Fine-grained Opinion Analysis: Structure-aware Approaches

by Ye Jin Choi

FINE-GRAINED OPINION ANALYSIS: STRUCTURE-AWARE APPROACHES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Ye Jin Choi

August 2010

FINE-GRAINED OPINION ANALYSIS: STRUCTURE-AWARE APPROACHES

Ye Jin Choi, Ph.D.

Cornell University 2010

Natural language reflects the affective nature of the human mind. Accordingly, expressions of affect and opinion appear profusely in natural language utterances — either explicitly or implicitly. Recognizing and interpreting the subjective information, beyond factual information such as topics and events thereby constitute an important aspect of natural language understanding. Indeed in recent years, there has been a great surge of research interest to help computers understand the subjective side of natural language.

In this dissertation, we explore computational methods that can push the envelope for sentiment analysis in text. There are two distinctive themes in our contributions: First, our focus will be on fine-grained opinion analysis, which has been relatively less explored than coarse-grained analysis (e.g., document-level classification). Second, the approaches developed in our work are structure-aware in that we design the inference and/or learning algorithms reflecting the task-specific linguistic structure. We tackle five different sets of problems under these themes, and the key results are summarized in the paragraphs below:

Joint Extraction of Opinion Elements and Relations: In this work, we present a system for extracting fine-grained opinion elements such as *opinion expressions* and the *sources of opinions*, and the relations among those elements, using machine learning techniques and integer linear programming. The extracted opin-

ion elements can then be used as building blocks for various opinion applications, such as opinion summarization or opinion-oriented question answering.

Joint Extraction of Opinions and their Attributes: We recognize that the task of determining polarity is related to the task of determining intensity. Based on this observation, we develop a hierarchical sequential learning technique to extract opinion expressions and their attributes – polarity and intensity – simultaneously.

Polarity Inference in light of Compositional Semantics: In this work, we investigate methods for fine-grained polarity classification by drawing a connection to *compositional semantics*, one of the classic branches of research across linguistics and logic. This work attempts to bridge the gap between theories in compositional semantics and practical approaches based on machine learning techniques, by incorporating simple *compositional* rules based on syntactic patterns as structural inference for the learning algorithm.

Lexicon Adaptation as Constraint Optimization: Although there has been plentiful research in the creation of lexical resources for sentiment analysis, most is conducted in isolation from actual applications. As a result, a purportedly better lexical resource might not lead to better performance when utilized for a specific natural language application. To address this problem, we develop a method that adapts a general-purpose polarity lexicon into a domain-specific one in the context of a specific NLP task, by casting the problem as a constraint optimization problem using integer linear programming.

Structured local training for coreference resolution: Once we have identified fine-grained opinion elements in text, we need to determine whether some

of the extracted phrases are referring to an identical entity – namely, coreference resolution. In this work, we develop "structured local training", a machine learning technique based on Conditional Random Fields (CRFs) that directly incorporates the interaction between local decisions and global decisions into the learning procedure. We also propose "biased potential functions" that can empirically drive CRFs towards performance improvements with respect to the preferred evaluation measure.

BIOGRAPHICAL SKETCH

Ye Jin Choi is originally from Korea. She received her B.S. degree in Computer Engineering and Science from Seoul National University, and Ph.D. degree in Computer Science from Cornell University.

Bear, Bear, Bear

ACKNOWLEDGEMENTS

First and foremost, my biggest thank goes to my advisor Claire Cardie. Looking back, I arrived at Cornell as such a rough stone, and it is Claire who has introduced me to the fascinating field of Natural Language Processing, and inspired me and nurtured me to become an independent researcher. I would have not been able to come this far, if it were not Claire's endless support and infinitely patient encouragement. I am eternally indebted to her, and I wish to follow her foot step in inspiring and helping other people.

I would like to thank other faculty who had a positive influence on me: I thank Lillian Lee for giving me invaluable and strangely powerful advice at right moments. I will miss her passion and vigor for research. I am very much grateful to Ellen Riloff for her very insightful advice, which was as if she was reading my mind. I thank Johannes Gehrke for being a very kind and joyful committee member, and also for helpful and encouraging remarks. I also thank my minor committee member, David Williamson, for his amazing class 'Approximation Algorithms', which turned out to be more relevant to my research than I imagined. Yet another thank goes to Joe Halpern for being such a wonderful help when I first arrived at Cornell and about to form my research direction. Ken Birman was another source of great advice, some of which I engraved to my heart. I will miss his summer parties at lake house. I also thank Bart Selman for the Advanced Artificial Intelligence class that I particularly enjoyed, and also for giving me very insightful advice regarding research at the beginning of my journey at Cornell. I thank Thorsten Joachims and Jon Kleinberg for a number of helpful comments, and John Hale and Mats Rooth for sharing their insights at our NLP seminar. I am grateful to Jeff Hancock for introducing me to the world of lie detection, and opening my eyes to interdisciplinary research.

I would like to thank everyone who came to the NLP and MLDG seminar for many lively and insightful discussions. I would like to thank Bo Pang for very helpful professional advice over a number of occasions, and researchers at Yahoo! Research for inviting me as an intern in 2009.

My life in Ithaca has been a very pleasant one due to many wonderful friends. One of my deepest thanks should go to Maya. I especially thank her for choosing me as an officemate to sit right next to her even before knowing me. It was incredibly kind of her and Ilya to share the house with me during my internship at Yahoo! Research in 2009, as it really helped me to survive from the sadness to be taken away from my dear husband for the first time. And as it turns out, Ilya is a great source of exciting insight and wisdom about pretty much anything in life, and I truly enjoyed my summer in California around this couple. I will conclude my expression of gratitude to Maya by sending her a "moo"! I thank Art for all the laughters I had due to his smart humor. I must say I also learned a lot from him for his communication and presentation skills. I thank Cristian for being another great source of laughters, and also being a great inspiration for research. I thank Ainur for being a great colleague as well as a friend. I will miss Yisong for his insight and passion for research as well as life. I also thank Mohamed for being very supportive throughout the initial class work and for skillfully making fun of my worries so that I worry less. I thank Ves for being always so optimistic, and being a great friend throughout my stay at Cornell. I much enjoyed a lot of thought-provoking conversations with Yookyung about research as well as Korea.

What I became today was not possible without the support and love from my family: I particularly thank my parents for giving me the strength and energy, and also thank Marylka and Tadek for treating me like their own daughter.

Last but not least, my achievement would not mean anything without my dear husband Krzys. Despite the hard times I had to go through during the course of my Ph.D. degree, Ithaca has been one little paradise for me because of him.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 A Brief Overview

Natural language reflects the affective nature of the human mind. Accordingly, expressions of affect and opinion appear profusely in natural language utterances — either explicitly or implicitly (e.g. Wiebe et al. (2005), Pinker (2007), Greene and Resnik (2009)). Recognizing and interpreting the subjective information, beyond factual information such as topics and events thereby constitute an important aspect of natural language understanding. Even the newspaper articles, whose apparent purpose is delivering factual news, are rich in opinions for two different reasons. First, the content of news frequently includes news-worthy opinions of notable people or organizations. Second, the particular choice of wording and syntactic frames often reveal the affective state or viewpoint of the author toward the topic of the news. As a concrete example, consider the following headlines taken from three different news agencies:

a. Fact checking Sarah Palin's [Going Rogue].[1]

b. Palin's book goes rogue on some facts.[2]

c. The AP goes rogue fact checking Sarah Palin.[3]

Although all of these headlines consist of very similar sets of words, they reflect drastically different opinions (i.e., political perspectives) of different news

[1] CBS News, Nov 17 2009.
[2] Associated Press, Nov 13 2009.
[3] www.AmericanThinker.com, Nov 14 2009.

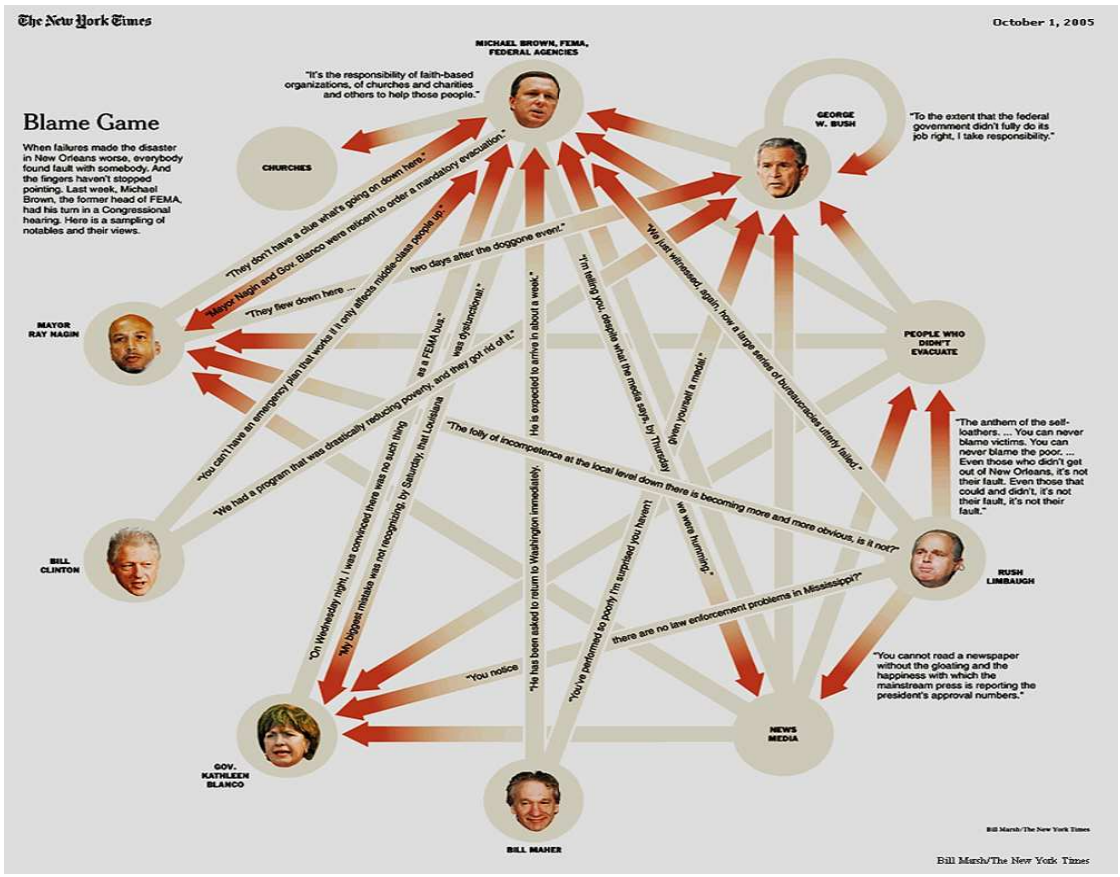

Figure 1.1: Blame Game (New York Times, 2005)

agencies: the first one conveys a rather neutral voice, while the second and third convey more pronounced (and opposing) opinions. Can we teach computers to understand such semantic differences among these titles?

Figure 1.1 shows an example of what we might be able to do using a computer system that can automatically recognize and analyze opinions in text. The graph shown in Figure 1.1, titled as "Blame Game", provides a summarized view of opinions among political figures regarding the natural disaster caused by Katrina. This particular graph was constructed manually by New York Times in 2005, however, our hope is to have computers read news articles and analyze

salient opinions for us.

In this dissertation, we explore computational methods that can push the envelope for automatic opinion analysis in text. There are two distinctive themes in our contributions: first, the focus of problems will be on *fine-grained* opinion analysis rather than coarse-grained analysis. Second, approaches explored in this dissertation are *structure-aware*. Each of these themes is elaborated in the next two subsections.

1.1.1 Fine-grained Opinion Analysis

There has been a great surge of research interest in the area of sentiment analysis and opinion mining (Pang and Lee, 2008). The majority of research in this area has been coarse-grained analysis, that is, the decision units for analysis are either at document-level or at paragraph-level. For instance, document-level sentiment classification (positive or negative) can be viewed as coarse-grained analysis (e.g. Pang et al. (2002), Blitzer et al. (2007)). Fine-grained opinion analysis on the other hand deals with text spans that are shorter than sentence boundaries, such as phrases or grammatical constituents (e.g. Wilson et al. (2005)). To see a concrete example, consider the following excerpt taken from a blog. [4]

> *[In a statement]*$_{\text{OP (=)}}$ *headed 'The Tyrant Visits Tirana' carried by the [Cuban news agency]*$_{\text{SRC}}$*, [Castro]*$_{\text{SRC + TAR}}$ *slammed [Bush]*$_{\text{SRC + TAR}}$ *for [voicing support]*$_{\text{OP (+)}}$ *for [Kosovo]*$_{\text{TAR}}$*'s independence [without the least respect]*$_{\text{OP (–)}}$ *for the interests of [Serbia]*$_{\text{TAR}}$ *and [Russia]*$_{\text{TAR}}$*...*

[4] http://balkanupdate.blogspot.com

Text spans that correspond to fine-grained opinion elements are annotated with brackets, where the types of opinion elements are marked by subscripts. *Opinion expressions* are labeled as 'OP', which are the text spans where the opinions are expressed either explicitly or implicitly. Each opinion expression is also annotated with the polarity of the sentiment: '(+)', '(-)', and '(=)' denote positive, negative, and neutral opinion respectively. The *sources of opinions* are labeled as 'SRC', which are people or organizations holding or expressing the opinions.[5] Finally, the *targets of opinions* are labeled as 'TAR', which are the targets or topics of opinions toward which opinions are expressed. Note that some of the targets of opinions are also sources of opinions.

