word criticized. The source is Tom. The target is the student. In (Ex7.2), the source of the negative opinion criticized is still Tom, though it is according to Mary's unexpected thoughts. In (Ex7.3), there is a positive opinion. The opinion expression is the word considerate. The positive opinion is stated by Mary, and the source of it is Mary as well. In (Ex7.4), the positive opinion toward Jack is attributed to the writer (or the speaker). In (Ex7.5), there is a negative opinion. The opinion expression is the word embezzled. The source is the writer. The target is He.

As we can see, the source of an opinion can be the writer or an entity represented by a noun phrase in the sentence (e.g., Mary, Tom). Several previous works contribute to recognizing noun phrases as sources (Choi et al., 2005, 2006; Wiegand and Klakow, 2010). They use sequence labeling techniques to label phrases as sources or classify noun phrases in the sentence. A few recent works (Yang and Cardie, 2013a; Johansson and Moschitti, 2013a) develop binary classifiers to determine whether the source is the writer. If not, they use similar methods as previous works did to label noun phrases as sources. In short, previous works categorize opinions as the ones whose sources are the writers or the ones whose sources are noun phrases. According to the previous works, (Ex7.1), (Ex7.2) and (Ex7.3) are in the same category since the sources are noun phrases, while (Ex7.4) and (Ex7.5) are in the other category since the sources are the writer.

However, not all the noun phrase sources play the same role in terms of the opinions. The opinion expressions in (Ex7.1) and (Ex7.2) are events (i.e., actions) and the sources are the agents of the events. (Tom is the agent of the criticizing event which represents a negative opinion.) However, though the opinion expression in (Ex7.5) is also an event (embezzled), the source is not the agent (He) but the writer. The opinion expressions in (Ex7.3) and (Ex7.4) are not events and the sources are not semantic roles of the opinion. Further, if we interpret (Ex7.3) as *Mary:"Jack is very considerate."*, (Ex7.3) is very similar to (Ex7.4) since the sources are the persons who state the opinions. Similarly, the source in (Ex7.5) is the writer who state the opinion instead of anyone participating in the embezzling event. Thus, the methods developed to recognize sources in (Ex7.3), (Ex7.4) and (Ex7.5) should be different from the methods developed to recognize sources in (Ex7.1) and (Ex7.2). However,

previous works develop the same method to recognize sources in (Ex7.1), (Ex7.2) and (Ex7.3) since the sources are all noun phrases. They use semantic role labeling outputs as important features during the training. This may result in failing to find the source in (Ex7.3) because the source is not a semantic role of the opinion. This may also result in misclassifying him as the source in (Ex7.5) because he is the agent of the event. Different from previous works, we first classify (Ex7.1) and (Ex7.2) in one category, and classify (Ex7.3), (Ex7.4) and (Ex7.5) in the other category, and then recognize the sources. Specifically, we name opinions such as (Ex7.1) and (Ex7.2) as **participant opinions** because the sources are participants in the events that trigger the opinions. we name opinions such as (Ex7.3), (Ex7.4) and (Ex7.5) as **non-participant opinions** because the sources are non participants. For example, the source can be someone who states the opinions.

Therefore, here we define a new categorization of the opinion types (Section 7.1). We also build a new automatic system to recognize the sources based on the new categorization of the opinion types (Section 7.2). We also conduct an experiment to show that the new categorization can help recognize sources (Section 7.3).

7.1 DEFINITIONS OF TWO TYPES OF OPINIONS

A participant opinion is an opinion attributed to someone who is a participant in the event that triggers the opinion. The opinion expression of a participant opinion is usually an event directly triggering opinions (e.g., criticize in (Ex7.1) and (Ex7.2)). The source of it (e.g., Tom in (Ex7.1) and (Ex7.2), **participant source**) is usually a noun phrase.

A non-participant opinion is an opinion attributed to someone who is not a participant of the opinion. The opinion expression of a non-participant opinion is usually a description of the target (e.g., considerate in (Ex7.3) and (Ex7.4)). The source (**non-participant source**) can be the writer of the document such as (Ex7.4) and (Ex7.5) (**writer source**). There is no span for the writer source in the text. A non-participant source can be an entity in the text such as (Ex7.3) (**nonParticipantNP source**). A nonParticipantNP source is usually a noun phrase.

One thing to point out is that in some opinion-oriented corpora such as MPQA (Wiebe et al., 2005) a few opinions do not have explicit sources in the sentence. Consider "*Insulting the Prophet is a violation to human rights.*" The insulting event does not have an explicit agent. The negative opinion triggered by the insulting event does not have an explicit source as well. The source may refer to anyone in the world. Now consider "*Chavez' candidacy raised expectations.*" Though it is annotated as a positive opinion, the sentence does not specify whose expectations are raised, neither does it specify who are positive about this. The goal of this paper is to anchor the opinion sources to specific entities (either the writer or heads of noun phrases in the sentence). But such implicit sources cannot be anchored to a specific entity. Thus, we do not consider opinions whose sources are implicit. About 5 percent opinions have implicit sources in MPQA.

7.2 MODEL

The model recognizing sources consists of two steps. In the first step, the model classifies the type of the opinion as whether the opinion has a participant source or a non-participant source. Based on the result of the first step, the model then recognizes the sources. We will talk about the two steps in the following sections.

7.2.1 Classifying Two Types of Opinions.

We develop a binary classifier to distinguish non-participant opinions from participant opinions.

Features. We use embeddings of opinion expressions as features for the binary classifier. We did not use any linguistic feature such as Part-Of-Speech or N-gram. Compare (Ex7.1) and (Ex7.5) at the beginning of this section. Both opinion expressions are events and they have agents. Further, both opinion expressions are the words (criticized in (Ex7.1) and embezzled in (Ex7.5)) which can be found in sentiment or connotation lexicons. But (Ex7.1) is a participant opinion while (Ex7.5) is a non-participant opinion. Rather, we want to

capture the differences in the meanings of the opinions expressions. We follow the same method in (Socher et al., 2011) to generate word-level and phrase-level embeddings, which were used to recognize paraphrases. It is promising to use such embeddings to represent the meanings of the phrases.

In (Socher et al., 2011)[1], an unfolding recursive autoencoder ($uRAE$) is used to learn the embeddings for nodes on the binary parse tree where each parent has two children. During the encoding, the parent vector p_1 is computed from the children vectors (c_1,c_2), recursively the parent vector p_2 is computed from the children vectors (c_3,p_1).

$$p_1 = f(W_e[c_1; c_2] + b_e) \quad p_2 = f(W_e[c_3; p_1] + b_e) \tag{7.1}$$

where the W_e is the encoding matrix to learn. Recursively we encode all the non-terminal nodes on the binary parse tree. Then during the decoding process, the parent vector is decoded to the two children vectors via reconstruction:

$$[c_3'; p_1'] = f(W_d[p_2] + b_d) \quad [c_1'; c_2'] = f(W_d[p_1'] + b_d) \tag{7.2}$$

where W_d is the decoding matrix to learn. Similarly to encoding, the decoding is recursively conducted on each node.

For the node p_2 that spans from word c_1, c_2 to c_3, the Euclidean distance between the original input of the leaves and the reconstructed representations of leaves is:

$$E(p_2) = \left\| [c_1'; c_2'; c_3'] - [c_1; c_2; c_3] \right\|^2 \tag{7.3}$$

By minimizing the sum of all the Euclidean distances between all the nodes, we can learn the encoding and decoding matrices.

Following (Socher et al., 2011), we generate a 100-dimension embedding vector for each opinion expression. A binary classifier is trained to learn the weight of each dimension. In the next section we discuss how we obtain the training data and how we use the training data to learn the weights.

[1]Available at http://goo.gl/4vKQGu

Training. A big challenge of this classification is the lack of labeled data. Though fine-grained opinion annotated corpora such as MPQA provide the source annotations of opinions, the annotations do not contain labels specifying whether the source is a participant or not. Thus, we develop methods to utilize the existing resources annotated with the opinion sources to collect training data for our new categorization. Basically, we select non-participant opinion instances and participant opinion instances. The corpus we use is MPQA 2.0 (Wiebe et al., 2005; Wilson, 2008), which has been annotated with many opinions on various topics together with the opinion sources. Since there is no such label corresponding to the distinction between the two types of opinions, we use heuristics to select non-participant opinion instances and participant opinion instances. A selected non-participant opinion instance should have a much higher confidence of being a non-participant opinion than being a participant opinion. For non-participant opinion instances, we collect the opinion expressions whose sources are annotated as the writer since we are sure such opinions are non-participant opinions. Similarly, a selected participant opinion instance should have a much higher confidence being a participant opinion than being a non-participant opinion. Based on our observation (e.g., (Ex7.1) and (Ex7.2)), a participant opinion is usually a predicate and its source is usually the subject (A0) of the predicate. For participant opinion instances, we collect the opinion expressions which are predicates and at the same time their sources are the subjects of the predicates. For the remaining opinion annotations, we treat them as unknown instances.

We use two different training methods to learn the weights. In the first method, we simply use the selected non-participant opinion and participant opinion instances. We train an SVM classifier (Vapnik, 2013; Joachims, 1999a) to learn the weights (*non-transductive SVM*). In the second method, we use all the instances including the unknown instances to train a transductive SVM (Joachims, 1999b) to learn the weights (*transductive SVM*)[2].

The transductive SVM uses the unlabeled data to adjust the boundary so that the hyperlane separates both labeled and unlabeled data in the training set. Note that, only unlabeled data in the **training** set are used to learn the weights. None of the testing data is observed during training. It adds slack variables (ξ_i and $\hat{\xi_u}$), which allow the model to

[2]Available at http://svmlight.joachims.org/

trade-off between misclassifying labeled data (x_i, y_i) and excluding unlabeled data $(\hat{x_u}, \hat{y_u})$ (Joachims, 1999b).

$$\begin{aligned} \min_{w,\hat{y_u},\xi_i,\hat{\xi_u}} \quad & \tfrac{1}{2}||w||^2 + C\sum_i \xi_i + \hat{C}\sum_u \hat{\xi_u} \\ \text{s.t.} \quad \forall i \quad & y_i(wx_i + b) \quad \geq 1 - \xi_i \\ \forall u \quad & \hat{y_u}(w\hat{x_u} + b) \quad \geq 1 - \hat{\xi_u} \end{aligned} \tag{7.4}$$

7.2.2 Recognizing Sources of Two Types of Opinions

(R1) Opinion(o) $\wedge$ NPW(s) $\wedge$ NonParticipantOpinion(o) $\wedge$ NonParticipantNP(o,s) $\rightarrow$ Source(o,s)
(R2) Opinion(o) $\wedge$ NPW(s) $\wedge$ NonParticipantOpinion(o) $\wedge$ Writer(o,s) $\rightarrow$ Source(o,s)
(R3) Opinion(o) $\wedge$ NPW(s) $\wedge$ ParticipantOpinion(o) $\wedge$ CRF(o,s) $\rightarrow$ Source(o,s)
(R4) Opinion(o) $\wedge$ NPW(s) $\wedge$ ParticipantOpinion(o) $\wedge$ SemanticAgent(o,s) $\rightarrow$ Source(o,s)
(R5) Opinion(o) $\wedge$ NPW(s) $\wedge$ ParticipantOpinion(o) $\wedge$ SyntacticAgent(o,s) $\rightarrow$ Source(o,s)
(R6) NonParticipantOpinion(o) $\rightarrow \sim$ ParticipantOpinion(o)
(R7) ParticipantOpinion(o) $\rightarrow \sim$ NonParticipantOpinion(o)

Table 7.2: Rules in PSL

To recognize sources of the two types of opinions, we choose a joint model instead of a pipeline end-to-end system which may suffer from accumulated errors. Different from the joint models used in the previous works which extract both writer sources and noun phrase sources, we choose Probabilistic Soft Logic (PSL) (Broecheler et al., 2010)[3]. Previously we have used the PSL models to recognize entity/event-level sentiments in Section 6.2. To give an overview of the PSL model used in this section, we define the literal Source(o,s) to represent the grounding that the source of the opinion o is s, where o is an opinion expression and s can be the writer or a noun phrase in the sentence. If o and s are constants, then Source(o,s) is a grounded literal. Each grounded literal is assigned a score by an individual system, PSL takes as input all the individual scores and the constraints defined

[3]Available at http://psl.umiacs.umd.edu/

by rules among literals. In the final output, for example if the score of the grounded literal SOURCE(criticized, Jack) is larger than zero, it means that PSL thinks Jack is the source of the opinion criticized, and the score SOURCE(criticized, writer) being 0 represents that PSL thinks the writer is not the source of that opinion.