A graphical representation of these fine-grained opinion elements and the relations among the opinion elements is given in Figure 1.2. Nodes in light colored shapes correspond to the sources of opinions, while nodes in oval shapes correspond to the targets of opinions. The directed edges represent opinion expressions, where the direction of the edge indicates the direction of the opinion, and the color of the edge encodes the polarity of the opinion.

The graph shown in Figure 1.2 depicts the key components of fine-grained opinion analysis, however, by no means it represents the complete set of problems that needs to be addressed for fine-grained opinion analysis. There are other components of fine-grained opinion analysis not specified in the graph, such as determining the intensity of opinion expressions (i.e., strong opinion v.s. weak opinion), determining the opinion elements that refer to the same person or organization (i.e., coreference resolution), and summarizing the salient opinions. In this dissertation, we investigate some of these key problems that constitute fine-grained opinion analysis.

[5]Some researchers use the term *opinion holders* instead of *sources of opinions*.

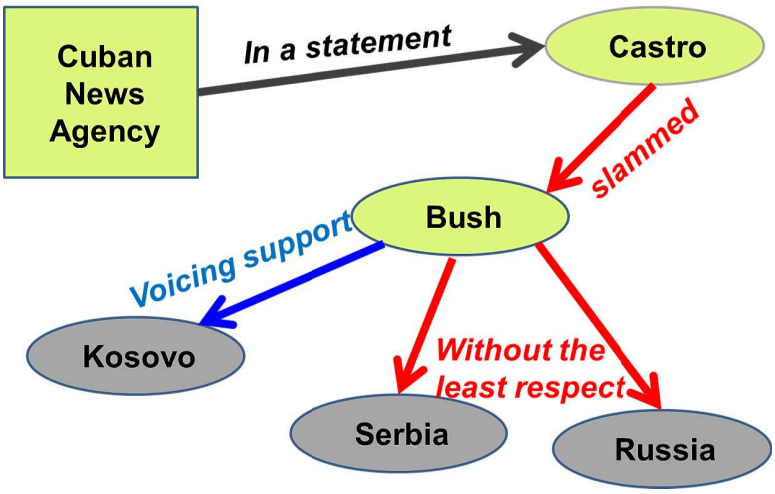

Figure 1.2: Structure Of Opinion Elements

1.1.2 Structure-aware Approaches

The approaches developed in our work are structure-aware in that we design the inference and/or learning algorithms reflecting the task-specific linguistic (syntactic and/or semantic) structure. A significant portion of research in sentiment analysis leverages existing techniques that have been well-studied for NLP tasks that do not aim for sentiment analysis. For instance, the task of document-level sentiment analysis is typically modeled as a more general text categorization task. However, as has been shown by a number of researchers, sentiment analysis creates new challenges that are not found in general text analysis (e.g. Pang et al. (2002), Wilson et al. (2005)). Therefore, recognizing task-specific linguistic structure, and incorporating such structure into either inference (e.g. Pang and Lee (2004), Choi et al. (2006)), or learning algorithms (e.g. Blitzer et al. (2007)) can significantly improve the performance. This theme is pursued throughout this dissertation as follows:

(1) For extracting opinion elements such as opinion expressions and sources of opinions, and finding links among the two types of opinion elements, we recognize various interactions among the opinion elements. For instance, each source of an opinion typically links to a single opinion expression in each sentence. We exploit such interactions by encoding them as soft and hard constraints for integer linear programming (Section 1.2.1 & Chapter 2).

(2) For extracting opinion expressions together with attributes such as polarity and intensity, we recognize a hierarchical structure among different tasks. For instance, extracting a *positive* opinion expression is a part of extracting any opinion expression, and extracting a *strongly positive* opinion expression is a part of extracting positive opinion expressions with any intensity. Also, extracting a *strongly positive* opinion expression (e.g. *"very cost-efficient"*) has a connection to extracting a *strongly negative* opinion expression (e.g. *"very inefficient"*) in that words such as "very" can be used in both positive and negative context to intensify the opinion. We employ a hierarchical sequential learning algorithm to reflect such hierarchical structure among different and yet related tasks (Section 1.2.2 & Chapter 3).

(3) For expression-level polarity classification, we observe the compositional nature in the way human infer the expression-level polarity based on word-level polarities and compositional rules that combine polarities in a bottom-up manner. We incorporate this observation into the learning procedure by taking the compositional inference rules as a structural inference subroutine (Section 1.2.3 & Chapter 4).

(4) For adapting a polarity lexicon for a domain specific NLP task, we recog-

nize word-to-word relations and word-to-expression relations concerning word-level polarity and expression-level polarity. For instance, if a word appears in positive opinion expressions most of the time, such a word is likely to carry a positive polarity. On the other hand, if a word associates with any polarity equally likely, then there is a good chance that such a word does not carry a distinctive polarity. We encode such observations as soft and hard constraints for integer linear programming (Section 1.2.4 & Chapter 5).

(5) For coreference resolution, we develop a learning algorithm that exploits the relationship between local decisions (i.e., coreferent decisions on each pair of mentions, where each decision is made independently from other decisions) and global decisions (i.e., coreferent decisions for all pairs of mentions, where decisions are made simultaneously) (Section 1.2.5 & Chapter 6).

1.2 Tasks

We now introduce five sets of problems that are tackled in this dissertation. Each of these will be further elaborated in Chapter 2 - Chapter 6.

1.2.1 Joint Extraction of Opinion Elements and Relations

Although keyword-based, or bag-of-words based approaches have been highly effective for document-level sentiment analysis, such approaches are unable to catch subtle differences in perspective like those illustrated in the examples dis-

cussed at the beginning of this section. Therefore, an important step toward human-like interpretation of opinion is recognizing fine-grained opinion elements, such as the *opinion expressions* and the *sources of opinions*, as described in Section 1.1.1. The extracted opinion elements can then be used as building blocks for various opinion applications, such as opinion summarization (e.g. Hu and Liu (2004a)) or opinion-oriented question answering (e.g Yu and Hatzivassiloglou (2003), Claire Cardie and Litman. (2004)).

Toward this goal, we have designed and built a system for extracting fine-grained opinion elements by casting the problem as an information extraction and relation learning task (Breck et al., 2007, Choi et al., 2006, Choi et al., 2005). The resulting system (Choi et al., 2006) has been employed by start-up Jodange (http://www.jodange.com)[6], as one of the core components of the company's opinion utility that extracts and aggregates opinions from social media.

1.2.2 Joint Extraction of Opinions and their Attributes

Previous research has recognized that determining polarity is closely related to the task of determining intensity (e.g. Pang and Lee (2005), Mao and Lebanon (2007), Zhao et al. (2008)), and learning approaches exploiting such relationship between polarity and intensity have shown to improve sentiment analysis. For instance, recognizing a *strongly positive* opinion expression (e.g. *"very cost-efficient"*) has a connection to recognizing a *strongly negative* opinion expression (e.g. *"very inefficient"*) in that words such as "very" can be used in both positive and negative context to intensify the opinion.

[6]It is co-founded by Claire Cardie.

We hypothesize that a similar benefit can be obtained for fine-grained opinion analysis as well. In particular, our goal is to extract opinion expressions from raw text, together with their attributes – polarity and intensity. We encode the relation between polarity and intensity as a hierarchy of classes (i.e., labels of tags for each token), then experiment with a hierarchical sequential learning technique (Choi and Cardie, 2010) based on Conditional Random Fields (CRF) (Lafferty et al., 2001).

1.2.3 Polarity Inference in Light of Compositional Semantics

Another fundamental operation toward fine-grained sentiment analysis is determining the polarity of sentiments conveyed in opinion expressions. Most previous NLP research in polarity classification relies on an inference step that does not resemble the compositional nature by which humans interpret natural language. For instance, given the news title "EPA Reports Decrease in Toxic Chemical Pollution", one would consider it as good news, not because there is any individual word that directly conjures positive affect, but because the word "decrease" effectively *negates* the negative affect of "Toxic Chemical Pollution" when the larger phrase "Decrease in Toxic Chemical Pollution" is interpreted (e.g. Moilanen and Stephen (2007)). We realize that such compositional nature in fine-grained polarity inference can be connected to *compositional semantics* (Montague, 1974), a classic body of research crossing linguistics and logic.

To bridge the gap between theories in compositional semantics and practical approaches based on machine learning techniques, we investigate computational methods that can combine the goodness of the two

(Choi and Cardie, 2008). More specifically, we devise simple *compositional* rules based on syntactic patterns for determining the polarity of sentiment expressions, and then incorporate the rules as structural inference for the learning algorithm.

1.2.4 Lexicon Adaptation as Constraint Optimization

Lexical resources, a polarity lexicon in particular, has shown to be a critical component for fine-grained sentiment analysis, as most of previous research dealing with fine-grained sentiment analysis starts by building or obtaining a polarity lexicon (e.g. Kim and Hovy (2004), Kennedy and Inkpen (2005), Wilson et al. (2005)). As a result, some researchers have focused more directly on the development of polarity lexicon (e.g. Takamura et al. (2005), Esuli and Sebastiani (2006)).

We look at the acquisition of polarity lexicon in a slightly different angle: in particular, we question how to adapt a general-purpose polarity lexicon for a specific application and a specific domain of data.

Such a question stems from the practical difficulty one can encounter when trying to make the best use of existing lexical resources for sentiment-analysis: although there has been plentiful research in the creation of lexical resources for sentiment analysis, most is conducted in isolation from actual applications. As a result, a purportedly better lexical resource might not lead to better performance when utilized for a specific natural language application. We conjecture that the basis of the problem is that the meaning of a word in isolation is often not the same as its meaning in context (Chomsky, 1965).

In this work, we investigate how to adapt a general-purpose polarity lexicon into a domain-specific one in the context of a specific NLP task. In order to exploit a number of statistical and linguistic clues between words and opinion expressions, we cast the adaptation problem as a constraint optimization problem using integer linear programming (Choi and Cardie, 2009).

1.2.5 Coreference Resolution

Once we have identified fine-grained opinion elements in text, we need to determine whether some of the extracted phrases are referring to an identical entity – namely, coreference resolution. For this task, we develop "structured local training", a machine learning technique based on Conditional Random Fields (CRFs) (Lafferty et al., 2001) that directly incorporates the interaction between local decisions (i.e., coreferent decisions on each pair of mentions, where each decision is made independently from other decisions) and global decisions (i.e., coreferent decisions for all pairs of mentions, where decisions are made simultaneously) into the learning procedure (Choi and Cardie, 2007).

The key insight is that the optimal gold standard for the local decisions might not necessarily coincide with the gold standard for the global decisions. This can happen because some of the local decisions cannot be made correctly based only on the local information. For instance, it is hard to judge whether a mention of "she" in one sentence refers to the same person as a mention of "she" in another sentence, without resolving other co-referent mentions that provide more information. Therefore, we model the (unknown) gold standard for the local decisions as hidden variables, instead of assuming that the gold standard

for the local decisions should be identical to the gold standard for the global decisions.

As a secondary contribution, we propose "biased potential functions" that can empirically drive CRFs towards performance improvements with respect to the preferred evaluation measure.

1.3 Related Work

There has been a great surge of research interest in automatic sentiment analysis and opinion mining in the past decade. In their highly insightful and thorough survey of the field, Pang and Lee (2008) trace the beginning of research activities on sentiment analysis based on statistical approaches to around 2001 (e.g., Das and Chen. (2001), Tateishi et al. (2001), Tong (2001), Morinaga et al. (2002), Turney (2002), Pang et al. (2002)).

It seems that it was not until 2003 when researchers started recognizing the need and the possibilities (e.g. Cardie et al. (2003), Cardie et al. (2004), Riloff and Wiebe (2003), Bethard et al. (2004), Stoyanov et al. (2005)) for more fine-grained sentiment analysis that deals with text spans that are shorter than sentence boundaries, such as phrases or grammatical constituents.

Perhaps one of the first work that presented a statistical approach to recognizing fine-grained opinion expressions is the bootstrap method by Riloff and Wiebe (2003). Some researchers (e.g. Wiebe et al. (2005), Wilson et al. (2005)) employed a simple dictionary look-up method to identify opinion expressions in order to assist other sentiment-based NLP tasks. Other researchers (e.g. Kim

and Hovy (2005a), Choi et al. (2006), Kim and Hovy (2006), Breck et al. (2007)) explored supervised learning techniques to learn opinion expressions from human annotated data.

Bethard et al. (2004) are the first to present a statistical approach to identify sources of opinions, however, their study was confined to *propositional* opinion whose content appears in the propositional argument of a verb that introduces an opinion. For instance, in the following example,

"I *believe* [you have to use the system to change it]."

the verb "believe" introduces an opinion whose content is in the propositional argument marked with brackets. Other researchers in the subsequent years (e.g. Choi et al. (2005), Kim and Hovy (2005b), Choi et al. (2006)) looked for sources of opinions for broader ranges of opinions.

Much of research in sentiment analysis and opinion mining has bloomed around product reviews (e.g Pang et al. (2002), Morinaga et al. (2002), Dave et al. (2003), Hu and Liu (2004a), Blitzer et al. (2007)), as product reviews are of great interest for both users and companies who seek for information from the web (Pang and Lee, 2008). Another reason that facilitated high research activities around this type of data that it is relatively easy to harvest a large amount of data with gold standard without incurring human annotation (e.g. Pang et al. (2002), Blitzer et al. (2007)).

There have been research around other types of data as well. Some researchers have focused on analyzing newswire articles for opinion analysis (e.g. Wiebe et al. (2005), Kim and Hovy (2005b), Stoyanov et al. (2005), Choi et al. (2006), Ku et al. (2006), Fukuhara et al. (2007), Somasundaran et al. (2007b),

Stepinski and Mittal (2007)).

In comparison to product reviews, newswire data poses unique challenges as follows: first, the general topics in product reviews are typically restricted to a certain type of product (e.g. camera). However, in newswire corpus, each document can be about any random event, which makes it harder to learn various lexical items and syntactic patterns that associates with opinion elements. Second, in product reviews, the sources of opinions are the writer of the reviews in almost all cases. In contrast, in newswire articles, there can be multiple sources of writers within a single document, or even within a single sentence, thereby requiring the need for identifying sources of opinions. Third, people tend to be more explicit in expressing opinions in product reviews (e.g "I loved this movie" or "I hate this camera") than it is in newswire articles.

Recently, there have been growing interest in analyzing user-created blogs for sentiment analysis as well (e.g. Adamic and Glance (2005), Chesley et al. (2006), Eguchi and Shah (2006), Zhou and Hovy (2006), Conrad and Schilder (2007), Liu et al. (2007), Bautin et al. (2008)). Similarly as newswire articles, the general topics of blogs are not as confined as product reviews.

Other interesting data for sentiment analysis include medical domain (e.g. Niu et al. (2005)), meeting transcripts (e.g. Somasundaran et al. (2007a)), and congressional floor-debate transcripts (e.g. Thomas et al. (2006)).

In this dissertation, we do not pursue automatic identification of targets or topics of opinions, as the availability of annotated data for targets of opinions has been rather limited until very recent for open-topic data such as newswire corpus. Stoyanov and Cardie (2008a) introduced the first annotation of targets

of opinions for newswire articles, and Stoyanov and Cardie (2008b) investigated algorithms for topic identification and topic coreference resolution. Defining and recognizing the targets of opinions in product review domain are generally considered to be more viable. Indeed, there have been a number of research that recognize the product *features* or *aspects* that correspond specific targets of opinions (e.g. Hu and Liu (2004b), Popescu and Etzioni (2005a), Snyder and Barzilay (2007), Titov and McDonald (2008)).

CHAPTER 2

JOINT EXTRACTION OF OPINION ELEMENTS AND RELATIONS

2.1 Introduction

Information extraction tasks such as recognizing entities and relations have long been considered critical to many domain-specific NLP tasks (e.g. Mooney and Bunescu (2005), Prager et al. (2000), White et al. (2001)). Researchers have further shown that *opinion-oriented information extraction* can provide analogous benefits to a variety of practical applications including product reputation tracking (e.g. Morinaga et al. (2002)), opinion-oriented question answering (Stoyanov et al., 2005), and opinion-oriented summarization (e.g. Cardie et al. (2004), Liu et al. (2005)). Moreover, much progress has been made in the area of opinion extraction: it is possible to identify sources of opinions (i.e. the opinion holders) (e.g. Choi et al. (2005) and Kim and Hovy (2005b)), to determine the polarity and strength of opinion expressions (e.g. Wilson et al. (2005)), and to recognize propositional opinions and their sources (e.g. Bethard et al. (2004)) with reasonable accuracy. To date, however, there has been no effort to simultaneously identify arbitrary opinion expressions, their sources, and the relations between them. Without progress on the *joint extraction of opinion entities **and** their relations*, the capabilities of opinion-based applications will remain limited.

Fortunately, research in machine learning has produced methods for global inference and joint classification that can help to address this deficiency (e.g. Bunescu et al. (2004), Roth and tau Yih (2004)). Moreover, it has been shown that exploiting dependencies among entities and/or relations via global inference not only solves the joint extraction task, but often boosts performance on

the individual tasks when compared to classifiers that handle the tasks independently — for semantic role labeling (e.g. Punyakanok et al. (2004)), information extraction (e.g. Roth and tau Yih (2004)), and sequence tagging (e.g. Sutton et al. (2007)).

In this work, we present a global inference approach (Roth and tau Yih, 2004) to the extraction of opinion-related entities and relations. In particular, we aim to identify two types of entities (i.e. spans of text): entities that express opinions and entities that denote sources of opinions. More specifically, we use the term *opinion expression* to denote all direct expressions of subjectivity including opinions, emotions, beliefs, sentiment, etc., as well as all speech expressions that introduce subjective propositions; and use the term *source* to denote the person or entity (e.g. a report) that holds the opinion.[1] In addition, we aim to identify the relations between opinion expression entities and source entities. That is, for a given opinion expression O_i and source entity S_j, we determine whether the relation $L_{i,j} \stackrel{\text{def}}{=} (S_j \text{ expresses } O_i)$ obtains, i.e. whether S_j is the source of opinion expression O_i. In what follows, we refer to this particular relation as the *link* relation. Consider, for example, the following sentences:

S1. [*Bush*](1) intends(1) to curb the increase in harmful gas emissions and is counting on(1) the good will(2) of [*US industrialists*](2).

S2. By questioning(3) [*the Imam*](4)'s edict(4) [*the Islamic Republic of Iran*](3) made [*the people of the world*](5) understand(5)...

The underlined phrases above are opinion expressions and phrases marked with square brackets are source entities. The numeric superscripts on entities indicate link relations: a source entity and an opinion expression with the same

[1] See Wiebe et al. (2005) for additional details.

number satisfy the link relation. For instance, the source entity *"Bush"* and the opinion expression *"intends"* satisfy the link relation, and so do *"Bush"* and *"counting on."* Notice that a sentence may contain more than one link relation, and link relations are not one-to-one mappings between sources and opinions. Also, the pair of entities in a link relation may not be the closest entities to each other, as is the case in the second sentence, between *"questioning"* and *"the Islamic Republic of Iran."*

We expect the extraction of opinion relations to be critical for many opinion-oriented NLP applications. For instance, consider the following question that might be given to a question-answering system:

- What is *the Imam*'s opinion toward *the Islamic Republic of Iran*?

Without in-depth opinion analysis, the question-answering system might mistake example S2 as relevant to the query, even though S2 exhibits the opinion of the Islamic Republic of Iran toward Imam, not the other way around.

2.1.1 Key Ideas

Inspired by Roth and tau Yih (2004), we model our task as global, constraint-based inference over separately trained entity and relation classifiers. In particular, we develop three base classifiers: two sequence-tagging classifiers for the extraction of opinion expressions and sources, and a binary classifier to identify the link relation. The global inference procedure is implemented via integer linear programming (ILP) to produce an optimal and coherent extraction of entities and relations.

Because many (60%) opinion-source relations appear as predicate-argument relations, where the predicate is a verb, we also hypothesize that semantic role labeling (SRL) will be very useful for our task. We present two baseline methods for the joint opinion-source recognition task that use a state-of-the-art SRL system (Koomen et al., 2005), and describe two additional methods for incorporating SRL into our ILP-based system.

2.1.2 Summary of Results

Our experiments show that the global inference approach not only improves relation extraction over the base classifier, but does the same for individual entity extractions. For source extraction in particular, our system achieves an F-measure of 78.1, significantly outperforming previous results in this area (Choi et al., 2005), which obtained an F-measure of 69.4 on the same corpus. In addition, we achieve an F-measure of 68.9 for link relation identification and 82.0 for opinion expression extraction; for the latter task, our system achieves human-level performance.[2]

2.2 The Big Picture

Our system developed in this work operates in the following three phases.

[2]Wiebe et al. (2005) reports human annotation agreement for opinion expression as 82.0 by F1 measure.

2.2.1 Extraction of Opinion and Source Entities

We begin by developing two separate token-level sequence-tagging classifiers for opinion expression extraction and source extraction, using linear-chain Conditional Random Fields (CRFs) (Lafferty et al., 2001). The sequence-tagging classifiers are trained using only local syntactic and lexical information to extract each type of entity without knowledge of any nearby or neighboring entities or relations. We collect *n*-best sequences from each sequence tagger in order to boost the recall of the final system.

2.2.2 Link Relation Classification

We also develop a relation classifier that is trained and tested on all pairs of opinion and source entities extracted from the aforementioned *n*-best opinion expression and source sequences. The relation classifier is modeled using Markov order-0 CRFs(Lafferty et al., 2001), which are equivalent to maximum entropy models. It is trained using only local syntactic information potentially useful for connecting a pair of entities, but has no knowledge of nearby or neighboring extracted entities and link relations.

2.2.3 Integer Linear Programming

Finally, we formulate an integer linear programming problem for each sentence using the results from the previous two phases. In particular, we specify a number of soft and hard constraints among relations and entities that take into account the confidence values provided by the supporting entity and relation

classifiers, and that encode a number of heuristics to ensure coherent output. Given these constraints, global inference via ILP finds the optimal, coherent set of opinion-source pairs by exploiting mutual dependencies among the entities and relations.

While good performance in entity or relation extraction can contribute to better performance of the final system, this is not always the case. (Punyakanok et al., 2004) notes that, in general, it is better to have high recall from the classifiers included in the ILP formulation. For this reason, it is not our goal to directly optimize the performance of our opinion and source entity extraction models or our relation classifier.

The rest of the paper is organized as follows. Related work is outlined below. Section 3 describes the components of the first phase of our system, the opinion and source extraction classifiers. Section 4 describes the construction of the link relation classifier for phase two. Section 5 describes the ILP formulation to perform global inference over the results from the previous two phases. Experimental results that compare our ILP approach to a number of baselines are presented in Section 6. Section 7 describes how SRL can be incorporated into our global inference system to further improve the performance. Final experimental results and discussion comprise Section 8.

2.3 Extraction of Opinion and Source Entities

We develop two separate sequence tagging classifiers for opinion extraction and source extraction, using linear-chain Conditional Random Fields (CRFs) (Lafferty et al., 2001). The sequence tagging is encoded as the typical 'BIO'

scheme.[3] Each training or test instance represents a sentence, encoded as a linear chain of tokens and their associated features. Our feature set is based on that of (Choi et al., 2005) for source extraction[4], but we include additional lexical and WordNet-based features. For simplicity, we use the same features for opinion entity extraction and source extraction, and let the CRFs learn appropriate feature weights for each task.

2.3.1 Features

For each token x_i, we include the following features. For details, see (Choi et al., 2005).

word: words in a [-4, +4] window centered on x_i.

part-of-speech: POS tags in a [-2, +2] window.[5]

grammatical role: grammatical role (subject, object, prepositional phrase types) of x_i derived from a dependency parse.[6]

dictionary: whether x_i is in the opinion expression dictionary culled from the training data and augmented by approximately 500 opinion words from the MPQA Final Report[7]. Also computed for tokens in a [-1, +1] window and for x_i's parent "chunk" in the dependency parse.

semantic class: x_i's semantic class.[8]

WordNet: the WordNet hypernym of x_i.[9]

[3] 'B' is for the token that **b**egins an entity, 'I' is for tokens that are **i**nside an entity, and 'O' is for tokens **o**utside an entity.

[4] We omit only the extraction pattern features.

[5] Using GATE: `http://gate.ac.uk/`

[6] Provided by Rebecca Hwa, based on the Collins parser: `ftp://ftp.cis.upenn.edu/pub/mcollins/PARSER.tar.gz`

[7] `https://rrc.mitre.org/pubs/mpqaFinalReport.pdf`

[8] Using SUNDANCE: (`http://www.cs.utah.edu/řiloff/ publications.html#sundance`)

[9] `http://wordnet.princeton.edu/`

2.4 Relation Classification

We also develop a maximum entropy binary classifier for opinion-source *link* relation classification. Given an opinion-source pair, O_i-S_j, the relation classifier decides whether the pair exhibits a valid link relation, $L_{i,j}$. The relation classifier focuses only on the syntactic structure and lexical properties between the two entities of a given pair, without knowing whether the proposed entities are correct. Opinion and source entities are taken from the *n*-best sequences of the entity extraction models; therefore, some are invariably incorrect.