Next, we introduce the literals defined for recognizing sources. Then we introduce the rules used as constraints.

Literals. Two sets of variables are used in the PSL in this paper. The first set consists of opinion expressions, each of which is denoted o. The second set consists of sources, each of which is denoted s. Since this paper focuses on recognizing sources, each o is an opinion expression in the gold standard. Given an opinion expression o in the sentence, we automatically generate a set S_o consisting of different source candidates of o. Each $s_o \in S_o$ is either the writer or the head of an NP. Note that we filter out any s that may not be an entity's head. We require that an entity must meet at least one of the three criteria: (1) it is a named entity; (2) it is a pronoun; (3) it is an animate according to the lexicon (Ji and Lin, 2009). Each entry of the lexicon consists of an NP, the frequency that NP is used as an animate (labeled as *who*) and the frequency that NP is used as a non-animate (labeled as *which*, *when*, or *where*). We consider an NP as an animate if the frequency of *who* is higher than the frequency of any other label.

First of all, we define three basic literals.

(A1) OPINION(o): o is an opinion expression

(A2) NPW(s): entity s is a source (NP or writer)

(A3) SOURCE(o,s): the source of opinion o is s

For an opinion o in the gold standard, we create an OPINION(o) and the score is 1.0. For each source candidate s that individual systems generate, we create an NPW(s) and the score is 1.0. The scores of SOURCE(o,s) will be the outputs calculated by the joint model.

Next, we define two literals to describe o.

(A4) NONPARTICIPANTOPINION(o): o is a non-participant opinion

(A5) PARTICIPANTOPINION(o): o is a participant opinion

The classifier in Section 7.2.1 outputs a score of each o. If the output is larger than zero, we create a NONPARTICIPANTOPINION(o) and the score is the output. If the output is

smaller than zero, we create a ParticipantOpinion(o) and the score is the absolute value of the output. The absolute scores outside the range [0,1] is set as 1.0.

Further, we define literals representing how we automatically generate source candidates. All the literals defined below represent the grounding that s is a source candidate of o and they are assigned with score 1.0.

(A6) NonParticipantNP(o,s): s is an NP head as non-participant

(A7) Writer(o,s): s is the writer

(A8) CRF(o,s): s is an NP head extracted by a CRF model

(A9) SemanticAgent(o,s): s is the semantic agent of o

(A10) SyntacticAgent(o,s): s is the syntactic agent of o

NonParticipantNP(o,s) is created if o is a clause and s is the NP head that dominates o on the constituency parse tree. Specifically, if o is a clause (e.g., its parent node is labeled as *SBAR* in the parse tree), we go up the parse tree from o till the root, and collect the heads of the noun phrases along the path.

Writer(o, writer) is created if no NonParticipantNP literal of o is created.

CRF(o,s) is created if a pre-trained Conditional Random Filed (CRF) model extracts s as the source of o. Previous experiments have shown that CRF is a strong model in extracting noun phrases as sources (Yang and Cardie, 2013a; Johansson and Moschitti, 2013a). We expect a CRF model could recognize the participant sources. Note that if the output from CRF is an NP, we choose the head of it as s. The features used in the model are typical linguistic features used in the previous works (Yang and Cardie, 2013a).

SemanticAgent(o,s) is created if o is a predicate and s is the head of the subject (A0) of the predicate extracted by a semantic role labeling tool. We use SENNA (Collobert et al., 2011) as the semantic role labeling tool in this paper.

SyntacticAgent(o,s) is created if s is the nsubj of o according to the dependency parser. We add the syntactic agent to retain the recall if the CRF model or semantic role labeling tool misses any source of an opinion. We use Stanford's dependency parser in this paper (Manning et al., 2014).

For an opinion o, the set S_o consists of all the source candidates. PSL assigns scores to each s_o in S_o. A subset $S_o^{\text{non-participant}}$ consists of s_o created for NonParticipantNP and

WRITER literals. The other subset $S_o^{\text{participant}}$ consists of s_o created for CRF, SEMANTICAGENT and SYNTACTICAGENT literals.

Rules. We define rules used as constraints in PSL to model the relations of literals, shown in Table 7.2. In the top box, Rules (R1) and (R2) are defined to find the sources of non-participant opinions. For example, Rule (R2) can be explained as: if the opinion expression o is a non-participant opinion and s is a source candidate which is the writer, then we infer the source of o is s. In the middle box, Rules (R3), (R4) and (R5) are defined to find the sources of participant opinions. In the bottom box, Rules (R6) and (R7) are defined to ensure that the same opinion cannot be both a non-participant opinion and a participant opinion. As introduced in Section 6.2.1, each rule is associated with a weight, representing how important the rule is. The weights of Rules (R6) and (R7) are infinite because they are used as hard constraints. The other weights are learnt on the training set in the experiment.

7.3 EXPERIMENTS AND RESULTS

Our experiments are conducted on MPQA 2.0[4] (Wiebe et al., 2005), a widely used corpus for fine-grained opinion analysis. 135 documents are used as a development set and a different set of 400 documents are used for 10-fold cross-validation. Our gold standard opinion expressions are corresponding to the direct subjectivity annotations and expressive subjectivity annotations. Our gold standard sources are corresponding to the agent annotations. We filter out the opinions whose sources are annotated as *implicit* (as stated in Section 7.1) and filter out the opinions whose sources are outside the sentence. The set of opinion expressions are the gold standard annotations. There are 11,364 opinion expressions in the cross-validation set. 3,826 opinion sources (33.67%) are annotated as the writer, and the other 7,538 opinion sources (66.33%) are annotated as noun phrases.

In the cross-validation, we use the training set in each fold to train three components of the whole model: (1) the classifier to classify opinion expressions in Section 7.2.1; (2) the CRF to extract noun phrases as sources in Section 7.2.2; (3) the PSL to conduct joint

[4]Available at http://mpqa.cs.pitt.edu/

prediction in Section 7.2.2. After training, the whole model extracts sources in the testing set in each fold.

For evaluation, similar to previous works (Yang and Cardie, 2013a), we use precision (P), recall (R) and F-measure (F1) according to *overlap* and *exact* matching metrics. For both metrics, if the automatically extracted source is the writer and the gold standard annotation is also the writer, it is a correct hit. In other cases, according to exact metric, if the automatically extracted source is the semantic head of the gold standard annotation span, it is a correct hit. According to overlap metric, if the automatically extracted source is within the gold standard annotation span, it is a correct hit.

We have conducted three experiments. The first experiment discusses the performances of our model in recognizing sources, compared to baselines and state-of-the-art works. The second experiment discusses the contribution of transductive SVM in recognizing sources, compared to non-transductive SVM. The third experiment discusses the learnt weights in the trained PSL. Next we talk about the three experiments.

7.3.1 Performance of Recognizing Sources.

We use two baseline methods. For each opinion o from the gold standard, the first baseline (S_o) uses the whole source candidate set S_o as described in Section 7.2.2. The second baseline ($S_o^{n/p}$) uses a subset of S_o based on the classifier output. If the classifier labels o as a non-participant opinion, then the second baseline chooses the subset $S_o^{\text{non-participant}}$ as described in Section 7.2.2. If the classifier labels o as a participant opinion, then the second baseline chooses the subset $S_o^{\text{participant}}$. For an opinion o, the outputs from the two baselines S_o and $S_o^{\text{n/p}}$ are sets of sources, which may contain more than one source. The third baseline ($1 \in S_o^{\text{n/p}}$) builds upon the second baseline. It chooses the writer source from $S_o^{\text{non-participant}}$ if o is classified as a non-participant opinion. It chooses the source candidate extracted by the CRF model if o is classified as a participant opinion. The third baseline always outputs a single source for an opinion. It combines the classification result and current CRF model in a pipeline approach. Our full model (*Joint*) uses the output from PSL. For an o, we take the s that has the highest positive score of $\textsc{Source}(o,s)$. The performances are shown in Table

7.3. A star in F-measures indicates statistical significance according to t-test ($p < 0.05$).

	exact			overlap		
	P	R	F1	P	R	F1
S_o	36.21	**71.57**	48.09	36.50	**72.14**	48.47
$S_o^{\mathrm{n/p}}$	47.50	58.35	52.37	50.19	58.88	54.19
$1 \in S_o^{\mathrm{n/p}}$	49.16	49.16	49.16	49.67	49.67	49.67
Joint	**67.74**	60.20	**63.75***	**68.33**	60.73	**64.31***

Table 7.3: Performances of Recognizing Sources

The first baseline S_o has the highest recall because it considers all the source candidates. The second baseline $S_o^{\mathrm{n/p}}$ successfully removes some wrong candidates by improving the precision and F-measure over S_o, indicating that classifying opinions can help recognize sources. The third baseline $1 \in S_o^{\mathrm{n/p}}$ has a higher precision but a sharp drop in recall. This is because a pipeline approach may rule out correct candidates. The full model *Joint* achieves the best performance. Note that, the performances using overlap metric is only slightly better than using exact metric. This indicates that when our model recognizes a NP head as the source, in most cases it is the semantic head of the gold standard annotation. Note that the recall of S_o is not 100%. The errors are cases where the heads of sources are not nouns or pronouns. For example, the head of the source span "*Those signing the document*" is the word those, whose Part-Of-Speech label is DT.

In addition to the two baselines, we also choose three models from state-of-the-art works for comparison. The state-of-the-art works were conducted on MPQA with 10-fold cross validations as well. Thus we compare to their reported numbers directly. The first model (*Pipeline*), which is a pipeline approach, (Yang and Cardie, 2013a) uses CRF to extract opinion expressions, opinion sources and opinion targets. Then binary classifiers are used to link the extracted sources (including the writer) and targets to opinions. Based on the first model's result, the second model (*ILP*) (Yang and Cardie, 2013a) uses ILP to optimize the results. The third model (*Re-Rank*) is very similar to the second model, except that the third model only extracts opinion expressions and opinion sources and it uses a re-ranker

to optimize the result. Since the state-of-the-art models automatically extracted opinion expressions, for a better comparison we train a CRF as described in Yang and Cardie (2014) to extract opinion expressions as well. The training of the CRF is also conducted on the training set in each fold. Our model (*Auto+Joint*) in this experiment is different from the *Joint* in Table 7.3 since *Auto+Joint* takes as input automatically extracted opinion expressions. Using the evaluation methods in Yang and Cardie (2013a), we evaluate on the opinions that are correctly extracted by the model, which are a subset of all the opinions in the corpus. Re-Rank calculates the percentage of overlapping tokens if an automatically extract source span overlaps with the gold standard span. The performances are shown in Table 7.4.

Method	P	R	F1	metric
Auto+Joint	**66.95**	**60.29**	**63.45**	overlap
Pipeline	47.73	54.40	50.84	overlap
ILP	64.97	58.61	61.63	overlap
Re-Rank	53.20	55.10	54.20	percentage

Table 7.4: Comparisons to State-of-the-art Models

Our model (*Auto+Joint*) has the highest F-measure. The Pipeline has the lowest F-measure, indicating that a joint approach is more appropriate for recognizing sources. The ILP is better than the Re-Rank, and ILP is slightly worse than our model. It optimizes both the opinion-source relation and the opinion-target relation. It is promising to use PSL to jointly optimize the extraction of sources and targets in the future.

7.3.2 Contribution of Transductive SVM

In Section 7.2.1, we introduce two methods to train the classifier, i.e., non-transduction and transduction. The F-measure of Joint using the transductive SVM in Table 7.3 is 63.75%. When we use the classification results from a non-transductive SVM for the Joint method, the F-measure is 61.47%, which is worse and the difference is statistically significant via

a t-test ($p < 0.05$). The F-measure of the baseline ($S_o^{n/p}$) using transductive SVM is also statistically better than a $S_o^{n/p}$ using non-transductive SVM (52.37% versus 50.59%). Our experiments have shown that using transduction to train the classifier is able to improve recognizing sources.