From each sentence, we create training and test instances for all possible opinion-source pairings that do not overlap: we create an instance for $L_{i,j}$ only if the span of O_i and S_j do not overlap.

For training, we also filter out instances for which neither the proposed opinion nor source entity overlaps with a correct opinion or source entity per the gold standard. This training instance filtering helps to avoid confusion between examples like the following (where entities marked in bold are the gold standard entities, and entities in square brackets represent the *n*-best output sequences from the entity extraction classifiers):

(1) **[The president]**_s_1 walked away from [the meeting]_o_1, [**[revealing]**_o_2 **his disappointment]**_o_3 with the deal.

(2) [The monster]_s_2 walked away, [revealing]_o_4 a little box hidden underneath.

For these sentences, we construct training instances for $L_{1,1}$, $L_{1,2}$, and $L_{1,3}$, but not $L_{2,4}$, which in fact has very similar sentential structure as $L_{1,2}$, and hence could

confuse the learning algorithm.

2.4.1 Features

The training and test instances for each (potential) link $L_{i,j}$ (with opinion candidate entity O_i and source candidate entity S_j) include the following features.

opinion entity word: the words contained in O_i.

phrase type: the syntactic category of the constituent in which the entity is embedded, e.g. NP or VP. We encode separate features for O_i and S_j.

grammatical role: the grammatical role of the constituent in which the entity is embedded. Grammatical roles are derived from dependency parse trees, as done for the entity extraction classifiers. We encode separate features for O_i and S_j.

position: a boolean value indicating whether S_j precedes O_i.

distance: the distance between O_i and S_j in numbers of tokens. We use four coarse categories: adjacent, very near, near, far.

dependency path: the path through the dependency tree from the head of S_j to the head of O_i. For instance, 'subj↑verb' or 'subj↑verb↓obj'.

voice: whether the voice of O_i is passive or active.

syntactic frame: key intra-sentential relations between O_i and S_j. The syntactic frames that we use are:

- [E_1:role]_[distance]_[E_2:role], where distance ∈ {adjacent, very near, near, far}, and E_i:role is the grammatical role of E_i. Either E_1 is an opinion entity and E_2 is a source, or vice versa.
- [E_1:phrase]_[distance]_[E_2:phrase], where E_i:phrase is the phrasal type of en-

tity E_i.

- [E_1:phrase]_[E_2:headword], where E_2 must be the opinion entity, and E_1 must be the source entity (i.e. no lexicalized frames for sources). E_1 and E_2 can be contiguous.
- [E_1:role]_[E_2:headword], where E_2 must be the opinion entity, and E_1 must be the source entity.
- [E_1:phrase]_NP_[E_2:phrase] indicates the presence of specific syntactic patterns, e.g. 'VP_NP_VP' depending on the possible phrase types of opinion and source entities. The three phrases do not need to be contiguous.
- [E_1:phrase]_VP_[E_2:phrase] (See above.)
- [E_1:phrase]_[wh-word]_[E_2:phrase] (See above.)
- Src_[distance]_[x]_[distance]_Op, where $x \in$ {by, of, from, for, between, among, and, have, be, will, not,], ", ... }.

When a syntactic frame is matched to a sentence, the bracketed items should be instantiated with particular values corresponding to the sentence. Pattern elements without square brackets are constants. For instance, the syntactic frame '[E_1:phrase]_NP_[E_2:phrase]' may be instantiated as 'VP_NP_VP'. Some frames are lexicalized with respect to the head of an opinion entity to reflect the fact that different verbs expect source entities in different argument positions (e.g. SOURCE *blamed* TARGET vs. TARGET *angered* SOURCE).

2.5 Integer Linear Programming

As noted in the introduction, we model our task as global, constraint-based inference over the separately trained entity and relation classifiers, and im-

Table 2.1: Binary ILP formulation

Objective function f
$= \sum_i (w_{o_i} O_i) + \sum_i (\bar{w}_{o_i} \bar{O}_i)$
$+ \sum_j (w_{s_j} S_j) + \sum_j (\bar{w}_{s_j} \bar{S}_j)$
$+ \sum_{i,j} (w_{l_{i,j}} L_{i,j}) + \sum_i (\bar{w}_{l_{i,j}} \bar{L}_{i,j})$
$\forall\ i,\ \ O_i + \bar{O}_i = 1$
$\forall\ j,\ \ S_j + \bar{S}_j = 1$
$\forall\ i,\ j,\ \ L_{i,j} + \bar{L}_{i,j} = 1$
$\forall\ i,\ \ O_i = \sum_j L_{i,j}$
$\forall\ j,\ \ S_j + A_j = \sum_i L_{i,j}$
$\forall\ j,\ \ A_j - S_j \leq 0$
$\forall\ i,\ j,\ i < j,\ X_i + X_j = 1,\ X \in \{S, O\}$

plement the inference procedure as binary integer linear programming (ILP) ((Roth and tau Yih, 2004), (Punyakanok et al., 2004)). ILP consists of an objective function which is a dot product between a vector of variables and a vector of weights, and a set of equality and inequality constraints among variables. Given an objective function and a set of constraints, LP finds the optimal assignment of values to variables, i.e. one that minimizes the objective function. In binary ILP, the assignments to variables must be either 0 or 1. The variables and constraints defined for the opinion recognition task are summarized in Table 2.1 and explained below.

2.5.1 Entity Variables and Weights

For each opinion entity, we add two variables, O_i and $\bar{O}_i$, where $O_i = 1$ means to extract the opinion entity, and $\bar{O}_i = 1$ means to discard the opinion entity. To ensure coherent assignments, we add equality constraints $\forall i,\ O_i + \bar{O}_i = 1$. The weights w_{o_i} and $\bar{w}_{o_i}$ for O_i and $\bar{O}_i$ respectively, are computed as a negative conditional probability of the span of an entity to be extracted (or suppressed) given the labelings of the adjacent variables of the CRFs:

$$
\begin{aligned}
w_{o_i} &\stackrel{\text{def}}{=} -\mathrm{P}\left(x_k, x_{k+1}, ..., x_l | x_{k-1}, x_{l+1}\right) \\
&\qquad \text{where } x_k = \text{`B'} \\
&\qquad \& \ x_m = \text{`I' for } m \in [k+1, l] \\
\bar{w}_{o_i} &\stackrel{\text{def}}{=} -\mathrm{P}\left(x_k, x_{k+1}, ..., x_l | x_{k-1}, x_{l+1}\right) \\
&\qquad \text{where } x_m = \text{`O' for } m \in [k, l]
\end{aligned}
$$

where x_i is the value assigned to the random variable of the CRF corresponding to an entity O_i. Likewise, for each source entity, we add two variables S_j and $\bar{S}_j$ and a constraint $S_j + \bar{S}_j = 1$. The weights for source variables are computed in the same way as opinion entities.

2.5.2 Relation Variables and Weights

For each link relation, we add two variables $L_{i,j}$ and $\bar{L}_{i,j}$, and a constraint $L_{i,j} + \bar{L}_{i,j} = 1$. By the definition of a link, if $L_{i,j} = 1$, then it is implied that $O_i = 1$ and $S_j = 1$. That is, if a link is extracted, then the pair of entities for the link

must be also extracted. Constraints to ensure this coherency are explained in the following subsection. The weights for link variables are based on probabilities from the binary link classifier.

2.5.3 Constraints for Link Coherency

In our corpus, a source entity can be linked to more than one opinion entity, but an opinion entity is linked to only one source. Nonetheless, the majority of opinion-source pairs involve one-to-one mappings, which we encode as hard and soft constraints as follows:

For each opinion entity, we add an equality constraint $O_i = \sum_j L_{i,j}$ to enforce that only one link can emanate from an opinion entity. For each source entity, we add an equality constraint and an inequality constraint that together allow a source to link to at most two opinions: $S_j + A_j = \sum_i L_{i,j}$ and $A_j - S_j \leq 0$, where A_j is an auxiliary variable, such that its weight is some positive constant value that suppresses A_j from being assigned to 1. And A_j can be assigned to 1 only if S_j is already assigned to 1. It is possible to add more auxiliary variables to allow more than two opinions to link to a source, but for our experiments two seemed to be a reasonable limit.

2.5.4 Constraints for Entity Coherency

When we use n-best sequences where $n > 1$, proposed entities can overlap. Because this should not be the case in the final result, we add an equality constraint $X_i + X_j = 1,\ X \in \{S, O\}$ for all pairs of entities with overlapping spans.

2.5.5 Adjustments to Weights

To balance the precision and recall, and to take into account the performance of different base classifiers, we apply adjustments to weights as follows.

1) We define six coefficients c_x and $\bar{c}_x$, where $x \in \{O, S, L\}$ to modify a group of weights as follows:

$$\forall i, x,\ w_{x_i} := w_{x_i} * c_x$$

$$\forall i, x,\ \bar{w}_{x_i} := \bar{w}_{x_i} * \bar{c}_x$$

In general, increasing c_x will promote recall, while increasing $\bar{c}_x$ will promote precision. Also, setting $c_o > c_s$ will put higher confidence on the opinion extraction classifier than the source extraction classifier.

2) We also define one constant c_A to set the weights for auxiliary variable A_i. That is,

$$\forall i,\ w_{A_i} := c_A$$

3) Finally, we adjust the confidence of the link variable based on n-th-best sequences of the entity extraction classifiers as follows:

$$\forall i,\ w_{L_{i,j}} := w_{L_{i,j}} * d$$

where $d \stackrel{\text{def}}{=} 4/(3 + min(m, n))$, when O_i is from an m-th sequence and S_j is from a n-th sequence.[10]

[10] This will smoothly degrade the confidence of a link based on the entities from higher n-th sequences. Values of d decrease as $4/4,\ 4/5,\ 4/6,\ 4/7$....

2.6 Experiments – Effect of Integer Linear Programming

We evaluate our system using the NRRC Multi-Perspective Question Answering (MPQA) corpus that contains 535 newswire articles that are manually annotated for opinion-related information. In particular, our gold standard opinion entities correspond to *direct subjective expression* annotations and *subjective speech event* annotations (i.e. speech events that introduce opinions) in the MPQA corpus (Wiebe et al., 2005). Gold standard source entities and link relations can be extracted from the *agent* attribute associated with each opinion entity. We use 135 documents as a development set and report 10-fold cross validation results on the remaining 400 documents in all experiments below.

We evaluate entity and link extraction using both an *overlap* and *exact* matching scheme.[11] Because the exact start and endpoints of the manual annotations are somewhat arbitrary, the overlap scheme is more reasonable for our task (Wiebe et al., 2005). We report results according to both matching schemes, but focus our discussion on results obtained using overlap matching.[12]

We use the Mallet[13] implementation of CRFs. For brevity, we will refer to the opinion extraction classifier as CRF-OP, the source extraction classifier as CRF-SRC, and the link relation classifier as CRF-LINK. For ILP, we use Matlab, which produced the optimal assignment in a matter of few seconds for each sentence. The weight adjustment constants defined for ILP are based on the development data.[14]

[11] Given two links $L_{1,1} = (O_1, S_1)$ and $L_{2,2} = (O_2, S_2)$, exact matching requires the spans of O_1 and O_2, and the spans of S_1 and S_2, to match exactly, while overlap matching requires the spans to overlap.

[12] (Wiebe et al., 2005) also reports the human annotation agreement study via the overlap scheme.

[13] Available at http://mallet.cs.umass.edu

[14] $c_o = 2.5, \bar{c}_o = 1.0, c_s = 1.5, \bar{c}_s = 1.0, c_L = 2.5, \bar{c}_L = 2.5, c_A = 0.2$. Values are picked so as to boost recall while reasonably suppressing incorrect links.

Table 2.2: Relation extraction performance

	Overlap Match			Exact Match		
	r(%)	p(%)	f(%)	r(%)	p(%)	f(%)
NEAREST-1	51.6	71.4	59.9	26.2	36.9	30.7
NEAREST-2	60.7	45.8	52.2	29.7	19.0	23.1
NEAREST-10	66.3	20.9	31.7	28.2	00.0	00.0
SRL	59.7	36.3	45.2	32.6	19.3	24.2
SRL+CRF-OP	45.6	83.2	58.9	27.6	49.7	35.5
ILP-1	51.6	80.8	63.0	26.4	42.0	32.4
ILP-10	64.0	72.4	68.0	31.0	34.8	32.8

2.6.1 Baselines

The link-nearest baselines For baselines, we first consider a *link-nearest* heuristic: for each opinion entity extracted by CRF-OP, the link-nearest heuristic creates a link relation with the closest source entity extracted by CRF-SRC. Recall that CRF-SRC and CRF-OP extract entities from n-best sequences. We test the link-nearest heuristic with $n = \{1, 2, 10\}$ where larger n will boost recall at the cost of precision. Results for the link-nearest heuristic on the full source-expresses-opinion relation extraction task are shown in the first three rows of table 2.6. NEAREST-1 performs the best in overlap-match F-measure, reaching 59.9. NEAREST-10 has higher recall (66.3%), but the precision is really low (20.9%). Performance of the opinion and source entity classifiers will be discussed in Section 8.

SRL baselines Next, we consider two baselines that use a state-of-the-art SRL system (Koomen et al., 2005). In many link relations, the opinion expression entity is a verb phrase and the source entity is in an agent argument position. Hence our second baseline, SRL, extracts all verb(V)-agent(A0) frames from the output of the SRL system and provides an upper bound on recall (59.7%) for systems that use SRL in isolation for our task. A more sophisticated baseline, SRL+CRF-OP, extracts only those V-A0 frames whose verb overlaps with entities extracted by the opinion expression extractor, CRF-OP. As shown in table 2.6, filtering out V-A0 frames that are incompatible with the opinion extractor boosts precision to 83.2%, but the F-measure (58.9) is lower than that of NEAREST-1.

2.6.2 Results

The ILP-*n* system in table 2.6 denotes the results of the ILP approach applied to the *n*-best sequences. ILP-10 reaches an F-measure of 68.0, a significant improvement over the highest performing baseline[15], and also a substantial improvement over ILP-1. Note that the performance of NEAREST-10 was much worse than that of NEAREST-1, because the 10-best sequences include many incorrect entities whereas the corresponding ILP formulation can discard the bad entities by considering dependencies among entities and relations.[16]

[15]Statistically significant by paired-t test, where $p < 0.001$.

[16]A potential issue with overlap precision and recall is that the measures may drastically overestimate the system's performance as follows: a system predicting a single link relation whose source and opinion expression both overlap with every token of a document would achieve 100% overlap precision and recall. We can ensure this does not happen by measuring the average number of (source, opinion) pairs to which each correct or predicted pair is aligned (excluding pairs not aligned at all). In our data, this does not exceed 1.08, (except for baselines), so we can conclude these evaluation measures are behaving reasonably.

Table 2.3: Relation extraction with ILP and SRL

	Overlap Match			Exact Match		
	r(%)	p(%)	f(%)	r(%)	p(%)	f(%)
ILP-1	51.6	80.8	63.0	26.4	42.0	32.4
ILP-10	64.0	72.4	68.0	31.0	34.8	32.8
ILP+SRL-f-1	51.7	81.5	63.3	26.6	42.5	32.7
ILP+SRL-f-10	65.7	72.4	68.9	31.5	34.3	32.9
ILP+SRL-fc-10	64.0	73.5	68.4	28.4	31.3	29.8

2.7 Incorporating Semantic Role Labeling

We next explore two approaches for more directly incorporating SRL into our system.

Extra SRL Features for the Link classifier We incorporate SRL into the link classifier by adding extra features based on SRL. We add boolean features to check whether the span of an SRL argument and an entity matches exactly. In addition, we include **syntactic frame** features as follows:

- [E_1:srl-arg]_[E_2:srl-arg], where E_i:srl-arg indicates the SRL argument type of entity E_i.
- [E_1.srl-arg]_[E_1:headword]_[E_2:srl-arg], where E_1 must be an opinion entity, and E_2 must be a source entity.

Table 2.4: Entity extraction performance (by overlap-matching)

		Opinion			Source			Link		
		r(%)	p(%)	f(%)	r(%)	p(%)	f(%)	r(%)	p(%)	f(%)
Before ILP	CRF-* w/ 1 best	76.4	88.4	81.9	67.3	81.9	73.9	60.5	50.5	55.0
	merged 10 best	95.7	31.2	47.0	95.3	24.5	38.9	N/A		
After ILP	ILP-SRL-f-10	75.1	82.9	78.8	80.6	75.7	78.1	65.7	72.4	68.9
	ILP-SRL-f-10 $\cup$ CRF-* w/ 1 best	82.3	81.7	82.0	81.5	73.4	77.3	N/A		

Extra SRL Constraints for the ILP phase We also incorporate SRL into the ILP phase of our system by adding extra constraints based on SRL. In particular, we assign very high weights for links that match V-A0 frames generated by SRL, in order to force the extraction of V-A0 frames.

2.8 Experiments – Effect of Semantic Role Labeling

Results using SRL are shown in Table 2.3. In the table, ILP+SRL-f denotes the ILP approach using the link classifier with the extra SRL 'f'eatures, and ILP+SRL-fc denotes the ILP approach using both the extra SRL 'f'eatures and the SRL 'c'onstraints. For comparison, the ILP-1 and ILP-10 results from Table 2.6 are shown in rows 1 and 2.

The F-measure score of ILP+SRL-f-10 is 68.9, about a 1 point increase from that of ILP-10, which shows that extra SRL features for the link classifier further improve the performance over our previous best results.[17] ILP+SRL-fc-10

[17]Statistically significant by paired-t test, where $p < 0.001$.

also performs better than ILP-10 in F-measure, although it is slightly worse than ILP+SRL-f-10. This indicates that the link classifier with extra SRL features already makes good use of the V-A0 frames from the SRL system, so that forcing the extraction of such frames via extra ILP constraints only hurts performance by not allowing the extraction of non-V-A0 pairs in the neighborhood that could have been better choices.

Contribution of the ILP phase In order to highlight the contribution of the ILP phase for our task, we present 'before' and 'after' performance in Table 2.4. The first row shows the performance of the individual CRF-OP, CRF-SRC, and CRF-LINK classifiers before the ILP phase. Without the ILP phase, the 1-best sequence generates the best scores. However, we also present the performance with merged 10-best entity sequences[18] in order to demonstrate that using 10-best sequences without ILP will only hurt performance. The precision of the merged 10-best sequences system is very low, however the recall level is above 95% for both CRF-OP and CRF-SRC, giving an upper bound for recall for our approach. The third row presents results after the ILP phase is applied for the 10-best sequences, and we see that, in addition to the improved link extraction described in Section 7, the performance on source extraction is substantially improved, from F-measure of 73.9 to 78.1. Performance on opinion expression extraction decreases from F-measure of 81.9 to 78.8. This decrease is largely due to *implicit* links, which we will explain below. The fourth row takes the union of the entities from ILP-SRL-f-10 and the entities from the best sequences from CRF-OP and CRF-SRC. This process brings the F-measure of CRF-OP up to 82.0, with a different precision-recall break down from those of 1-best sequences

[18]If an entity E_i extracted by the ith-best sequence overlaps with an entity E_j extracted by the jth-best sequence, where $i < j$, then we discard E_j. If E_i and E_j do not overlap, then we extract both entities.

Table 2.5: Relation extraction with ILP weight adjustment. (All cases using ILP+SRL-f-10)

	Overlap Match			Exact Match		
	r(%)	p(%)	f(%)	r(%)	p(%)	f(%)
DEV.CONF	65.7	72.4	68.9	31.5	34.3	32.9
NO.CONF	63.7	76.2	69.4	30.9	36.7	33.5

without ILP phase. In particular, the recall on opinion expressions now reaches 82.3%, while maintaining a high precision of 81.7%.

Effects of ILP weight adjustment Finally, we show the effect of weight adjustment in the ILP formulation in Table 2.5. The DEV.CONF row shows relation extraction performance using a weight configuration based from the development data. In order to see the effect of weight adjustment, we ran an experiment, NO.CONF, using fixed default weights.[19] Not surprisingly, our weight adjustment tuned from the development set is not the optimal choice for cross-validation set. Nevertheless, the weight adjustment helps to balance the precision and recall, i.e. it improves recall at the cost of precision. The weight adjustment is more effective when the gap between precision and recall is large, as was the case with the development data.

Implicit links A good portion of errors stem from the *implicit* link relation, which our system did not model directly. An implicit link relation holds for an opinion entity without an associated source entity. In this case, the opinion entity is linked to an *implicit* source. Consider the following example.

[19] To be precise, $c_x = 1.0, \bar{c}_x = 1.0$ for $x \in \{O, S, L\}$, but $c_A = 0.2$ is the same as before.

- Anti-Soviet hysteria was firmly oppressed.

Notice that opinion expressions such as *"Anti-Soviet hysteria"* and *"firmly oppressed"* do not have associated source entities, because sources of these opinion expressions are not explicitly mentioned in the text. Because our system forces each opinion to be linked with an explicit source entity, opinion expressions that do not have explicit source entities will be dropped during the global inference phase of our system. Implicit links amount to 7% of the link relations in our corpus, so the upper bound for recall for our ILP system is 93%. In the future we will extend our system to handle implicit links as well. Note that we report results against a gold standard that includes implicit links. Excluding them from the gold standard, the performance of our final system ILP+SRL-f-10 is 72.6% in recall, 72.4% in precision, and 72.5 in F-measure.

2.9 Related Work

The definition of our source-expresses-opinion task is similar to that of (Bethard et al., 2004); however, our definition of opinion and source entities are much more extensive, going beyond single sentences and propositional opinion expressions. In particular, we evaluate our approach with respect to (1) a wide variety of opinion expressions, (2) explicit and implicit[20] sources, (3) multiple opinion-source link relations per sentence, and (4) link relations that span more than one sentence. In addition, the link relation model explicitly exploits mutual dependencies among entities and relations, while (Bethard et al., 2004) does not directly capture the potential influence among entities.

[20] *Implicit* sources are those that are not explicitly mentioned. See Section 2.8 for more details.

(Kim and Hovy, 2005b) and (Choi et al., 2005) focus only on the extraction of sources of opinions, without extracting opinion expressions. Specifically, (Kim and Hovy, 2005b) assume a priori existence of the opinion expressions and extract a single source for each, while (Choi et al., 2005) do not explicitly extract opinion expressions nor link an opinion expression to a source even though their model implicitly learns approximations of opinion expressions in order to identify opinion sources. Other previous research focuses only on the extraction of opinion expressions (e.g. (Kim and Hovy, 2005a), (Munson et al., 2005) and (Wilson et al., 2005)), omitting source identification altogether.

There have also been previous efforts to simultaneously extract entities and relations by exploiting their mutual dependencies. (Roth and Yih, 2002) formulated global inference using a Bayesian network, where they captured the influence between a relation and a pair of entities via the conditional probability of a relation, given a pair of entities. This approach however, could not exploit dependencies between relations. (Roth and tau Yih, 2004) later formulated global inference using integer linear programming, which is the approach that we apply here. In contrast to our work, (Roth and tau Yih, 2004) operated in the domain of factual information extraction rather than opinion extraction, and assumed that the exact boundaries of entities from the gold standard are known a priori, which may not be available in practice.

2.10 Summary of Chapter

In this work, we presented a global inference approach to jointly extract entities and relations in the context of opinion oriented information extraction. The final

system achieves performance levels that are potentially good enough for many practical NLP applications.

CHAPTER 3

JOINT EXTRACTION OF OPINIONS AND THEIR ATTRIBUTES

3.1 Introduction

Automatic opinion recognition involves a number of related tasks, such as identifying expressions of opinion (e.g. (Kim and Hovy, 2005a), (Breck et al., 2007), (Popescu and Etzioni, 2005b)), determining their polarity (e.g. (Kim and Hovy, 2004), (Popescu and Etzioni, 2005b), (Wilson et al., 2005)), and determining their strength, or intensity (e.g. (Popescu and Etzioni, 2005b), (Wilson et al., 2005)). Most previous work treats each subtask in isolation: opinion expression extraction (i.e. detecting the boundaries of opinion expressions) and opinion attribute classification (e.g. determining values for polarity and intensity) are tackled as separate steps in opinion recognition systems.[1] Even the seemingly related tasks of classifying an opinion expression according to its polarity and its intensity have been treated as two orthogonal problems. Unfortunately, errors from individual components will propagate in systems with cascaded component architectures, causing performance degradation in the end-to-end system (e.g. (Finkel et al., 2006)) — in our case, in the end-to-end opinion recognition system.

[1](Popescu and Etzioni, 2005b) places a weak interaction between the task of identifying opinion expressions and the task of classifying its polarity in that they first identify *potential* opinion expressions, then determine the polarity, and then filter out any *potential* opinion expressions with neutral polarity. However, the interaction between the two tasks is minimal. Moreover, other researchers (e.g. (Kim and Hovy, 2006), (Wilson et al., 2005)) believe it is important to retain opinion expressions with neutral polarity.

3.1.1 Key Ideas

In this work, we present a novel *hierarchical sequential learning* technique using Conditional Random Fields (CRFs) (Lafferty et al., 2001) that can jointly extract entities from unstructured text as well as determine their attributes. We apply the technique to opinion recognition, detecting the boundaries of opinion expressions and assigning values to two of their key attributes — polarity and intensity. Our approach is motivated by the growing body of research in multi-class classification, which has shown that improvements in performance can be gained by exploiting the hierarchical structure among the classes (e.g. (Cai and Hofmann, 2004), (GuoDong et al., 2006)). These improvements have also been extended (e.g. in (Fine et al., 1998), (Skounakis et al., 2003), and (Deschacht and Moens, 2006)) to some of the sequence tagging algorithms that have worked well for information extraction tasks.