7.3.3 Discussion of Trained PSL

We train the PSL model to learn the weights of rules. The weights learnt in each fold follow the same trend. Since the initial scores for Literals (A6)-(A10) are set to be 1.0, the learnt rule weights are good estimates for how important each source candidate is. (R2) has the highest weight, indicating that if there is a writer source candidate extracted, it is very likely the correct source of an opinion. (R4) has a slightly higher weight than (R3). Though in most cases the CRF candidate is the same as the semantic agent candidate, the model prefers the semantic agent candidate if the two are not the same. (R5) has the lowest weight, which is not surprising because syntactic agent is a weak candidate. Furthermore, we run a PSL model without learning the weights but assigning each rule with the same weight. The performances are slightly worse. This shows that the trained weights of rules help recognize correct sources.

7.4 RELATED WORK

Different from the works analyzing reviews that assume the sources are the writers (Liu, 2012; Socher et al., 2013), Choi et al. (2005) use Conditional Random Field (CRF) to recognize which phrases are the sources of opinions. Later, Choi et al. (2006) use CRF to automatically extract both opinion expressions and opinion sources. A binary classifier is run to assign sources to opinions. Finally an Integer Linear Programming model is run to choose the best configuration of correspondences of opinions and sources. Wiegand and Klakow (2010) consider all the noun phrases in the sentence and train a binary SVM classifier to judge whether a noun phrase is the source of a given opinion expression. They develop new

convolution kernels used in SVM which are able to identify meaningful fragments of sequences or trees. Later Wiegand and Klakow (2012) develop generalization features to improve cross-domain opinion source extractions. Different from all the aforementioned works, this paper focuses on both cases where the sources can be the writers or phrases.

A few previous works extract sources including both writers and phrases in the text (Yang and Cardie, 2013a; Johansson and Moschitti, 2013a). They follow a procedure similar to that of (Choi et al., 2006). One of the differences from (Choi et al., 2006) is that a binary classifier is run to predict when the source is not a noun phrase. By this classifier, the model tries to recognize writer sources. Though the state-of-the-art works take into account both writer and noun phrases as potential sources, they did not model the distinction between participant opinions from non-participant opinions.

The sources of some non-participant opinions are the people in the text who state the opinion such as (Ex7.3). Recognizing such sources is similar to speaker attribution in quotation analysis (Glass and Bangay, 2007; Elson and McKeown, 2010; O'Keefe et al., 2012; Pareti et al., 2013). We did not employ the techniques for speaker attribution in this paper because the features used in speaker attribution are extra-sentences and even extra-paragraphs, while we focus on recognizing sources of opinions within the sentence in this paper.

7.5 SUMMARY

This chapter improves recognizing sources of opinions based on a new categorization of opinions: non-participant opinion or participant opinion. A transductive SVM is built to classify an opinion utilizing existing limited resources. The categorization information is then utilized by a Probabilistic Soft Logic model to jointly recognize sources of the two types of opinions in a single model. The experiments have shown that the model based on this new categorization of opinions achieves better performances over baselines and several state-of-the-art works in recognizing sources.

8.0 SENTIMENT INFERENCE IN CHINESE

In the previous sections, we have developed corpora, inference rules, and computational models for the entity/event-level sentiment analysis task. We got the idea of inference from English text. The resources on which the experiments that are carried out and evaluated are developed in English text. However, people speaking different languages may think differently. It would be interesting to explore whether the same rules work for languages other than English. In this chapter, we focus on validating the inference rules in the Chinese contexts. We will also discuss whether there is any existing literature in Chinese Natural Language Processing to build an automatic system as we developed for English text in the previous chapters. We choose to focus on Chinese because it is a widely-spoken language and there are many existing NLP tools such as part-of-speech tagger and syntax analysis tool to use. Furthermore, not limited to repeating the experiments on Chinese text, this chapter also discusses the differences between Chinese and English in the inference rules and the potential cases that a Chinese automatic system needs to handle.

Similarly to previous chapters, we develop a small corpus in Chinese annotated with explicit and implicit sentiments in Section 8.1. We run the graph-based model previously developed in Section 5.1.2 to validate the rules, and discuss some Chinese context that may block the inference rules in Chinese text in Section 8.2. We also talk about what resources and systems are promising to utilize to build an automatic sentiment inference system for Chinese text in Section 8.3.

8.1 CHINESE IMPLICIT OPINIONS CORPUS

People may express their opinions in different ways in different languages. Since there is not any Chinese corpus annotated with the sentiments toward entities and events, we try to annotate a small corpus serving as resources of the analysis. Here we annotate the sentiments toward the agents and themes of the +/-effect events. We did not annotate the sentiments toward any entity or event like MPQA 3.0 because there is no such corpus as MPQA 2.0 which provides fine-grained sentiments annotations. On the contrary, it is feasible to annotate the sentiments toward the agents and themes of +/-effect events, as shown previously in Section **??**. Furthermore, the sentiment inferences triggered by +/-effect events are the main parts of the inferences in this thesis. Thus, we annotate such sentiments in the Chinese text and conduct an agreement study, which achieves good agreement scores, reported in Section 8.1.1. The good agreement scores providing evidence that it is feasible to provide reliable annotations of the sentiments toward the agents and themes of the +/-effect events. In the disagreement analysis, we have observed interesting cases which are +/-effect events in semantics but are triggered by Chinese syntax. We discuss the cases in Section 8.1.2.

8.1.1 Agreement Study

8.1.1.1 Data. We collect 100 political editorials from the Opinion Column in the Chinese version of New York Times[1], where each political editorial has an English version and a Chinese version. The Chinese editorial is a translated and paraphrased version of the corresponding English editorial, written by professional translators. The English version and the Chinese version are paragraph aligned. In the previous agreement study Deng et al. (2013), the annotators are asked to annotate the whole document. Because not all the sentences contain +/-effect events and the documents are long, a large proportion of disagreement we find that is due to negligence. In order to reduce negligence and provide a more dense data for annotation, first, we collect a lexicon of English +/-effect event words or phrases in the English +/-Effect Event Sentiment Corpus in Section 4.1. Then we find the English sentences

[1] http://cn.nytimes.com/opinion/

containing English +/-effect event words or phrases and select the paragraphs containing those sentences. The parallel Chinese paragraphs are collected. Though a paragraph may contain more than one sentence and some sentences do not have +/-effect events, it is much more dense to annotate such paragraphs than to annotate the document as a whole. When presenting data to the annotators, we do not provide an isolated paragraph since it may lose the context information. Instead, we present the original Chinese editorials and highlight the selected paragraphs. The annotators are told to read through the whole document but only need to annotate the highlighted paragraphs.

We adopt our English manual in Deng et al. (2013) to train the annotators. The annotators read through the manual and several Chinese +/-effect examples. Then, the annotators label several paragraphs and discuss their disagreements to reconcile their differences. For the formal agreement study, we randomly select 60 paragraphs, which have a total of 253 Chinese sentences. These paragraphs are different from the paragraphs discussed during training. The annotators then independently annotated the 60 selected paragraphs.

8.1.1.2 Evaluation and Result.

Similar to the agreement study evaluation in Section 4.1, we use the same measurement for agreement for all types of spans. (The type is either +/-effect event, agent, theme, or influencer). Suppose A is a set of annotations of a particular type and B is the set of annotations of the same type from the other annotator. For any text span $a \in A$ and $b \in B$, the span coverage c counts the percentage of overlapping Chinese characters between a and b,

$$c(a, b) = \frac{|a \cap b|}{|b|} \tag{8.1}$$

where $|a|$ is the number of characters in span a, and $\cap$ gives the set of characters that two spans have in common (Johansson and Moschitti, 2013b). As Breck et al. (2007) points out, this measure penalizes a span covering the whole sentence.

Following Wilson and Wiebe (2003), we treat each set A and B in turn as the gold-

$agr(A, B)$	+/-effect event	agent	theme
Anno 1& 2	0.7929	0.9091	0.9091
Anno 1 & 3	0.7044	0.9524	1.0
κ	+/-effect event effect	agent sentiment	theme sentiment
Anno 1 & 2	0.9385	0.7830	0.7238
Anno 1 & 3	0.8966	0.5913	0.8478

Table 8.1: Results for Agreement Study Analysis.

standard and calculate the average F-measure ($agr(A, B)$).

$$agr(A||B) = \frac{\sum_{\substack{a \in A, b \in B, \\ |a \cap b| > 0}} c(a, b)}{|B|} \tag{8.2}$$

$$agr(A, B) = \frac{agr(A||B) + agr(B||A)}{2} \tag{8.3}$$

Now that we have the sets of annotations on which the annotators agree, we use κ (Artstein and Poesio, 2008) to measure agreement for the attributes. We report three κ values: one for the effects of the +/-effect events and the influencers, and the other two for the writer's sentiments toward the agents and themes.

Three annotators participated in the agreement study. All of them are Chinese graduate students studying in US. One of the annotators is me (*Anno1*). The other two (*Anno2, Anno3*) do not know details of +/-effect events or sentiment inferences before . Since *Anno1* is familiar with this work, we compare the other two's annotations to *Anno1*'s. In Table 8.1, the upper half is the agreement for span overlapping ($agr(A, B)$), and the lower half is the agreement for attribute (κ).

The results have shown that the annotators have good agreement scores, though our training period is not long and our training data cover multiple topics. In particular, the annotators agree quite well on recognizing the agents and themes and judging the effect of

+/-effect events and influencers.

For recognizing +/-effect events, we have found two interesting cases caused by the Chinese syntax that is different from English, elaborated in the next section. Among the spans only one annotator marked, one third are due to the two cases above; one third are borderline cases that could be marked either way; one third are incorrect. For the spans two annotator mark but the third doesn't, we regard it as negligence.

For judging the writer's sentiments toward the agents and themes, we can see from Table 8.1 that *Anno 2* and *Anno 3* behave differently. This is understandable because we are marking the implicit opinions of the writer. Though trained, different annotators have different thresholds for judging whether an opinion is expressed here. Some annotators may be more sensitive than the others. If we don't count the spans that one annotator marks it as *none* (i.e. neutral) but the other doesn't, the κ scores increase a lot, as the Row *Polar* shows in Table 8.2. This indicates that the annotators mainly disagree on whether the sentiment is neutral or not, rather than the polarity of opinions.

To further investigate whether the disagreement is caused by Chinese, or is due to the annotators' inherent different sensitivities of opinions, we randomly select 5 documents from the English +/-Effect Event Sentiment Corpus developed in Section 4.1, delete the writer's sentiments toward the agents and themes but keep the remaining annotations. The annotators are then told to mark the sentiments. As Row *Eng* in Table 8.2 shows, we have got consistent agreement results within the same annotators when they annotate in English and in Chinese. This supports the idea that the differences between the annotators are differences in the underlying task, regardless of the language.

8.1.2 +/-Effect Event Triggered by Chinese Syntax

During the analysis of disagreement, we have found some +/-effect events which are triggered by Chinese syntax that is different from English. Since the annotators are trained with the English manual, some annotators stay consistent with the English syntax, but the others go beyond syntax and identify new +/-effect events according to semantics and pragmatics, which leads to disagreement. In this section we list two major cases caused by the Chinese

	Anno 1 & 2		Anno 1& 3	
	agent	theme	agent	theme
Table 8.1	0.783	0.723	0.591	0.848
Polar	0.875	0.915	1	0.88
Eng	0.738	0.652	0.4633	0.8734

Table 8.2: κ for Agreement Study Analysis.

syntax that is different from English syntax. This suggests that additional guidance to annotate such cases should be added to the English manual to develop a Chinese +/-effect event annotation manual.

The first case is due to the unclear expression of passive voice in Chinese. In English, the noun phrase that would be the theme of an active sentence (Our troops **defeated** the enemy) appears as the subject of a sentence with passive voice (The enemy **was defeated** by our troops)[2]. It is clear that enemy is the theme and our troops are the agent in both sentences. However, this is not intuitive for some Chinese sentences.

A Chinese example is "经济潜力似乎**得以释放**", whose English meaning is: "The economic potential ... appeared to **be unleashed**". A word-to-word translation would be "...appeared to **have got unleashed**". In the two English versions, potential is obviously the theme of the unleashed event. However, some annotators analyze this sentence according to syntax[3]. The dependency syntax between the theme potential (潜力) and the +/-effect event unleash (释放) is **nsubj**(释放-5, 潜力-2) so it is not marked. Some annotators view from pragmatics and read the sentences as they have passive voices. Since there is no word transformation of Chinese verbs for passive voice (e.g. the word unleash changes to the word unleashed in English), this raises disagreement.

The other case is related to one constraint defined in Deng et al. (2013). According to the manual, the effect of a +/-effect event must be determined within the triple. According

[2]http://en.wikipedia.org/wiki/English_passive_voice.
[3]We use Stanford's dependency parser in this section.

to the definition in Section 3.2.1, in (Ex3.2) "*His uncle left him a massive amount of debt*", the triple ⟨Tom, left, his cousin⟩ is not a +/-effect event, since we cannot judge whether this event is beneficial or harmful to him without knowing what his uncle leaves to him. However, in the sentence "*They decrease the manufacturing costs*", the decreasing event is a -effect event no matter how many or by what means the costs are decreased. However, a Chinese instance is, "**把改革置于死地** ", whose translation is "**put** the reform **to die**". Whether the putting event (把) is good for or bad for the reform (改革), depends on whether the agent puts the reform to die or puts the reform to revive, for instance. However, in Chinese, 把 is not main verb (Li and Thompson, 1989), the theme (改革, reform) of the main verb (置于死地, die) is placed after the function word (把), and the verb is placed after the theme, forming a subject, object, verb (SOV) sentence (Chao, 1968), which is defined as a *ba structure* (Chao, 1968; Li and Thompson, 1989; Sybesma, 1992). Thus, in Chinese the sentence is read as: "kill the reform", which could be seen as a +/-effect event. This structure is very common in Chinese. Whether to annotate such structure as a +/-effect event is not defined in the English manual, so it raises disagreement.

In summary, according to the annotations, there are very similar sentiment inferences in Chinese. However, in order to fully study the +/-effect events in Chinese, the manual should be revised to provide guidance for annotating the cases mentioned above.

8.2 INFERENCE RULES FOR CHINESE

Previously in Section 5.1.2, we incorporate the inference rules w.r.t +/-effect events developed in English into a graph-based model to conduct sentiment propagation among entities (agents and themes) of +/-effect events (Deng and Wiebe, 2014b). In this section, we run the same graph-based model on the Chinese annotations developed in Section 5.1.2. The positive results of sentiment propagation support our hypothesis that the inference rules apply for Chinese as well. Further, we categorize interesting +/-effect cases where the inferences are blocked in Section 8.2.1. From our observation, the blocking inferences are similar to what we have found in English (Wiebe and Deng, 2014).

In the graph-based model, a node represents an entity (agent, or theme), and an edge exists between two nodes if the two entities participate in one or more +/-effect events with each other. Scores on the nodes represent the sentiments, if any, expressed by the writer toward the entities. Scores on the edges are based on constraints derived from the rules. Loopy Belief Propagation (Pearl, 1982; Yedidia et al., 2005) is applied to accomplish sentiment propagation in the graph. Given a graph built from manual annotations, an evaluation is carried out to assess the ability to propagate sentiment of the model. In the experiment, for each subgraph (connected component), we assign one of the nodes in the subgraph with its gold standard sentiment polarity. Then we run LBP on each node in the subgraph. The experiment is run on the subgraph $|S|$ times, where $|S|$ is the number of nodes in the subgraph. Therefore, each node is assigned its gold-standard polarity exactly once, and each node is given a propagated value $|S|-1$ times, as propagated by each of the other nodes in its subgraph. We use Equations (8.4) and (8.5) to evaluate the chance of a node given a correct propagated label.

$$correct(a|b) = \begin{cases} 1 & a \ is \ correct \\ 0 & otherwise \end{cases} \tag{8.4}$$

$$correctness(a) = \frac{\sum_{b \in S_a, b \neq a} correct(a|b)}{|S_a| - 1} \tag{8.5}$$

Here we run the graph-based model on the Chinese annotations. The data we use include the training and testing paragraphs in the agreement study, in total 85 paragraphs, 341 sentences and 160 +/-effect triples. Later we use this corpus consisting of 160 +/-effect triples for analysis (denoted **Chinese +/-Effect Event Sentiment Corpus**). Since the edge scores of the model are defined according to the inference rules, if the sentiments are propagated correctly, this is a good evidence that the inference rules apply to Chinese.

The performances of the sentiment propagation are good, reported in Table 8.3. The model has a 70%-83% chance of propagating sentiments correctly in Chinese. This shows that the inference rules apply in Chinese and further we can utilize these rules to assist Chinese sentiment analysis. Compared to the scores of *correctness* reported in Deng and Wiebe (2014b), which are 0.8874 for all subgraphs and 0.9030 for multi-node subgraphs, the

Dataset	# subgraph	correctness
all subgraphs	136	0.7058
multi-node subgraphs	61	0.8251

Table 8.3: Performance of Graph-Based Model in Chinese.

scores of propagation in Chinese are lower. We analyze the reasons for the gap between our scores in Chinese and in English in the next section.

8.2.1 Blocking the Inference

A wrong propagation indicates the inferences related to that propagation are blocked. During the error analysis, we have found three interesting categories of cases where the inferences are blocked. Interestingly, we have observed these cases in English as well (Wiebe and Deng, 2014). In other words, we didn't find any blocking case specific to Chinese. The lower scores of *correctness* in Chinese might be due to the smaller amount of experimental data and more blocking cases in this corpus. Below we present examples to illustrate each case then the inference rules should be blocked. in the examples, the agent and theme are underlined and the +/-effect event is boldfaced.

Irrealis. This category contains +/-effect events that haven't or will not happen. One of the case is when the agent tried to conduct the +/-effect event, but failed. In (Ex7.1), by the rules, the writer has the same sentiment toward the agents and themes in +effect events and opposite sentiments toward the agents and themes in -effect events (Deng and Wiebe, 2014b). In (Ex7.1) below, the writer is negative toward both the agent and the theme, though this is a -effect event. This is because the event does not exist due to the failure, which is implied by the phrase, intended to. The inferences for +/-effect events in this category are blocked because the writer expresses the sentiments toward entities based on what they have done so far.

(Ex7.1) ...monetary policy activism intended to **counter** the cyclical bumps and grinds of the free market.

Forced +/-effect. This category contains +/-effect events whose agents don't intend to do that or being forced to conduct the event. For example, in (Ex7.2) below, though the triple ⟨Obama, delay, mandate⟩ is an event which does not happen, it is different from (Ex7.1). Here, Obama is forced to conduct the delaying event, though he does not want to and the writer does not blame him if he does so. For the entities involved in forced events, (at least the writer believes the entities are involuntary,) the forced event will not affect the writer's sentiments toward the entities so that the inferences are blocked.

(Ex7.2) Some of them even seem to think that they can bully Mr. Obama into **delaying** the individual mandate too.

Quoted +/-effect. This category contains +/-effect events in the quotations. Consider (Ex7.3) below, where one of the +/-effect triples is ⟨law, reduce, amount of labor⟩. In the original editorial, the writer supports the law and the writer has a positive sentiment toward the number of jobs (because he/she expects to see more job opportunities). But merely from the annotated +/-effect triple, it is inferred that the law has negative effect since it reduces the number of jobs. This is not contradictory with the writer's stance because the writer regards the event as a deliberate misreading he/she doesn't believe. The actual agent of the event should be (misreading, Obama). This example shows that inferences of a triple in the quotation are blocked, or event flipped, based on the writer's sentiment toward the agent saying the quotation. The agent in a quoted +/-effect is similar to the notion of *nested source* in sentiment analysis (Wilson and Wiebe, 2003).

(Ex7.3) Some of the job-killer scare stories are based on *a deliberate misreading* that estimated the law would "**reduce** the amount of labor used in the economy" by about 800,000 jobs.

In summary, the good performance in our pilot study gives supporting evidence for our hypothesis. That is, the inference rules apply for Chinese. Moreover, there is no evidence showing that the cases where the inferences are blocked only happen in Chinese.

8.3 COMPUTATIONAL MODEL IN CHINESE

Since the focus of this thesis is investigating sentiment inference and building modeling to automatically analyze sentiments in English text, we do not dive into developing specific systems analyzing sentiments in Chines. However, we still discuss the feasibility of building such automatic systems in Chinese. In this section, we will talk about the possible resources to utilize for sentiment analysis and inference in Chinese. As discussed in Section 8.1.2, there are some differences of +/-effect events in Chinese and English. Thus, in this section we will also talk about what changes need to be made in order to adapt to sentiment inference w.r.t. +/-effect event information in Chinese. It is promising to use the methods and resources presented in this section to build an automatic sentiment analysis system for Chinese text.

Specifically, we talk about how to build a Chinese +/-effect event lexicon in Section 8.3.1, how to recognize reversers in Chinese in Section 8.3.2, how to recognize agent and theme in Chinese in Section 8.3.3 and what are the state-of-the-art Chinese sentiment analysis systems in Section 8.3.4.

8.3.1 Chinese +/-Effect Event Lexicon

In this section, we compare the +/-effect spans in the Chinese +/-effect corpus and the corresponding English sentences, to investigate the possibility of deriving a bilingual +/-effect lexicon. Though the Chinese and English editorials are paragraph aligned, they are not sentence aligned, because an English sentence may be translated into multiple Chinese sentences and several English sentences may be merged into one Chinese sentence. Therefore, instead of automatic word-alignment, we manually pick up the parallel English spans of the Chinese annotated +/-effects. The correspondences of Chinese and English spans are categorized in Table 8.4. We present pairs of examples from the Chinese +/-Effect Event Sentiment Corpus, beginning with the original English sentence (*Eng*), followed by another English sentence which is the word-by-word translation of the Chinese sentence (*Chi*).

Parallel Span. This category contains instances where the Chinese annotated +/-effect spans have the parallel translations in the English sentences, and the English spans are also

Description	Count (Percentage %)	
Parallel Span	122	(76.25%)
Chinese Adding +/-effect	10	(6.875%)
Chinese Adding theme	6	(3.75%)
English Out Of Triple	5	(3.125%)
English Neutral	6	(3.125%)
Paraphrase	11	(6.875%)

Table 8.4: Counts of Chinese-English Corresponds

+/-effect words.

Chinese Adding +/-Effect Event. In the original English sentence below, its own making is a noun phrase rather than a +/-effect verb used as a noun. However, in the Chinese version, there is a clear triple, ⟨itself, makes, a monetary prison⟩. In such case the Chinese version adds a +/-effect event into the sentence.

[**Eng:**] ...the Fed is domiciled in a monetary prison of **its own making**.
[**Chi:**] ...the Fed is domiciled in a monetary prison **which itself makes**.

Chinese Adding Theme. As stated in the manual, all +/-effect triples should have themes. Thus, in the original sentence below, we will not mark the word exclusion because the theme is implicit. However, the Chinese version clearly states the theme, patients.

[**Eng:**] ...no more exclusion based on pre-existing conditions...
[**Chi:**] ...no more exclusion **of the patients** based on pre-existing conditions...

English Out Of Triple. Recall from Section 8.1.2, the +/-effect must be sufficient to perceive the +/-effect within the triple. In the example below, the ⟨the Fed, get, unemployment⟩ below cannot be considered as a +/-effect, since whether it is good for or bad for the unemployment depending on whether it is below 6.5% or up 6.5%, for instance. On the contrary, the Chinese version uses the word decrease, which is a -effect word, no matter how many percent are changed.

[**Eng:**] If and when the Fed, which now promises to **get** unemployment **below 6.5%**...
[**Chi:**] If and when the Fed, which now promises to **decrease** the unemployment to 6.5%...

English Neutral. Sometimes the English word doesn't have a +/-effect meaning but the Chinese word has one, based on the translator's interpretation of the whole editorial, though the triple structures are the same in English and Chinese versions.

[**Eng:**] We've **had eight decades of** increasingly frenetic monetary policy activism...
[**Chi:**] We've been **insisting** increasingly frenetic monetary policy activism for eight decades...

In the original English sentence, the phrase, had eight decades of, is hardly regarded as a +/-effect word. However, in the translated version, the word insisting is a +effect word. The change of wording introduces a new +/-effect event into the sentence.

Paraphrase. There are other cases where the sentences are paraphrased so largely that we cannot find a corresponding parallel span of the annotated Chinese span in the original English sentence. A majority of cases in this category are +/-effect events triggered by the Chinese syntax in Section 8.1.2.

In conclusion, the percentage of 76.25% in Row *Parallel Span* indicates that it is applicable to derive a bilingual +/-effect lexicon from a parallel corpus. However, we need to take into consideration the 23.75% mismatches for higher precision.

8.3.2 Chinese Reversers

The effect of a +/-effect event could be changed by a reverser (Deng et al., 2013). A common class of reversers is the negation. For example, in the sentence, "*the bill will not increase the costs*", the increasing event changes from +effect to -effect via the negation, not. In this section, we analyze Chinese reversers.