As a result, we hypothesize that the opinion recognition task will reap the same performance benefits if we jointly identify the opinion expressions, their polarity, and their intensity via methods that capture the hierarchical structure among the classes associated with the three different, yet related, natural language learning tasks. Because the class hierarchy for opinion recognition does not form a tree structure (see Figure 3.1), conventional Hierarchical Hidden Markov Models (HHMMs) (Fine et al., 1998, Skounakis et al., 2003) are not suitable. Also, the iterative top-down approach by (Deschacht and Moens, 2006) does not seem very suitable for our task, as their approach is geared toward very deep and large tree structures.

Instead, our approach is based on the hierarchical parameter-sharing technique proposed by (Cai and Hofmann, 2004) for SVMs and a hierarchical docu-

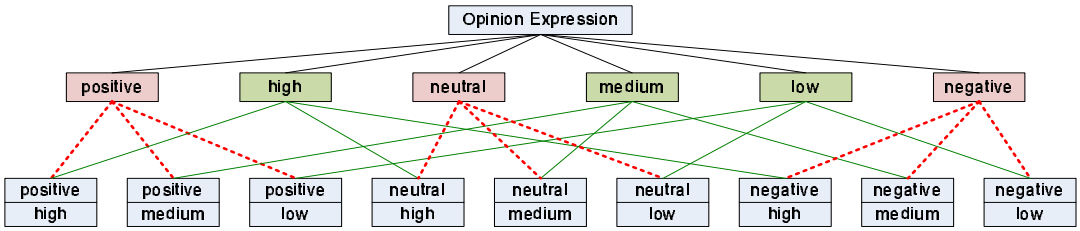

Figure 3.1: The hierarchical structure of classes for opinion expressions with polarity (positive, neutral, negative) and intensity (high, medium, low)

ment categorization application. We extend their approach here for sequential learning with CRFs and apply it to the problem of opinion recognition.

3.1.2 Summary of Results

Our proposed approach jointly extract opinion expressions from unstructured text and determine their attributes — polarity and intensity. Empirical results indicate that the simple joint sequential tagging approach even without exploiting the hierarchy brings a better performance than combining two separately developed systems. In addition, we found that the hierarchical joint sequential learning approach improves the performance over the simple joint sequential tagging method.

3.2 Hierarchical Sequential Learning

We define the problem of joint extraction of opinion expressions and their attributes as a sequence tagging task as follows. Given a sequence of tokens, $x = x_1 \ldots x_n$, we predict a sequence of labels, $y = y_1 \ldots y_n$, where $y_i \in \{0, ..., 9\}$

Table 3.1: Labels for Opinion Extraction with Polarity and Intensity: 'pos' stands for "positive", 'neu' stands for "neutral", and 'neg' stands for "negative".

LABEL	0	1	2	3	4	5	6	7	8	9
POLARITY	none	pos	pos	pos	neu	neu	neu	neg	neg	neg
INTENSITY	none	high	medium	low	high	medium	low	high	medium	low

are defined as conjunctive values of polarity labels and intensity labels, as shown in Table 3.1. Then the conditional probability $p(y|x)$ for linear-chain CRFs (Lafferty et al., 2001) is given as

$$P(y|x) = \frac{1}{Z_x} \exp \sum_i \Big(\lambda \, f(y_i, x, i) + \lambda' \, f'(y_{i-1}, y_i, x, i) \Big)$$

where Z_x is the normalization factor. In order to apply a hierarchical parameter sharing technique (e.g., Cai and Hofmann (2004), Zhao et al. (2008)), we extend parameters as follows.

$$\begin{aligned} \lambda \, f(y_i, x, i) &= \lambda_\alpha \, g_{\text{O}}(\alpha, x, i) \qquad (1) \\ &+ \lambda_\beta \, g_{\text{P}}(\beta, x, i) \\ &+ \lambda_\gamma \, g_{\text{S}}(\gamma, x, i) \end{aligned}$$

$$\begin{aligned} \lambda' \, f'(y_{i-1}, y_i, x, i) &= \lambda'_{\alpha,\hat{\alpha}} \, g'_{\text{O}}(\alpha, \hat{\alpha}, x, i) \\ &+ \lambda'_{\beta,\hat{\beta}} \, g'_{\text{P}}(\beta, \hat{\beta}, x, i) \\ &+ \lambda'_{\gamma,\hat{\gamma}} \, g'_{\text{S}}(\gamma, \hat{\gamma}, x, i) \end{aligned}$$

where g_{O} and g'_{O} are feature vectors defined for **O**pinion extraction, g_{P} and g'_{P} are feature vectors defined for **P**olarity extraction, and g_{S} and g'_{S} are feature vectors

defined for **S**trength extraction, and

$$\begin{aligned}
\alpha, \hat{\alpha} &\in \{\text{OPINION, NO-OPINION}\} \\
\beta, \hat{\beta} &\in \{\text{POSITIVE, NEGATIVE, NEUTRAL, NO-POLARITY}\} \\
\gamma, \hat{\gamma} &\in \{\text{HIGH, MEDIUM, LOW, NO-INTENSITY}\}
\end{aligned}$$

For instance, if $y_i = 1$, then

$$\begin{aligned}
\lambda\ f(1, x, i) &= \lambda_{\text{OPINION}}\ g_{\text{O}}(\text{OPINION}, x, i) \\
&+ \lambda_{\text{POSITIVE}}\ g_{\text{P}}(\text{POSITVE}, x, i) \\
&+ \lambda_{\text{HIGH}}\ g_{\text{S}}(\text{HIGH}, x, i)
\end{aligned}$$

If $y_{i-1} = 0, y_i = 4$, then

$$\begin{aligned}
&\lambda'\ f'(0, 4, x, i) \\
&= \lambda'_{\text{NO-OPINION,OPINION}}\ g'_{\text{O}}(\text{NO-OPINION, OPINION}, x, i) \\
&+ \lambda'_{\text{NO-POLARITY, NEUTRAL}}\ g'_{\text{P}}(\text{NO-POLARITY, NEUTRAL}, x, i) \\
&+ \lambda'_{\text{NO-INTENSITY, HIGH}}\ g'_{\text{S}}(\text{NO-INTENSITY, HIGH}, x, i)
\end{aligned}$$

This hierarchical construction of feature and weight vectors allows similar labels to share the same subcomponents of feature and weight vectors. For instance, all $\lambda\ f(y_i, x, i)$ such that $y_i \in \{1, 2, 3\}$ will share the same component $\lambda_{\text{POSITIVE}}\ g_{\text{P}}(\text{POSITVE}, x, i)$. Note that there can be other variations of hierarchical construction. For instance, one can add $\lambda_\delta\ g_{\text{I}}(\delta, x, i)$ and $\lambda'_{\delta,\hat{\delta}}\ g'_{\text{I}}(\delta, \hat{\delta}, x, i)$ to Equation (3.1) for $\delta \in \{0, 1, ..., 9\}$, in order to allow more individualized learning for each label.

Notice also that the number of sets of parameters constructed by Equation (3.1) is significantly smaller than the number of sets of parameters that are needed without the hierarchy. The former requires (2+4+4)+(2×2+4×4+4×4) =

46 sets of parameters, but the latter requires $(10) + (10 \times 10) = 110$ sets of parameters. Because a combination of a polarity component and an intensity component can distinguish each label, it is not necessary to define a separate set of parameters for each label.

3.3 Features

We first introduce definitions of key terms that will be used to describe features.

- **PRIOR-POLARITY & PRIOR-INTENSITY:**
  We obtain these prior-attributes from the *polarity lexicon* populated by Wilson et al. (2005).
- **EXP-POLARITY, EXP-INTENSITY & EXP-SPAN**: Words in a given opinion expression often do not share the same prior-attributes. Such discontinuous distribution of features can make it harder to learn the desired opinion expression boundaries. Therefore, we try to obtain expression-level attributes (EXP-POLARITY and EXP-INTENSITY) using simple heuristics. In order to derive EXP-POLARITY, we perform simple voting. If there is a word with a negation effect, such as "never", "not", "hardly", "against", then we flip the polarity. For EXP-INTENSITY, we use the highest PRIOR-INTENSITY in the span. The text span with the same expression-level attributes are referred to as EXP-SPAN.

3.3.1 Per-Token Features

Per-token features are defined in the form of $g_{\text{O}}(\alpha, x, i), g_{\text{P}}(\beta, x, i)$ and $g_{\text{S}}(\gamma, x, i)$. The domains of α, β, γ are as given in Section 3.

Common Per-Token Features

Following features are common for all class labels. The notation $\otimes$ indicates conjunctive operation of two values.

- PART-OF-SPEECH(x_i):

  based on GATE (Cunningham et al., 2002).

- WORD(x_i), WORD(x_{i-1}), WORD(x_{i+1})

- WORDNET-HYPERNYM(x_i):

  based on WordNet (Miller, 1995).

- OPINION-LEXICON(x_i):

  based on *opinion lexicon* (Wiebe et al., 2002).

- SHALLOW-PARSER(x_i):

  based on CASS partial parser (Abney, 1996).

- PRIOR-POLARITY(x_i) $\otimes$ PRIOR-INTENSITY(x_i)

- EXP-POLARITY(x_i) $\otimes$ EXP-INTENSITY(x_i)

- EXP-POLARITY(x_i) $\otimes$ EXP-INTENSITY(x_i) $\otimes$

  STEM(x_i)

- EXP-SPAN(x_i):

  boolean to indicate whether x_i is in an EXP-SPAN.

- DISTANCE-TO-EXP-SPAN(x_i): $0, 1, 2, 3+$.
- EXP-POLARITY(x_i) $\otimes$ EXP-INTENSITY(x_i) $\otimes$ EXP-SPAN(x_i)

Polarity Per-Token Features

These features are included only for $g_{\text{O}}(\alpha, x, i)$ and $g_{\text{P}}(\beta, x, i)$, which are the feature functions corresponding to the polarity-based classes.

- PRIOR-POLARITY(x_i), EXP-POLARITY((x_i)
- STEM(x_i) $\otimes$ EXP-POLARITY(x_i)
- COUNT-OF-*Polarity*:

  where *Polarity* $\in$ {positive, neutral, negative}. This feature encodes the number of positive, neutral, and negative EXP-POLARITY words respectively, in the current sentence.
- STEM(x_i) $\otimes$ COUNT-OF-*Polarity*
- EXP-POLARITY(x_i) $\otimes$ COUNT-OF-*Polarity*
- EXP-SPAN(x_i) and EXP-POLARITY(x_i)
- DISTANCE-TO-EXP-SPAN(x_i) $\otimes$ EXP-POLARITY(x_p)

Intensity Per-Token Features

These features are included only for $g_{\text{O}}(\alpha, x, i)$ and $g_{\text{S}}(\gamma, x, i)$, which are the feature functions corresponding to the intensity-based classes.

- PRIOR-INTENSITY(x_i), EXP-INTENSITY(x_i)

- STEM(x_i) ⊗ EXP-INTENSITY(x_i)
- COUNT-OF-STRONG, COUNT-OF-WEAK:

  the number of strong and weak EXP-INTENSITY words in the current sentence.
- INTENSIFIER(x_i): whether x_i is an intensifier, such as "extremely", "highly", "really".
- STRONGMODAL(x_i): whether x_i is a strong modal verb, such as "must", "can", "will".
- WEAKMODAL(x_i): whether x_i is a weak modal verb, such as "may", "could", "would".
- DIMINISHER(x_i): whether x_i is a diminisher, such as "little", "somewhat", "less".
- PRECEDED-BY-τ(x_i),

  PRECEDED-BY-τ(x_i) ⊗ EXP-INTENSITY(x_i):

  where $\tau \in$ { INTENSIFIER, STRONGMODAL, WEAKMODAL, DIMINISHER}
- $\tau(x_i)$ ⊗ EXP-INTENSITY(x_i),

  $\tau(x_i)$ ⊗ EXP-INTENSITY(x_{i-1}),

  $\tau(x_{i-1})$ ⊗ EXP-INTENSITY(x_{i+1})
- EXP-SPAN(x_i) ⊗ EXP-INTENSITY(x_i)
- DISTANCE-TO-EXP-SPAN(x_i) ⊗ EXP-INTENSITY(x_p)

3.3.2 Transition Features

Transition features are employed to help with boundary extraction as follows:

Table 3.2: Performance of Opinion Extraction with Correct Polarity Attribute

	Positive			Neutral			Negative		
Method Description	r(%)	p(%)	f(%)	r(%)	p(%)	f(%)	r(%)	p(%)	f(%)
Polarity-Only ∩ Intensity-Only (BASELINE1)	29.6	65.7	40.8	26.5	69.1	38.3	35.5	77.0	48.6
Joint without Hierarchy (BASELINE2)	30.7	65.7	41.9	29.9	66.5	41.2	37.3	77.1	50.3
Joint with Hierarchy	31.8	67.1	**43.1**	31.9	66.6	**43.1**	40.4	76.2	**52.8**

Polarity Transition Features

Polarity transition features are features that are used only for $g'_{\mathrm{O}}(\alpha, \hat{\alpha}, x, i)$ and $g'_{\mathrm{P}}(\beta, \hat{\beta}, x, i)$.