All of the reversers in the Chinese +/-effect corpus happen to be negations. In the English sentences, the negations are easily extracted by *neg* dependency relation. About 50% of the Chinese negations are linked to the +/-effect events via *neg* as well. Among this half, there are two negations commonly seen. One is 不 (Not), often labeled as AD (adverb)

in terms of Part-Of-Speech, the other is 没有 (do not have), labeled as VV (verb), shown below. The negation is underlined and the +/-effect event it negates is boldfaced.

(Ex7.4) 不/AD **接受/VV** 同性恋/NN

(Ex7.5) 没有/VV **刺激/VV** 贷款/NN

For the other half, the error mostly arises from segmentations. For the sentence below, though 没有 (doesn't have), often labeled as VB, could be regarded as a complete token, if we segment the two characters into two independent tokens, the parse is more similar to the English one. Below we only list the most relevant part of the parses.

[Eng:] He does n't have ability control war budget
[Eng dep:] neg(have-4, n't-3), root(ROOT-0, have-4), dobj(have-4, ability-6)
[Chi:] 他 没 有 能力 控制 战争 预算
[wrong dep:] root(ROOT-0, 没有-2), nsubj(控制-4, 能力-3), dep(没有-2, 控制-4)
[correct dep:] neg(有-3, 没-2), root(ROOT-0, 有-3), nsubj(控制-5, 能力-4)

In summary, it is feasible to recognize reversers in Chinese but it calls for a suitable word segmentation as input.

8.3.3 Syntax of Agent/theme in Chinese

According to Deng et al. (2013), the agent is the entity conducting the +/-effect event and the theme is the entity that the +/-effect event affects. This definition is very similar to the subject and (in)direct theme in semantic role labeling. Xue and Palmer (2004) investigate Chinese semantic role labeling. They utilize the PropBank and the constituency parser. However, from a preliminary analysis of constituency parse, we cannot distinguish the agent and theme merely from the parse tree, because the sentences in the editorials are usually complicated and it is difficult to classify whether a noun phrase (NP) constituency is the agent or theme in terms of its position. Kozhevnikov and Titov (2013) adopt a model transfer between different languages using dependency parser. In our case, the dependency parser has labels such as "nsubj" and "dobj", which are strong indications of agents and themes. Thus, we use the Stanford dependency parser, which has both English and Chinese parsers, to analyze the syntax of agents/themes in the +/-effect events. We count the types

of dependencies on the path in a dependency parse between the tokens of agents/themes and the tokens of +/-effect events in the DCW corpus and the Chinese +/-effect corpus.

Among all the dependency types, 19.57% of the labels between agents and +/-effect events are the ones specially designed for Chinese and 25.82% between themes and +/-effect events are the ones specially designed for Chinese. This indicates there is a considerable number of differences in dependency types. Chang et al. (2009), who create the Chinese parser, discuss the differences between Chinese and English types, which are similar to our observations.

First, there are more *nsubj* in Chinese for agents (21.53%) and more *dobj* in Chinese for themes (21.59%), compared to English (17.43% and 14.01%), which are easier for the parser to detect.

Second, the most common types specially designed for Chinese are *assm*, *assmod* and *cpm* (in total 12.23% for agents and 16.14% for themes). The relations *assm* is associative marker, *assmod* is associative modifier, and *cpm* is complementizer. These are defined because of the frequent usage of 的 (whose, of) in Chinese. Though there is not a direct mapping between Chinese and English dependency types, they are similar to two common types in English: *prep* and *pobj* (together 23.36% for agents and 31.62% for themes).

Third, there are more *rcmod* in Chinese than those in English. There are 7.05% and 6.5% *rcmod* in Chinese agents and themes, respectively. But there are only 1.7% and 2.16% in English agents and themes. The type *rcomd* is a relative clause modifier. If a verb is used as the modifier of a noun, it will be labelled *rcmod*. Instead, English writers tend to use more adjectives to modify nouns, which will be labeled *amod* (4.04% and 4.48%).

Fourth, there are 7.63% and 6.22% *punct* in Chinese agents and theme, compared to both 0% in English. In addition, there are 3.36% and 3.31% *conj* in English agents and themes.Chang et al. (2009) explain that English use conjunctions (*conj*) to link clauses while Chinese tend to use punctuation. Another finding in our corpus is that, translators tend to break down a long English sentence into several Chinese clauses, linked by punctuations.

For the other Chinese types, most of them are modifiers, which may be grouped with similar English modifiers.

Thus the statistics suggest that a separate agent or theme detector should be trained on

the Chinese data in order to correctly recognize the agents and themes.

8.3.4 Chinese Explicit Sentiment Analysis

There are various available resources for Chinese sentiment analysis, such as sentiment lexicon from HowNet[4], NTU Sentiment Dictionary (NTUSD) (Ku and Chen, 2007)[5] and the sentiment lexicon from Tsinghua University (Li and Sun, 2007). The sentiments recognized from lexicon hits are explicit, meaning that the writers use sentiment words to express his/her opinions. These explicit sentiment results are provided to the graph-based model as input. Note that the model plays a role of sentiment inference, instead of directly detecting sentiments from the text. The inferred sentiments are implicit, meaning that the writers express his/her opinions even without using a sentiment lexical clue.

8.4 RELATED WORK

Many works in Chinese sentiment analysis develop heuristics for adapting methods in English to methods appropriate for Chinese (Tsou et al., 2005; Wang et al., 2007; Li and Sun, 2007). Instead of projecting English methods and resources into Chinese versions, there are also works utilizing Chinese-English parallel corpus to assist Chinese sentiment analysis. Wan (2008) translates Chinese sentiment sentences into English and ensemble the sentiment classification results from both English and Chinese sentiment classifiers. Wan (2009) adopt co-training methods, utilizing labeled English sentences and unlabelled Chinese sentences. Lu et al. (2011) assumes parallel sentences in different languages bear the same sentiment. They utilize unlabelled Chinese-English parallel corpus to jointly improve sentiment classification in both languages. Boyd-Graber and Resnik (2010) present a generative model, jointly modeling topics that are consistent across languages, to improve sentiment rating predictions.

[4]Available at: http://www.keenage.com/html/e_index.html
[5]Available at: http://nlg18.csie.ntu.edu.tw:8080/lwku/pub1.html

8.5 SUMMARY

In this work we investigate implicit opinions expressed via +/-effect events in Chinese. The positive results have provided evidence that such implicit opinions and inference rules are similar in Chinese and English. There are some +/-effect events caused by the Chinese syntax, guidance for which could be added to the current English manual to develop a Chinese manual. Moreover, there is no evidence showing that the blocked inferences only happen in Chinese. We also assess the feasibility of acquiring components of +/-effect events from Chinese text using currently available resources. In the future, it is promising to utilize +/-effect event information to assist sentiment analysis in Chinese.

9.0 FUTURE DIRECTIONS

There are several remaining research questions that deserve consideration in the future. First, a better local system is expected to provide more reliable input to the joint prediction model. For example, a more confident result of recognized explicit sentiments is expected to infer more confident results of implicit sentiments. In the previous chapter, we have developed a system to improve recognizing sources of opinions. As the experimental results have shown, recognizing the sources is still a difficult task. Improving each individual component of the whole joint model is very important as future works.

Not limited to improving explicit sentiment analysis, the sentiment analysis overall can be improved from other aspects. Now let's review what resources we have used to accomplish this goal. Recall the example (Ex1.1) in Chapter 1 at the very beginning of this thesis.

(Ex1.1) It is great that Hillary Clinton defeated Donald Trump.

The sentiment lexicons are used to recognize *great* as a positive opinion, the semantic role labeling features are used to recognize the target is the *defeated* event (Yang and Cardie, 2013b), and then the inference rules are used to recognize the writer is positive toward Hillary Clinton and negative toward Donald Trump since the writer is positive toward the defeating event which harms Donald Trump (Deng et al., 2014; Deng and Wiebe, 2015a). These works mainly rely on the clues that directly indicate opinions (e.g., recognizing *great* as a positive opinion), or indicate components of opinions (e.g., recognizing the target being *defeated*), or indicate other opinions based on the information within the sentence (e.g., recognizing a positive opinion toward *Hillary Clinton*). They do not exploit the vast amount of knowledge outside the sentence, which are outputs from many NLP tasks. But the task of sentiment analysis may benefit from those tasks.

Consider (Ex8.1), for example.

(Ex8.1) President Obama proposed the healthcare reform. I support him.

we recognize in the second the sentence that the writer (i.e., I) is positive toward him. Further, we recognize the writer is positive toward President Obama since by co-reference resolution we know that him refers to President Obama.

Meanwhile, other NLP tasks may benefit from sentiment analysis. Consider, for example,

(Ex8.2) The allies successfully defeated Nazi. They are really brave.

The sentiment analysis system may infer that the writer is positive toward the allies and negative toward the Nazi. Based on this information, we can infer that they in the second sentence refer to the allies instead of the Nazi. Thus the sentiment analysis outputs help the co-reference resolution task.

The relation of sentiment analysis and other NLP tasks cannot be easily modelled as a pipeline. For example, in (Ex8.1) a co-reference resolution needs to be run first to infer the writer is positive toward Obama, while in (Ex8.2) the positive sentiment toward the allies needs to be recognized first to infer that they refer to the allies.

In Chapter 6 we have developed joint models to infer sentiments based on the sentiment inference rules (e.g, (Ex1.1) mentioned above). We first develop individual systems to recognize sentiments and components of sentiments. Then joint approaches are used to take the outputs from independent systems as input and globally infer sentiments based on all the input information. The sentiment inference rules are used as constraints in the joint approaches. Similar to the joint models, we can define more inference rules to be used as constraints in the joint models, which is promising to jointly resolve sentiment analysis and more NLP tasks. Furthermore, though the representations of the knowledge that different NLP tasks generate are various, the dependencies defined by inference rules in this paper are expressed in a unified way: propositional logics.

In summary, this chapter presents the newly defined dependency rules to describe connections of different NLP tasks in order to exploit various kinds of knowledge to make progress toward a deeper interpretation of subjective language. The set of rules in this chapter is an

extension to the sentiment inference in this thesis, which serve as a promising exploration and future works of this thesis.

Previously in the sentiment inference rules, each literal corresponds to an NLP task. Similarly here each newly defined literal in the new rules corresponds to an NLP task. The new rules are grouped to two classes. One class contains rules that exploit the knowledge in the document (Section 9.1), the other class contains rules exploiting the knowledge outside the documents (Section 9.2). A rule will be presented as an instantiated rule applied to an example in the rest of this chapter.

9.1 RULES USING INTRA-DOCUMENT KNOWLEDGE

9.1.1 Co-reference Resolution

Recall (Ex8.1) at the beginning of the chapter. The writer is positive toward Obama because the word him refers to Obama. The instantiated rule is:

$$\textsc{PositivePair}(\text{writer},\text{him}) \wedge \textsc{sameEntity}(\text{him},\text{Obama}) \Rightarrow \textsc{PositivePair}(\text{writer},\text{Obama})$$

where we can use co-reference solutions to assign values to the literal SameEntity.

9.1.2 Agree

We may also infer that the writer has the same sentiments as sources with whom he or she agrees. Though many previous works detects agreement at the turn level in conversation (Michel Galley, 2004; Wang et al., 2011), or identifies participants who agree with one another (Hassan et al., 2012; Abu-Jbara et al., 2012; Park et al., 2011), there are recent works on detecting agreement within documents (Wang and Cardie, 2014; Abbott et al., 2011; Misra and Walker, 2013). Consider, *I agree with Paul. ... The plan is a brilliant idea.* The writer (I) agree with Paul, and the writer is positive toward the plan. Then we infer that probably Paul is positive toward the plan.

AGREE(writer,Paul) ∧ POSITIVEPAIR(writer,plan)
⇒ POSITIVEPAIR(Paul,plan)

9.1.3 Opinion-oriented Discourse Models

Previous works have developed opinion-oriented discourse models (*OODM*s) (Somasundaran, 2010). The OODM models recognize toward which entities the writer's sentiments are the same (SAMEENTITY), and toward which entities the writer's sentiments are opposite (ALTENTITY). The discourse *SameEntity* relation covers not only identity, but also part-whole, synonymy, generalization, specialization, entity-attribute/aspect, instantiation, cause-effect, and implicit background topic, i.e., relations that have been studied by many researchers in the context of anaphora and co-reference (Clark, 1975; Vieira and Poesio, 2000; Mueller and Strube, 2001, etc). Two entities are in an ALTENTITY relation if they are mutually exclusive options in the context of the discourse. For example, in a debate about mobile phones, the iPhone and iOS are considered as SAMEENTITY, while the Android and iPhone are considered as ALTENTITY. In OODM models, same sentiments toward same entities express the same stance, and opposite sentiments toward alternative targets express the same overall stance (Somasundaran, 2010).

POSITIVEPAIR(writer,iOS) ∧ SAMEENTITY(iOS,iPhone)
⇒ POSITIVEPAIR(writer,iPhone)
POSITIVEPAIR(writer,iOS) ∧ ALTENTITY(iOS,Android)
⇒ NEGATIVEPAIR(writer,Android)

However, the opinions throughout the documents may not always be consistent. In the same document, a source may be both positive and negative toward a target. In this chapter, we define **rules to explain conflicting opinions** in the document.

9.1.4 Aspect-Based Sentiment Analysis

In one case, the source has different opinions about different aspects of the same target. Consider *The iPhone display is beautiful. But it is too expensive.* The writer is positive toward the display while negative toward the price. Such case can be modelled via the rule:

PositivePair(writer,iPhone) ∧ NegativePair(writer,it) ∧ SameEntity(iPhone,it) ⇔
Aspect(iPhone,display) ∧ PositivePair(writer,display) ∧
Aspect(it,price) ∧ NegativePair(writer,price)

Several researchers have focused on the task of mining data to discover aspects of products and sentiments toward different aspects (Liu, 2012).

9.1.5 (Non-)Reinforcing Sentiment Analysis.

In the other case, people may be ambivalent, or change their minds in the course of a document. Two sentiments may be in *reinforcing* or *non-reinforcing* discourse scenarios. Reinforcing relations exist between opinions when they contribute to the same overall stance. Non-reinforcing relations exist between opinions that show ambivalence, which represents a discourse scenario in which inconsistent sentiments are expressed with respect to a stance (Somasundaran, 2010; Trivedi and Eisenstein, 2013; Bhatia et al., 2015). Consider, *It is expensive. ... However, I think it is worth a try if I loan to buy the phone.* Previous works (Somasundaran, 2010) may recognize that two non-reinforcing sentiments occur (indicated by however). *S1* represents the negative opinion in the first sentence expressed toward it, and *S2* represents the positive opinion in the second sentence expressed toward the phone.

NonReinforcing(S1,S2) ∧
Source(S1,writer) ∧ Source(S2,writer) ∧
Target(S1,It) ∧ Target(S2,the phone) ∧
SameEntity(It, the phone) ∧
NegativePair(writer,It)
⇒ PositivePair(writer,the phone)

Two non-reinforcing opinions can also be expressed toward alternative entities. Consider, *The iPhone is too expensive. ... But the price of Android cannot guarantee a satisfactorily smooth operating system. S1* represents the negative opinion in the first sentence expressed toward iPhone, and *S2* represents the negative opinion in the second sentence expressed toward Android. *S1* and *S2* are non-reinforcing sentiments, as indicated by the word but. Thus, even though it is difficult to recognize the negative opinion in the second sentence, the non-reinforcing relation can help us via the following rule.

NonReinforcing(S1,S2) ∧
Source(S1,writer) ∧ Source(S2,writer) ∧
Target(S1,iPhone) ∧ Target(S2,Android) ∧
AltEntity(iPhone, Android)
∧ NegativePair(writer,iPhone)
⇒ NegativePair(writer,Android)

9.2 RULES USING EXTRA-DOCUMENT KNOWLEDGE

9.2.1 Entity Linking.

Knowledge from outside the document is also important. For example, the works in entity linking map entity mentions (e.g.,Obama, US President) in the text to entries in the knowledge base (e.g., *Barack Obama*) (Ji and Grishman, 2011; Rao et al., 2013). Such information can be exploited to recognize SameEntity, as shown below.

SameEntity(Obama, *Barack Obama*) ∧ SameEntity(US President, *Barack Obama*)
⇒ SameEntity(Obama, US President)

Thus, we can use the knowledge base to enrich the recognition of SameEntity and help recognize more sentiments.

9.2.2 Ideology.

Groups of people sharing the same ideology tend to have the same opinions about certain things. Suppose we have known that Donald Trump is conservative, and a conservative ideology is against the concept of gun control, then we probably infer that he is opposed to gun control in the context.

Ideology(Donald Trump,*Conservative*) ∧
NegativePair(*Conservative*,*Gun Control*) ∧
SameEntity(*Gun Control*, gun control)
⇒ NegativePair(Donald Trump,gun control)

Rather than attempt to computationally define a general notion of ideology, people in NLP tend to use data for which specific ideologies have been defined. Previous works have

studied recognizing ideologies including political party affiliation (Iyyer et al., 2014), or labels such as *left*, *right*, and *center* (Sim et al., 2013), or use a proxy for ideology such as voting record (Gerrish and Blei, 2011).

9.3 SUMMARY

In short, the sentiment analysis task is not an isolated task. In the previous chapters, we have used the information from other NLP tasks such as semantic role labeling to help improve sentiment analysis. Not limited to improving the performances of NLP tasks mentioned in the previous chapters which is expected to contribute to a better sentiment analysis system, we can also investigate using the information from more NLP tasks. This chapter discusses the possibilities that many other NLP tasks can be used to improve sentiment analysis by providing the dependency rules between the other NLP tasks and sentiment analysis itself. The main idea here is that we can use different kinds of knowledge (both intra-document knowledge as stated in Section 9.1 and extra-document knowledge as stated in Section 9.2), including co-reference resolution, opinion discourse analysis, entity linking and ideology, etc. Moreover we are not talking about arbitrary NLP tasks but we have provided some works in the literature so that these goals can be achieved using state-of-the-art works. Though we haven't conducted experiments validating the rules, they are promising to bridge different tasks of sentiment analysis and various tasks in NLP together to provide a holistic approach to sentiment analysis and the other tasks as well.

10.0 CONCLUSIONS

This thesis focuses on entity/event-level sentiment detection and inference. The source of a sentiment is the writer or an entity in the text. The target of a sentiment is an entity or event. The sentiment is not necessarily expressed via a sentiment expression (explicit sentiments), but it can be inferred by inference rules (implicit sentiments). We mainly investigate computational models to automatically recognize both explicit and implicit sentiments expressed in the sentences. The thesis is shaped around six main hypotheses.

First of all, we define the representations of the new task in this thesis, the **entity/event-level sentiment analysis**. We define the representations of the sentiments that this task aims to identify. There are two elements that are well defined for the first time in this thesis. (1) The first element includes the definitions of explicit sentiment and implicit sentiment, and the differences between them. Briefly speaking, an explicit sentiment is always associated with an opinion expression linking the source to the target of the sentiment, while an implicit sentiment is not directly associated with an opinion expression. We have developed representations to depict both explicit sentiments and implicit sentiments. The representations are fundamental to the new task. (2) The second element includes the definitions of +/-effect events. The inferences arise from interactions between explicit sentiment expressions and events such as defeated in (Ex1.1) in Chapter 1, which negatively affect the themes of the events (-effect events), and events such as stand by, which positively affect the themes of the events (+effect events). Though a few works in sentiment analysis have utilized similar events as clues to find more sentiments (Zhang and Liu, 2011; Anand and Reschke, 2010; Reschke and Anand, 2011; Goyal et al., 2012), we are the first to give a full definition of such events in the thesis.

Hypothesis 1. *Annotation schemes can be developed to guide annotators to reliably annotate expressions of +/-effect events, the agents, and the themes, and their attributes.*
Hypothesis 2. *Annotation schemes can be developed to guide annotators to reliably annotate expressions of both entity/event-level explicit and implicit sentiments and their attributes.*

Next, we introduce the corpora we have developed to serve as the resources for this thesis and further research. The first corpus, *+/-Effect Event Sentiment Corpus*, is annotated with the writer's sentiments toward the agents and themes of +/-effect events (Deng et al., 2013). It fills in a gap by presenting an annotation scheme for +/-effect events and the writer's sentiments toward the agents and themes of those events. We have conducted an agreement study, the results of which are positive. Further, we have carried out consensus study providing a better estimation of how many disagreements were caused by negligence.

As a further step, we develop the MPQA 3.0 corpus (Deng and Wiebe, 2015b) by adding entity and event target annotations to the existing MPQA 2.0 corpus. Building upon the existing well-annotated corpus saves us much time in annotating opinions from scratch, and it preserves the original annotation structures in the existing corpus. We have designed a good annotation scheme as a transition from phrase-level sentiment corpora to entity/event-level sentiment corpora. Similarly, we have conducted an agreement study, the results of which are positive.

Both corpora are annotated with both explicit and implicit sentiments. The positive results of the agreement studies have provided supporting evidence for the first two hypotheses. Further, the positive results of the agreement studies have provided positive evidence that both corpora are promising to serve as good resources for the research focusing on identifying implicit sentiments.

Hypothesis 3. *Inferences to perceive implicit sentiments can be represented to build automatic systems.*

Two sets of sentiment inference rules are defined in the thesis. The rules are expressed in propositional logics. The first set of rules fills in the gap of the +/-effect event information and the sentiments expressed toward them. Basically, if a sentiment is expressed toward one component of the ⟨agent, +/-effect event, theme⟩, then the sentiments toward

the other two components can be inferred. The sentiment inference rules are novel in that they show dependencies between the outputs from the information extraction field which are entities, events and relations and the outputs from the sentiment analysis field. Based on the sentiment inference rules about sentiments toward +/-effect event information, we further define the second set of rules in Section 5.2. The second set of rules infers more sentiments in the *sentiment toward sentiment* structure. Though the superficial form of the second set of rules is similar to the first set of rules, the two sets are from different perspective. The first set of rules combines different NLP tasks, while the second set of rules dives deeper into the sentiment analysis outputs. Later we will show that both sets of rules are useful in improving recognizing entity/event-level sentiments. Since there are many ways to present the rules and it is consistent for us to use the representations we have defined, we do not evaluate the form of the rules. However, we evaluate whether the rules can help sentiment analysis in the later hypothesis.

Hypothesis 4. *+/-Effect event information is conductive to sentiment inference.*

We build a graph-based model defined by the sentiment inference rules to infer the sentiments expressed toward the agents and themes. We carry out an intrinsic evaluation of the inference ability of the rules. We find it has an 89% chance of propagating sentiments correctly. This is a good indicator that the rules give the correct inference in most context. For extrinsic evaluations, we introduce the fifth hypothesis.

Hypothesis 5. *Joint prediction models can be developed and can improve automatically recognizing entity/event-level sentiments.*

To test the fifth hypothesis, we have developed joint models to automatically recognize entity/event-level sentiments. We mainly investigate computational models for two tasks. First, as a pilot study, we focus on recognizing the writer's sentiments expressed toward the agents and themes of +/-effect events. The experiments have demonstrated that our model improves over local sentiment recognition by almost 20 points in F-measure and over all sentiment baselines by over 10 points in F-measure. The success of the pilot study encourages us to extend the model. Next, we focus on a more complicated task that is recognizing the sentiments whose sources can be the writer or any entity in the text and the

targets are entities or events. The model builds upon state-of-the-art span-based sentiment analysis systems to perform entity/event-level sentiment analysis covering both explicit and implicit sentiments expressed among entities and events in texts. The experiments have shown that the model jointly disambiguates the ambiguities in the opinion components and improve over baseline accuracies in recognizing entity/event-level sentiments.

The computational models in both tasks are joint prediction models. The input to the joint models are the results of individual NLP tasks, and the output of the joint models are answers to each individual task which are globally optimized. The joint models are constrained by the inference rules. As the experiments have shown, in addition to beating the baselines for sentiment detection, the joint model significantly improves the accuracy of +/-effect sense disambiguation without any loss in accuracy of the remaining tasks. The exciting conclusions drawn from this chapter include: 1) the inference rules define useful dependencies among different NLP tasks; 2) the joint prediction models defined by the inference rules are promising to jointly resolve the ambiguities in different NLP tasks and improve the performances in more than one tasks.