- PART-OF-SPEECH$(x_i) \otimes$ PART-OF-SPEECH$(x_{i+1}) \otimes$ EXP-POLARITY(x_i)
- EXP-POLARITY$(x_i) \otimes$ EXP-POLARITY(x_{i+1})

Intensity Transition Features

Intensity transition features are features that are used only for $g'_{\mathrm{O}}(\alpha, \hat{\alpha}, x, i)$ and $g'_{\mathrm{S}}(\gamma, \hat{\gamma}, x, i)$.

- PART-OF-SPEECH$(x_i) \otimes$ PART-OF-SPEECH$(x_{i+1}) \otimes$ EXP-INTENSITY(x_i)
- EXP-INTENSITY$(x_i) \otimes$ EXP-INTENSITY(x_{i+1})

Table 3.3: Performance of Opinion Extraction with Correct Intensity Attribute

Method Description	High			Medium			Low		
	r(%)	p(%)	f(%)	r(%)	p(%)	f(%)	r(%)	p(%)	f(%)
Polarity-Only ∩ Intensity-Only (BASELINE1)	26.4	58.3	36.3	29.7	59.0	39.6	15.4	60.3	24.5
Joint without Hierarchy (BASELINE2)	29.7	54.2	**38.4**	28.0	57.4	37.6	18.8	55.0	28.0
Joint with Hierarchy	27.1	55.2	36.3	32.0	56.5	**40.9**	21.1	56.3	**30.7**

3.4 Experiment

3.4.1 Configuration

We evaluate our system using the Multi-Perspective Question Answering (MPQA) corpus[2]. Our gold standard opinion expressions correspond to *direct subjective expression* and *expressive subjective element* (Wiebe et al., 2005).[3]

Our implementation of hierarchical sequential learning is based on the Mallet (McCallum, 2002) code for CRFs. In all experiments, we use a Gaussian prior of 1.0 for regularization. We use 135 documents for development, and test on a different set of 400 documents using 10-fold cross-validation. We investigate three options for jointly extracting opinion expressions with their attributes as follows:

[2]The MPQA corpus can be obtained at http://nrrc.mitre.org/NRRC/publications.htm.

[3]Only 1.5% of the polarity annotations correspond to *both*; hence, we merge *both* into the *neutral*. Similarly, for gold standard intensity, we merge *extremely high* into *high*.

Table 3.4: Performance of Opinion Extraction

Method Description	r(%)	p(%)	f(%)
Polar-Only ∩ Intensity-Only	43.3	92.0	58.9
Joint without Hierarchy	46.0	88.4	60.5
Joint with Hierarchy	48.0	87.8	**62.0**

3.4.2 Baselines

[Baseline-1] Polarity-Only ∩ Intensity-Only:
For this baseline, we train two separate sequence tagging CRFs: one that extracts opinion expressions only with the polarity attribute (using common features and polarity extraction features in Section 3), and another that extracts opinion expressions only with the intensity attribute (using common features and intensity extraction features in Section 3). We then combine the results from two separate CRFs by collecting all opinion entities extracted by both sequence taggers.[4] This baseline effectively represents a cascaded component approach.

[Baseline-2] Joint without Hierarchy: Here we use simple linear-chain CRFs without exploiting the class hierarchy for the opinion recognition task. We use the tags shown in Table 3.1.

Joint with Hierarchy: Finally, we test the hierarchical sequential learning approach elaborated in Section 3.

[4]We collect all entities whose portions of text spans are extracted by both models.

3.4.3 Results

We evaluate all experiments at the opinion entity level, i.e. at the level of each opinion expression rather than at the token level. We use three evaluation metrics: recall, precision, and F-measure with equally weighted recall and precision.

Table 3.4 shows the performance of opinion extraction without matching any attribute. That is, an extracted opinion entity is counted as correct if it overlaps[5] with a gold standard opinion expression, without checking the correctness of its attributes. Table 3.2 and 3.3 show the performance of opinion extraction with the correct polarity and intensity respectively.

From all of these evaluation criteria, JOINT WITH HIERARCHY performs the best, and the least effective one is BASELINE-1, which cascades two separately trained models. It is interesting that the simple sequential tagging approach even without exploiting the hierarchy (BASELINE-2) performs better than the cascaded approach (BASELINE-1).

When evaluating with respect to the polarity attribute, the performance of the negative class is substantially higher than the that of other classes. This is not surprising as there is approximately twice as much data for the negative class. When evaluating with respect to the intensity attribute, the performance of the LOW class is substantially lower than that of other classes. This result reflects the fact that it is inherently harder to distinguish an opinion expression with low intensity from no opinion. In general, we observe that determining

[5]Overlap matching is a reasonable choice as the annotator agreement study is also based on overlap matching (Wiebe et al., 2005). One might wonder whether the overlap matching scheme could allow a degenerative case where extracting the entire test dataset as one giant opinion expression would yield 100% recall and precision. Because each sentence corresponds to a different test instance in our model, and because some sentences do not contain any opinion expression in the dataset, such degenerative case is not possible in our experiments.

correct intensity attributes is a much harder task than determining correct polarity attributes.

In order to have a sense of upper bound, we also report the individual performance of two separately trained models used for BASELINE-1: for the Polarity-Only model that extracts opinion boundaries only with polarity attribute, the F-scores with respect to the positive, neutral, negative classes are 46.7, 47.5, 57.0, respectively. For the Intensity-Only model, the F-scores with respect to the high, medium, low classes are 37.1, 40.8, 26.6, respectively. Remind that neither of these models alone fully solve the joint task of extracting boundaries as well as determining two attributions simultaneously. As a result, when conjoining the results from the two models (BASELINE-1), the final performance drops substantially.

We conclude from our experiments that the simple joint sequential tagging approach even without exploiting the hierarchy brings a better performance than combining two separately developed systems. In addition, our hierarchical joint sequential learning approach brings a further performance gain over the simple joint sequential tagging method.

3.5 Related Work

Although there have been much research for fine-grained opinion analysis (e.g., Hu and Liu (2004a), Wilson et al. (2005), Wilson et al. (2006), Choi and Cardie (2008), Wilson et al. (2009)),[6] none is directly comparable to our results; much of

[6]For instance, the results of Wilson et al. (2005) is not comparable even for our Polarity-Only model used inside BASELINE-1, because Wilson et al. (2005) does not operate on the entire corpus as unstructured input. Instead, Wilson et al. (2005) evaluate only on known words that are in their opinion lexicon.

previous work studies only a subset of what we tackle in this work. However, as shown in Section 4.1, when we train the learning models only for a subset of the tasks, we can achieve a better performance instantly by making the problem simpler. Our work differs from most of previous work in that we investigate how solving multiple related tasks affects performance on sub-tasks.

The hierarchical parameter sharing technique used in this work has been previously used by Zhao et al. (2008) for opinion analysis. However, Zhao et al. (2008) employs this technique only to classify sentence-level attributes (polarity and intensity), without involving a much harder task of detecting boundaries of sub-sentential entities.

3.6 Summary of Chapter

We applied a hierarchical parameter sharing technique using Conditional Random Fields for fine-grained opinion analysis. Our proposed approach jointly extract opinion expressions from unstructured text and determine their attributes — polarity and intensity. Empirical results indicate that the simple joint sequential tagging approach even without exploiting the hierarchy brings a better performance than combining two separately developed systems. In addition, we found that the hierarchical joint sequential learning approach improves the performance over the simple joint sequential tagging method.

Furthermore, Wilson et al. (2005) simplifies the problem by combining neutral opinions and no opinions into the same class, while our system distinguishes the two.

CHAPTER 4

POLARITY INFERENCE IN LIGHT OF COMPOSITIONAL SEMANTICS

4.1 Introduction

Determining the polarity of sentiment-bearing expressions at or below the sentence level requires more than a simple bag-of-words approach. One of the difficulties is that words or constituents within the expression can interact with each other to yield a particular overall polarity. To facilitate our discussion, consider the following examples:

1: [I did [*not*]$^{\neg}$ have any [*doubt*]$^{-}$ about it.]$^{+}$

2: [The report [*eliminated*]$^{\neg}$ my [*doubt*]$^{-}$.]$^{+}$

3: [They could [*not*]$^{\neg}$ [*eliminate*]$^{\neg}$ my [*doubt*]$^{-}$.]$^{-}$

In the first example, "doubt" in isolation carries a negative sentiment, but the overall polarity of the sentence is positive because there is a *negator* "not", which flips the polarity. In the second example, both "eliminated" and "doubt" carry negative sentiment in isolation, but the overall polarity of the sentence is positive because "eliminated" acts as a negator for its argument "doubt". In the last example, there are effectively two negators – "not" and "eliminated" – which reverse the polarity of "doubt" twice, resulting in the negative polarity for the overall sentence.

These examples demonstrate that words or constituents interact with each other to yield the expression-level polarity. And a system that simply takes

the majority vote of the polarity of individual words will not work well on the above examples. Indeed, much of the previous learning-based research on this topic tries to incorporate salient interactions by encoding them as features. One approach includes features based on *contextual valence shifters*[1] (Polanyi and Zaenen, 2004), which are words that affect the polarity or intensity of sentiment over neighboring text spans (e.g., Kennedy and Inkpen (2005), Wilson et al. (2005), Shaikh et al. (2007)). Another approach encodes frequent subsentential patterns (e.g., Mcdonald et al. (2007)) as features; these might indirectly capture some of the subsentential interactions that affect polarity. However, both types of approach are based on learning models with a flat bag-of-features: some structural information can be encoded as higher order features, but the final representation of the input is still a flat feature vector that is inherently too limited to adequately reflect the complex structural nature of the underlying subsentential interactions (Liang et al., 2008).

Moilanen and Stephen (2007), on the other hand, handle the structural nature of the interactions more directly using the ideas from *compositional semantics* (e.g., Montague (1974), Dowty et al. (1981)). In short, *the Principle of Compositionality* states that the meaning of a compound expression is a function of the meaning of its parts and of the syntactic rules by which they are combined (e.g., Montague (1974), Dowty et al. (1981)). And Moilanen and Stephen (2007) develop a collection of composition rules to assign a sentiment value to individual expressions, clauses, or sentences. Their approach can be viewed as a type of structural inference, but their hand-written rules have not been empirically compared to learning-based alternatives, which one might expect to be more effective in handling some aspects of the polarity classification task.

[1]For instance, “never”, “nowhere”, “little”, “most”, “lack”, “scarcely”, “deeply”.

4.1.1 Key Ideas

In this work, we begin to close the gap between learning-based approaches to expression-level polarity classification and those founded on compositional semantics: we present a novel learning-based approach that incorporates structural inference motivated by compositional semantics into the learning procedure.

Adopting the view point of compositional semantics, our working assumption is that the polarity of a sentiment-bearing expression can be determined in a two-step process: (1) assess the polarities of the constituents of the expression, and then (2) apply a relatively simple set of inference rules to combine them recursively. Rather than a rigid application of hand-written compositional inference rules, however, we hypothesize that an ideal solution to the expression-level polarity classification task will be a method that can exploit ideas from compositional semantics while providing the flexibility needed to handle the complexities of real-world natural language — exceptions, unknown words, missing semantic features, and inaccurate or missing rules. The learning-based approach proposed in this paper takes a first step in this direction.

In addition to the novel learning approach, we present new insights for *content-word negators*, which we define as content words that can negate the polarity of neighboring words or constituents. (e.g., words such as "eliminated" in the example sentences). Unlike *function-word negators*, such as "not" or "never", content-word negators have been recognized and utilized less actively in previous work. (Notable exceptions include e.g., Niu et al. (2005), Wilson et al. (2005), and Moilanen and Stephen (2007).)

Table 4.1: Heuristic methods. (*n* refers to the number of negators found in a given expression.)

	VOTE	NEG(1)	NEG(N)	NEGEX(1)	NEGEX(N)	COMPO
type of negators	none	function-word		function-word & content-word		
# of negations applied	0	1	*n*	1	*n*	*n*
scope of negators	N/A	over the entire expression				compositional

4.1.2 Summary of Results

In our experiments, we compare learning- and non-learning-based approaches to expression-level polarity classification — with and without compositional semantics — and find that (1) simple heuristics based on compositional semantics outperform (89.7% in accuracy) other reasonable heuristics that do not incorporate compositional semantics (87.7%); they can also perform better than simple learning-based methods that do not incorporate compositional semantics (89.1%), (2) combining learning with the heuristic rules based on compositional semantics further improves the performance (90.7%), (3) content-word negators play an important role in determining the expression-level polarity, and, somewhat surprisingly, we find that (4) expression-level classification accuracy uniformly decreases as additional, potentially disambiguating, context is considered.