Hypothesis 6. *Categorizing opinions according to whether the source is a participant of the opinion or not can improve the recognition of opinion sources.*

To test the sixth hypothesis, we first formally give the definition of the new categorization. The source of a participant opinion participates the event that triggers the opinion, while the source of a non-participant opinion does not participate the event of the opinion. (Usually, the source is someone who states the opinion.) Based on this categorization, we develop a binary classifier trained on the existing limited resources to judge which type of the two that a given opinion is. Based on the classification result, we develop a model to correctly choose the source from several candidate sources which are automatically generated via different heuristics and methods. The experiments have shown that the new categorization can help improve recognizing sources. Another conclusion drawn from the experiment is that fully utilizing the existing resources by using transductive methods can give a better result for the downstream applications.

Hypothesis 7. *Inferences are not limited to English text only.*

To test the final hypothesis, we follow the same experimental paradigms used to test the hypothesis mentioned above. We carry out the same experiment used to test Hypothesis 3 to test this hypothesis. The positive results have provided evidence that such implicit opinions and inference rules are similar in Chinese and English. There are some +/-effect events caused by the Chinese syntax, guidance for which could be added to the current English manual to develop a Chinese manual. Moreover, there is no evidence showing that the blocked inferences only happen in Chinese. We also assess the feasibility of acquiring components of +/-effect events from Chinese text using currently available resources.

To summarize, this thesis has devoted to sentiment analysis by providing resources and models to infer implicit sentiments. Specifically, we first define a new sentiment analysis task (entity/event-level sentiment analysis task). We develop annotated corpora as the resources of the task and investigate joint prediction models integrating explicit sentiments, entity or event information, and inference rules together to automatically recognize both explicit and implicit sentiments expressed among entities and events in the text.

APPENDIX A

SENTIMENT REPRESENTATION RULES

(Rule1.1) $+\textsc{Sentiment}(y) \wedge \textsc{Source}(y,s) \wedge \textsc{Target}(y,t) \Rightarrow \textsc{PositivePair}(s,t)$

(Rule1.2) $-\textsc{Sentiment}(y) \wedge \textsc{Source}(y,s) \wedge \textsc{Target}(y,t) \Rightarrow \textsc{NegativePair}(s,t)$

Table A1: Sentiment Representation Rules.

APPENDIX B

SENTIMENT INFERENCE RULES W.R.T. +/-EFFECT EVENT

(Rule 2.1)	PositivePair(s,x) $\wedge$ Agent(x,a) $\wedge$ +Effect(x)	$\Rightarrow$ PositivePair(s,a)
(Rule 2.2)	PositivePair(s,x) $\wedge$ Agent(x,a) $\wedge$ -Effect(x)	$\Rightarrow$ PositivePair(s,a)
(Rule 2.3)	PositivePair(s,x) $\wedge$ Theme(x,h) $\wedge$ +Effect(x)	$\Rightarrow$ PositivePair(s,h)
(Rule 2.4)	PositivePair(s,x) $\wedge$ Theme(x,h) $\wedge$ -Effect(x)	$\Rightarrow$ NegativePair(s,h)
(Rule 2.5)	NegativePair(s,x) $\wedge$ Agent(x,a) $\wedge$ +Effect(x)	$\Rightarrow$ NegativePair(s,a)
(Rule 2.6)	NegativePair(s,x) $\wedge$ Agent(x,a) $\wedge$ -Effect(x)	$\Rightarrow$ NegativePair(s,a)
(Rule 2.7)	NegativePair(s,x) $\wedge$ Theme(x,h) $\wedge$ +Effect(x)	$\Rightarrow$ NegativePair(s,h)
(Rule 2.8)	NegativePair(s,x) $\wedge$ Theme(x,h) $\wedge$ -Effect(x)	$\Rightarrow$ PositivePair(s,h)
(Rule 2.9)	PositivePair(s,a) $\wedge$ Agent(x,a) $\wedge$ +Effect(x)	$\Rightarrow$ PositivePair(s,x)
(Rule 2.10)	PositivePair(s,a) $\wedge$ Agent(x,a) $\wedge$ -Effect(x)	$\Rightarrow$ PositivePair(s,x)
(Rule 2.11)	PositivePair(s,h) $\wedge$ Theme(x,h) $\wedge$ +Effect(x)	$\Rightarrow$ PositivePair(s,x)
(Rule 2.12)	PositivePair(s,h) $\wedge$ Theme(x,h) $\wedge$ -Effect(x)	$\Rightarrow$ NegativePair(s,x)
(Rule 2.13)	NegativePair(s,a) $\wedge$ Agent(x,a) $\wedge$ +Effect(x)	$\Rightarrow$ NegativePair(s,x)
(Rule 2.14)	NegativePair(s,a) $\wedge$ Agent(x,a) $\wedge$ -Effect(x)	$\Rightarrow$ NegativePair(s,x)
(Rule 2.15)	NegativePair(s,h) $\wedge$ Theme(x,h) $\wedge$ +Effect(x)	$\Rightarrow$ NegativePair(s,x)
(Rule 2.16)	NegativePair(s,h) $\wedge$ Theme(x,h) $\wedge$ -Effect(x)	$\Rightarrow$ PositivePair(s,x)

Table B1: Sentiment Inference Rules w.r.t +/-Effect Event

APPENDIX C

SENTIMENT INFERENCE RULES W.R.T. SENTIMENT-TOWARD-SENTIMENT STRUCTURE

(Rule 3.1)	PositivePair(s_1,y_2) $\wedge$ Source(y_2,s_2) $\wedge$ +Sentiment(y_2)	$\Rightarrow$ PositivePair(s_1,s_2)
(Rule 3.2)	PositivePair(s_1,y_2) $\wedge$ Source(y_2,s_2) $\wedge$ -Sentiment(y_2)	$\Rightarrow$ PositivePair(s_1,s_2)
(Rule 3.3)	PositivePair(s_1,y_2) $\wedge$ eTarget(y_2,t_2) $\wedge$ +Sentiment(y_2)	$\Rightarrow$ PositivePair(s_1,t_2)
(Rule 3.4)	PositivePair(s_1,y_2) $\wedge$ eTarget(y_2,t_2) $\wedge$ -Sentiment(y_2)	$\Rightarrow$ NegativePair(s_1,t_2)
(Rule 3.5)	NegativePair(s_1,y_2) $\wedge$ Source(y_2,s_2) $\wedge$ +Sentiment(y_2)	$\Rightarrow$ NegativePair(s_1,s_2)
(Rule 3.6)	NegativePair(s_1,y_2) $\wedge$ Source(y_2,s_2) $\wedge$ -Sentiment(y_2)	$\Rightarrow$ NegativePair(s_1,s_2)
(Rule 3.7)	NegativePair(s_1,y_2) $\wedge$ eTarget(y_2,t_2) $\wedge$ +Sentiment(y_2)	$\Rightarrow$ NegativePair(s_1,t_2)
(Rule 3.8)	NegativePair(s_1,y_2) $\wedge$ eTarget(y_2,t_2) $\wedge$ -Sentiment(y_2)	$\Rightarrow$ PositivePair(s_1,t_2)

Table C1: Sentiment Inference Rules w.r.t. Sentiment-Toward-Sentiment Structure.

BIBLIOGRAPHY

Abbott, R., Walker, M., Anand, P., Fox Tree, J. E., Bowmani, R., and King, J. (2011). How can you say such things?!?: Recognizing disagreement in informal political argument. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 2–11, Portland, Oregon. Association for Computational Linguistics.

Abu-Jbara, A., Dasigi, P., Diab, M., and Radev, D. (2012). Subgroup detection in ideological discussions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 399–409, Jeju Island, Korea. Association for Computational Linguistics.

Akkaya, C., Wiebe, J., and Mihalcea, R. (2009). Subjectivity word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 190–199, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anand, P. and Reschke, K. (2010). Verb classes as evaluativity functor classes. In *Interdisciplinary Workshop on Verbs. The Identification and Representation of Verb Features.*

Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Comput. Linguist.*, 34(4):555–596.

Bach, S. H., Huang, B., and Getoor, L. (2013). Learning latent groups with hinge-loss markov random fields. In *Inferning: ICML Workshop on Interactions between Inference and Learning.*

Beltagy, I., Erk, K., and Mooney, R. (2014). Probabilistic soft logic for semantic textual similarity. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1210–1219, Baltimore, Maryland. Association for Computational Linguistics.

Bhatia, P., Ji, Y., and Eisenstein, J. (2015). Better document-level sentiment analysis from rst discourse parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2212–2218, Lisbon, Portugal. Association for Computational Linguistics.

Boyd-Graber, J. and Resnik, P. (2010). Holistic sentiment analysis across languages: Multilingual supervised latent dirichlet allocation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 45–55, Cambridge, MA. Association for Computational Linguistics.

Breck, E., Choi, Y., and Cardie, C. (2007). Identifying expressions of opinion in context. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2683–2688, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Broecheler, M., Mihalkova, L., and Getoor, L. (2010). Probabilistic similarity logic. In *Uncertainty in Artificial Intelligence (UAI)*.

Chang, P.-C., Tseng, H., Jurafsky, D., and Manning, C. D. (2009). Discriminative reordering with chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pages 51–59. Association for Computational Linguistics.

Chao, Y. R. (1968). *A grammar of spoken Chinese*. Univ of California Press.

Choi, Y., Breck, E., and Cardie, C. (2006). Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 431–439, Stroudsburg, PA, USA. Association for Computational Linguistics.

Choi, Y. and Cardie, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 793–801, Honolulu, Hawaii. Association for Computational Linguistics.

Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. (2005). Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics.

Choi, Y. and Wiebe, J. (2014). +/-effectwordnet: Sense-level lexicon acquisition for opinion inference. In *EMNLP*, pages 1181–1191.

Choi, Y., Wiebe, J., and Deng, L. (2014). Lexical acquisition for opinion inference: A sense-level lexicon of benefactive and malefactive events. In *5th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*.

Clark, H. H. (1975). Bridging. *Theoretical issues in natural language processing . New York: Association for Computing Machinery*, page 6.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Conrad, A., Wiebe, J., Hwa, and Rebecca (2012). Recognizing arguing subjectivity and argument tags. In *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics*, ExProM '12, pages 80–88, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Das, D., Martins, A. F., and Smith, N. A. (2012). An exact dual decomposition algorithm for shallow semantic parsing with constraints. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 209–217. Association for Computational Linguistics.

Dasigi, P., Guo, W., and Diab, M. (2012). Genre independent subgroup detection in online discussion threads: A study of implicit attitude using textual latent semantics. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 65–69, Jeju Island, Korea. Association for Computational Linguistics.

Deng, L., Choi, Y., and Wiebe, J. (2013). Benefactive/malefactive event and writer attitude annotation. In *ACL 2013 (short paper)*. Association for Computational Linguistics.

Deng, L. and Wiebe, J. (2014a). An investigation for implicatures in chinese : Implicatures in chinese and in english are similar ! In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 8–17, Baltimore, Maryland. Association for Computational Linguistics.

Deng, L. and Wiebe, J. (2014b). Sentiment propagation via implicature constraints. In *Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2014)*.

Deng, L. and Wiebe, J. (2015a). Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 179–189, Lisbon, Portugal. Association for Computational Linguistics.

Deng, L. and Wiebe, J. (2015b). Mpqa 3.0: An entity/event-level sentiment corpus. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1323–1328, Denver, Colorado. Association for Computational Linguistics.

Deng, L. and Wiebe, J. (2016a). How can nlp tasks mutually benefit sentiment analysis? a holistic approach to sentiment analysis. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 53–59, San Diego, California. Association for Computational Linguistics.

Deng, L. and Wiebe, J. (2016b). Recognizing opinion sources based on a new categorization of opinion types. IJCAI'16.

Deng, L., Wiebe, J., and Choi, Y. (2014). Joint inference and disambiguation of implicit sentiments via implicature constraints. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 79–88, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Denis, P. and Baldridge, J. (2007). Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 236–243, Rochester, New York. Association for Computational Linguistics.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.

Elson, D. K. and McKeown, K. (2010). Automatic attribution of quoted speech in literary narrative. In *AAAI*. Citeseer.

Feng, S., Kang, J. S., Kuznetsova, P., and Choi, Y. (2013). Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria. Association for Computational Linguistics.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Gerrish, S. and Blei, D. M. (2011). Predicting legislative roll calls from text. In *ICML*.

Glass, K. and Bangay, S. (2007). A naive salience-based method for speaker identification in fiction books. In *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA'07)*, pages 1–6.

Goyal, A., Riloff, E., and III, H. D. (2012). A computational model for plot units. *Computational Intelligence*, pages 466–488.

Greene, S. and Resnik, P. (2009). More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 503–511, Boulder, Colorado. Association for Computational Linguistics.

Hassan, A., Abu-Jbara, A., and Radev, D. (2012). Detecting subgroups in online discussions by modeling positive and negative relations among participants. In *Proceedings of the*

2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 59–70, Jeju Island, Korea. Association for Computational Linguistics.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Huang, B., Bach, S. H., Norris, E., Pujara, J., and Getoor, L. (2012). Social group modeling with probabilistic soft logic. In *NIPS Workshop on Social Network and Social Media Analysis: Methods, Models, and Applications*.

Huang, B., Kimmig, A., Getoor, L., and Golbeck, J. (2013). A flexible framework for probabilistic models of social trust. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 265–273. Springer.

Irsoy, O. and Cardie, C. (2014). Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 720–728.

Iyyer, M., Enns, P., Boyd-Graber, J., and Resnik, P. (2014). Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1113–1122, Baltimore, Maryland. Association for Computational Linguistics.

Ji, H. and Grishman, R. (2011). Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1148–1158. Association for Computational Linguistics.

Ji, H. and Lin, D. (2009). Gender and animacy knowledge discovery from web-scale n-grams for unsupervised person mention detection. In *PACLIC*, pages 220–229.

Joachims, T. (1999a). Making large-scale SVM learning practical. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA.

Joachims, T. (1999b). Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209.

Johansson, R. and Moschitti, A. (2013a). Relational features in fine-grained opinion analysis. *Computational Linguistics*, 39(3).

Johansson, R. and Moschitti, A. (2013b). Relational features in fine-grained opinion analysis. *Computational Linguistics*, 39(3).

Klir, G. and Yuan, B. (1995). *Fuzzy sets and fuzzy logic*, volume 4. Prentice hall New Jersey.

Kozhevnikov, M. and Titov, I. (2013). Cross-lingual transfer of semantic role labeling models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1200, Sofia, Bulgaria. Association for Computational Linguistics.

Ku, L.-W. and Chen, H.-H. (2007). Mining opinions from the web: Beyond relevance retrieval. *Journal of the American Society for Information Science and Technology*, 58(12):1838–1850.

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., and Jurafsky, D. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Li, C. N. and Thompson, S. A. (1989). *Mandarin Chinese: A functional reference grammar*. Univ of California Press.

Li, J. and Sun, M. (2007). Experimental study on sentiment classification of chinese review using machine learning techniques. In *Natural Language Processing and Knowledge Engineering, 2007. NLP-KE 2007. International Conference on*, pages 393–400. IEEE.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Liu, K., Xu, L., and Zhao, J. (2013). Syntactic patterns versus word alignment: Extracting opinion targets from online reviews. In *ACL (1)*, pages 1754–1763.

Liu, K., Xu, L., and Zhao, J. (2014). Extracting opinion targets and opinion words from online reviews with graph co-ranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 314–324.

London, B., Khamis, S., Bach, S. H., Huang, B., Getoor, L., and Davis, L. (2013). Collective activity detection using hinge-loss Markov random fields. In *CVPR Workshop on Structured Prediction: Tractability, Learning and Inference*.

Lu, B., Tan, C., Cardie, C., and Tsou, B. K. (2011). Joint bilingual sentiment classification with unlabeled parallel corpora. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 320–330. Association for Computational Linguistics.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Martins, A. F. T. and Smith, N. a. (2009). Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming*

for Natural Langauge Processing - ILP '09, pages 1–9, Morristown, NJ, USA. Association for Computational Linguistics.

McDonald, R., Hannan, K., Neylon, T., Wells, M., and Reynar, J. (2007). Structured models for fine-to-coarse sentiment analysis. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 432. Citeseer.

Medlock, B. and Briscoe, T. (2007). Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

Memory, A., Kimmig, A., Bach, S., Raschid, L., and Getoor, L. (2012). Graph summarization in annotated data using probabilistic soft logic. In *Proceedings of the 8th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2012)*, volume 900, pages 75–86.

Michel Galley, Kathleen McKeown, J. H. E. S. (2004). Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics (ACL-2004)*.

Misra, A. and Walker, M. (2013). Topic independent identification of agreement and disagreement in social media dialogue. In *Proceedings of the SIGDIAL 2013 Conference*, pages 41–50, Metz, France. Association for Computational Linguistics.

Moilanen, K. and Pulman, S. (2007a). Sentiment composition. In *Proceedings of RANLP 2007*, Borovets, Bulgaria.

Moilanen, K. and Pulman, S. (2007b). Sentiment composition. In *Proceedings of RANLP*, volume 7, pages 378–382.

Moilanen, K., Pulman, S., and Zhang, Y. (2010). Packed feelings and ordered sentiments: Sentiment parsing with quasi-compositional polarity sequencing and compression. In *Proceedings of the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2010)*, pages 36–43.

Mueller, C. and Strube, M. (2001). Annotating anaphoric and bridging relations with mmax. In *2nd SIGdial Workshop on Discourse and Dialogue*.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2010). Recognition of affect, judgment, and appreciation in text. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 806–814, Stroudsburg, PA, USA. Association for Computational Linguistics.

O'Keefe, T., Pareti, S., Curran, J. R., Koprinska, I., and Honnibal, M. (2012). A sequence labelling approach to quote attribution. In *EMNLP*, pages 790–799. Association for Computational Linguistics.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Pareti, S., O'Keefe, T., Konstas, I., Curran, J. R., and Koprinska, I. (2013). Automatically detecting and attributing indirect quotations. In *EMNLP*, pages 989–999.

Park, S., Lee, K. S., and Song, J. (2011). Contrasting opposing views of news articles on contentious issues. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 340–349, Portland, Oregon, USA. Association for Computational Linguistics.

Pearl, J. (1982). Reverend bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the American Association of Artificial Intelligence National Conference on AI*, pages 133–136, Pittsburgh, PA.

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S. M., and Eryiğit, G. (2016). Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., and Androutsopoulos, I. (2015). Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado*, pages 486–495.

Pontiki, M. and Manandhar, S. (2014). Semeval-2014 task 4: Aspect based sentiment analysis. Citeseer.

Pujara, J., Miao, H., Getoor, L., and Cohen, W. (2013). Knowledge graph identification. In *The Semantic Web–ISWC 2013*, pages 542–557. Springer.

Punyakanok, V., Roth, D., and Yih, W.-t. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Punyakanok, V., Roth, D., Yih, W.-t., and Zimak, D. (2004). Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1346. Association for Computational Linguistics.

Quirk, R., Greenbaum, S., Leech, G., Svartvik, J., and Crystal, D. (1985). *A comprehensive grammar of the English language*, volume 397. Cambridge Univ Press.

Rao, D., McNamee, P., and Dredze, M. (2013). Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, multilingual information extraction and summarization*, pages 93–115. Springer.

Recasens, M., de Marneffe, M.-C., and Potts, C. (2013). The life and death of discourse entities: Identifying singleton mentions. In *HLT-NAACL*, pages 627–633.

Reschke, K. and Anand, P. (2011). Extracting contextual evaluativity. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pages 370–374, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine learning*, 62(1-2):107–136.

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. (2013a). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. (2013b). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.

Roth, D. and Yih, W.-t. (2004). A linear programming formulation for global inference in natural language tasks. In *CONLL*.

Scholz, T. and Conrad, S. (2013). Opinion mining in newspaper articles by entropy-based word connections. In *EMNLP*, pages 1828–1839.

Sim, Y., Acree, B. D. L., Gross, J. H., and Smith, N. A. (2013). Measuring ideological proportions in political speeches. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 91–101, Seattle, Washington, USA. Association for Computational Linguistics.

Socher, R., Huang, E. H., Pennin, J., Manning, C. D., and Ng, A. Y. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Citeseer.

Somasundaran, S. (2010). *Discourse-Level Relations for Opinion Analysis*. PhD thesis, Department of Computer Science, University of Pittsburgh.

Somasundaran, S. and Wiebe, J. (2009). Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 226–234, Suntec, Singapore. Association for Computational Linguistics.

Stone, P., Dunphy, D., Smith, M., and Ogilvie, D. (1966). *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge.

Stoyanov, V., Cardie, C., Gilbert, N., Riloff, E., Buttler, D., and Hysom, D. (2010). Coreference resolution with reconcile. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 156–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stoyanov, V., Cardie, C., and Wiebe, J. (2005). Multi-perspective question answering using the opqa corpus. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 923–930. Association for Computational Linguistics.

Sybesma, R. P. E. (1992). *Causatives and accomplishments: The case of Chinese ba*, volume 1. Holland Institute of Generative Linguistics.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565.

Titov, I. and McDonald, R. T. (2008). A joint model of text and aspect ratings for sentiment summarization. In *ACL*, volume 8, pages 308–316. Citeseer.

Trivedi, R. and Eisenstein, J. (2013). Discourse connectors for latent subjectivity in sentiment analysis. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 808–813, Atlanta, Georgia. Association for Computational Linguistics.

Tsou, B. K., Yuen, R. W., Kwong, O. Y., La, T., and Wong, W. L. (2005). Polarity classification of celebrity coverage in the chinese press. In *Proceedings of International Conference on Intelligence Analysis*.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media.

Vieira, R. and Poesio, M. (2000). An empirically based system for processing definite descriptions. *Computational Linguistics*, 26(4):539–593.

Wan, X. (2008). Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 553–561, Honolulu, Hawaii. Association for Computational Linguistics.

Wan, X. (2009). Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 235–243. Association for Computational Linguistics.

Wang, L. and Cardie, C. (2014). Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 97–106, Baltimore, Maryland. Association for Computational Linguistics.

Wang, S., Wei, Y., Li, D., Zhang, W., and Li, W. (2007). A hybrid method of feature selection for chinese text sentiment classification. In *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, volume 3, pages 435–439. IEEE.

Wang, W., Yaman, S., Precoda, K., Richey, C., and Raymond, G. (2011). Detection of agreement and disagreement in broadcast conversations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 374–378, Portland, Oregon, USA. Association for Computational Linguistics.

Wiebe, J. and Deng, L. (2014). An account of opinion implicatures. *arXiv*, 1404.6491[cs.CL].

Wiebe, J., Wilson, T., Bruce, R., Bell, M., and Martin, M. (2004). Learning subjective language. *Computational linguistics*, 30(3):277–308.

Wiebe, J., Wilson, T., and Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2/3):164–210.

Wiebe, J. M., Bruce, R. F., and O'Hara, T. P. (1999). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 246–253. Association for Computational Linguistics.

Wiegand, M. and Klakow, D. (2010). Convolution kernels for opinion holder extraction. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 795–803. Association for Computational Linguistics.

Wiegand, M. and Klakow, D. (2012). Generalization methods for in-domain and cross-domain opinion holder extraction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 325–335. Association for Computational Linguistics.

Wilson, T. (2008). *Fine-grained subjectivity analysis*. PhD thesis, Doctoral Dissertation, University of Pittsburgh.

Wilson, T., Wiebe, J., , and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *HLP/EMNLP*, pages 347–354.

Wilson, T. and Wiebe, J. (2003). Annotating opinions in the world press. In *Proceedings of the 4th ACL SIGdial Workshop on Discourse and Dialogue (SIGdial-03)*, pages 13–22.

Xu, L., Liu, K., Lai, S., Chen, Y., and Zhao, J. (2013). Mining opinion words and opinion targets in a two-stage framework. In *ACL (1)*, pages 1764–1773.

Xue, N. and Palmer, M. (2004). Calibrating features for semantic role labeling. In *EMNLP*, pages 88–94.

Yang, B. and Cardie, C. (2013a). Joint Inference for Fine-grained Opinion Extraction. In *Proceedings of ACL*, pages 1640–1649.

Yang, B. and Cardie, C. (2013b). Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649, Sofia, Bulgaria. Association for Computational Linguistics.

Yang, B. and Cardie, C. (2014). Joint modeling of opinion expression extraction and attribute classification. *Transactions of the Association for Computational Linguistics*, 2:505–516.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312.

Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136. Association for Computational Linguistics.

Zhang, L. and Liu, B. (2011). Identifying noun product features that imply opinions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 575–580, Portland, Oregon, USA. Association for Computational Linguistics.

Zhou, X., Wan, X., and Xiao, J. (2013). Collective opinion target extraction in chinese microblogs. In *EMNLP*, pages 1840–1850.