Table 4.2: Compositional inference rules motivated by compositional semantics.

	Rules		Examples
1	Polarity(not_[arg1])=	¬ Polarity(arg1)	not [bad]$_{arg1}$.
2	Polarity([VP]_[NP])=	Compose([VP], [NP])	[destroyed]$_{VP}$ [the terrorism]$_{NP}$.
3	Polarity([VP1]_to_[VP2])=	Compose([VP1], [VP2])	[refused]$_{VP1}$ to [deceive]$_{VP2}$ the man.
4	Polarity([adj]_to_[VP])=	Compose([adj], [VP])	[unlikely]$_{adj}$ to [destroy]$_{VP}$ the planet.
5	Polarity([NP1]_[IN]_[NP2])=	Compose([NP1], [NP2])	[lack]$_{NP1}$ [of]$_{IN}$ [crime]$_{NP2}$ in rural areas.
6	Polarity([NP]_[VP])=	Compose([VP], [NP])	[pollution]$_{NP}$ [has decreased]$_{VP}$.
7	Polarity([NP]_be_[adj])=	Compose([adj], [NP])	[harm]$_{NP}$ is [minimal]$_{adj}$.

Definition of Compose(arg1, arg2)

	Compose(arg1, arg2) =
For COMPOMC:	if (arg1 is a negator) then ¬ Polarity(arg2)
(**COMPO**sition w/ **M**ajority **C**lass)	else if (Polarity(arg1) == Polarity(arg2)) then Polarity(arg1)
	else the majority polarity of data
	Compose(arg1, arg2) =
For COMPOPR:	if (arg1 is a negator) then ¬ Polarity(arg2)
(**COMPO**sition w/ **PR**iority)	else Polarity(arg1)

4.2 Heuristic-Based Methods

We start by describing a set of heuristic-based methods for determining the polarity of a sentiment-bearing expression. Each assesses the polarity of the words or constituents using a polarity lexicon that indicates whether a word has positive or negative polarity, and finds negators in the given expression using a negator lexicon. The methods then infer the expression-level polarity using

voting-based heuristics (§ 2.1) or heuristics that incorporate compositional semantics (§2.2). The lexicons are described in §2.3.

4.2.1 Voting

We first explore five simple heuristics based on voting. VOTE is defined as the majority polarity vote by words in a given expression. That is, we count the number of positive polarity words and negative polarity words in a given expression, and assign the majority polarity to the expression. In the case of a tie, we default to the prevailing polarity of the data.

For NEG(1), we first determine the majority polarity vote as above, and then if the expression contains *any* function-word negator, flip the polarity of the majority vote once. NEG(N) is similar to NEG(1), except we flip the polarity of the majority vote *n* times after the majority vote, where *n* is the number of function-word negators in a given expression.

NEGEX(1) and NEGEX(N) are defined similarly as NEG(1) and NEG(N) above, except both function-word negators and content-word negators are considered as negators when flipping the polarity of the majority vote. See Table 4.1 for summary. Note that a word can be both a negator and have a negative prior polarity. For the purpose of voting, if a word is defined as a negator per the voting scheme, then that word does not participate in the majority vote.

For brevity, we refer to NEG(1) and NEG(N) collectively as NEG, and NEGEX(1) and NEGEX(N) collectively as NEGEX.

4.2.2 Compositional Semantics

Whereas the heuristics above use voting-based inference, those below employ a set of hand-written rules motivated by compositional semantics. Table 4.2 shows the definition of the rules along with motivating examples. In order to apply a rule, we first detect a syntactic pattern (e.g., [destroyed]$_{VP}$ [the terrorism]$_{NP}$), then apply the *Compose* function as defined in Table 4.2 (e.g., Compose([destroyed], [the terrorism]) by rule #2).

Compose first checks whether the first argument is a negator, and if so, flips the polarity of the second argument. Otherwise, *Compose* resolves the polarities of its two arguments. Note that if the second argument is a negator, we do not flip the polarity of the first argument, because the first argument in general is not in the semantic scope of the negation.[2] Instead, we treat the second argument as a constituent with negative polarity.

We experiment with two variations of the *Compose* function depending on how conflicting polarities are resolved: COMPOMC uses a *Compose* function that defaults to the **M**ajority **C**lass of the polarity of the data,[3] while COMPOPR uses a *Compose* function that selects the polarity of the argument that has higher semantic **PR**iority. For brevity, we refer to COMPOPR and COMPOMC collectively as COMPO.

Our implementation uses part-of-speech tags and function-words to coarsely determine the patterns. An alternative implementation could be based on parse trees. A parse tree based approach might be able to apply the rules

[2]Moilanen and Stephen (2007) provide more detailed discussion on the semantic scope of negations and the semantic priorities in resolving polarities.

[3]The majority polarity of the data we use for our experiments is negative.

more accurately for some cases, however, there might be other cases where the parser fails to parse the structure correctly, causing the application of the rules to fail as well. Because our goal in this work is not to hand-code the best heuristic rules, we chose the simpler method based on part-of-speech tags. In our simple implementation, we check the applicability of each rule in the order of #1 - #7. This ordering is rather arbitrary. If conflicting polarities still remain among constituents after applying a sequence of rules, and there is no more applicable rule, then we default to the majority polarity class of data.

4.2.3 Lexicons

The polarity lexicon is initialized with the lexicon of Wilson et al. (Wilson et al., 2005) and then expanded using the General Inquirer dictionary.[4] In particular, a word contained in at least two of the following categories is considered as positive: POSITIV, PSTV, POSAFF, PLEASUR, VIRTUE, INCREAS, and a word contained in at least one of the following categories is considered as negative: NEGATIV, NGTV, NEGAFF, PAIN, VICE, HOSTILE, FAIL, ENLLOSS, WLBLOSS, TRANLOSS.

For the (function- and content-word) negator lexicon, we collect a handful of seed words as well as General Inquirer words that appear in either NOTLW or DECREAS category. Then we expand the list of content-negators using the synonym information of WordNet (Miller, 1995) to take a simple vote among senses.

[4] Available at http://www.wjh.harvard.edu/~inquirer/. When consulting the General Inquirer dictionary, senses with less than 5% frequency and senses specific to an idiom are dropped.

Simple Classification	Classification with Compositional Inference
$y \leftarrow \text{argmax}_y \text{ score}(y)$	Find K best $\mathbf{z}$ and denote them as $\mathcal{Z} = \{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(K)}\}$
$l \leftarrow \text{loss\_flat}(y^*, y)$	$s.t.\ \forall\ i < j,\ \text{score}(\mathbf{z}^{(i)}) > \text{score}(\mathbf{z}^{(j)})$
$\mathbf{w} \leftarrow \text{update}(\mathbf{w}, l, y^*, y)$	$\mathbf{z}^{bad} \leftarrow \min_k \mathbf{z}^{(k)}\ s.t.\ \text{loss\_compo}(y^*, \mathbf{z}^{(k)}, x) > 0$
	(if such $\mathbf{z}^{bad}$ not found in $\mathcal{Z}$, skip parameter update for this.)
	If $\text{loss\_compo}(y^*, \mathbf{z}^*, x) > 0$
	$\mathbf{z}^{good} \leftarrow \min_k \mathbf{z}^{(k)}\ s.t.\ \text{loss\_compo}(y^*, \mathbf{z}^{(k)}, x) = 0$
	$z^* \leftarrow \mathbf{z}^{good}$
	(if such $\mathbf{z}^{good}$ not found in $\mathcal{Z}$, stick to the original z^*.)
	$l \leftarrow \text{loss\_compo}(y^*, \mathbf{z}^{bad}, x) - \text{loss\_compo}(y^*, \mathbf{z}^*, x)$
	$\mathbf{w} \leftarrow \text{update}(\mathbf{w}, l, \mathbf{z}^*, \mathbf{z}^{bad})$
Definitions of score functions and loss functions	
$\text{score}(y) := \mathbf{w} \cdot \mathbf{f}(x, y)$	$\text{score}(\mathbf{z}) := \sum_i \text{score}(z_i) := \sum_i \mathbf{w} \cdot \mathbf{f}(x, z_i, i)$
$\text{loss\_flat}(y^*, y) := \text{if } (y^* = y)\ 0 \text{ else } 1$	$\text{loss\_compo}(y^*, \mathbf{z}, x) := \text{if } (y^* = C(x, \mathbf{z}))\ 0 \text{ else } 1$

Figure 4.1: Training procedures. $y^* \in \{positive, negative\}$ denotes the true label for a given expression $x = x_1, ..., x_n$. $\mathbf{z}^*$ denotes the pseudo gold standard for hidden variables $\mathbf{z}$.

4.3 Learning-Based Methods

While we expect that a set of hand-written heuristic rules motivated by compositional semantics can be effective for determining the polarity of a sentiment-bearing expression, we do not expect them to be perfect. Interpreting natural language is such a complex task that writing a perfect set of rules would be extremely challenging. Therefore, a more ideal solution would be a learning-based method that can exploit ideas from compositional semantics while providing the flexibility to the rigid application of the heuristic rules. To this end, we present a novel learning-based approach that incorporates inference rules inspired by compositional semantics into the learning procedure (§3.2). To assess the effect of compositional semantics in the learning-based methods, we also experiment with a simple classification approach that does not incorporate compositional semantics (§3.1). The details of these two approaches are elabo-

rated in the following subsections.

4.3.1 Simple Classification (SC)

Given an expression x consisting of n words x_1, ..., x_n, the task is to determine the polarity $y \in \{positive, negative\}$ of x. In our simple binary classification approach, x is represented as a vector of features $\mathbf{f}(x)$, and the prediction y is given by $\text{argmax}_y \mathbf{w} \cdot \mathbf{f}(x, y)$, where $\mathbf{w}$ is a vector of parameters learned from training data. In our experiment, we use an online SVM algorithm called MIRA (Margin Infused Relaxed Algorithm) (Crammer and Singer, 2003)[5] for training.

For each x, we encode the following features:

- Lexical: We add every word x_i in x, and also add the lemma of x_i produced by the CASS partial parser toolkit (Abney, 1996).
- Dictionary: In order to mitigate the problem of unseen words in the test data, we add features that describe word categories based on the General Inquirer dictionary. We add this feature for each x_i that is not a stop word.

- Vote: We experiment with two variations of voting-related features: for SC-VOTE, we add a feature that indicates the dominant polarity of words in the given expression, without considering the effect of negators. For SC-NEGEX, we count the number of content-word negators as well as function-word negators to determine whether the final polarity should be flipped. Then we add a conjunctive feature that indicates the dominant polarity together

[5]We use the Java implementation of this algorithm available at http://www.seas.upenn.edu/~strctlrn/StructLearn /StructLearn.html.

For each token x_i,
  if x_i is a word in the negator lexicon
    then $z_i^* \leftarrow$ *negator*
  else if x_i is in the polarity lexicon as negative
    then $z_i^* \leftarrow$ *negative*
  else if x_i is in the polarity lexicon as positive
    then $z_i^* \leftarrow$ *positive*
  else
    then $z_i^* \leftarrow$ *none*

Figure 4.2: Constructing Soft Gold Standard $\mathbf{z}^*$

with whether the final polarity should be flipped. For brevity, we refer to SC-VOTE and SC-NEGEX collectively as SC.

Notice that in this simple binary classification setting, it is inherently difficult to capture the compositional structure among words in x, because $\mathbf{f}(x, y)$ is merely a flat bag of features, and the prediction is governed simply by the dot product of $\mathbf{f}(x, y)$ and the parameter vector w.

4.3.2 Classification with Compositional Inference (CCI)

Next, instead of determining y directly from x, we introduce hidden variables $\mathbf{z} = (z_1, ..., z_n)$ as intermediate decision variables, where $z_i \in \{positive, negative, negator, none\}$, so that z_i represents whether x_i is a word with positive/negative polarity, or a negator, or none of the above. For simplicity, we let each intermediate decision variable z_i (a) be determined independently from other intermediate decision variables, and (b) depend only on the input x, so that $z_i = \text{argmax}_{z_i} \mathbf{w} \cdot \mathbf{f}(x, z_i, i)$, where $\mathbf{f}(x, z_i, i)$ is the feature vector encoding around the ith word (described on the next page). Once we determine the intermedi-

ate decision variables, we apply the heuristic rules motivated by compositional semantics (from Table 2) in order to obtain the final polarity y of x. That is, $y = C(x, \mathbf{z})$, where C is the function that applies the compositional inference, either COMPOPR or COMPOMC.

For training, there are two issues we need to handle: the first issue is dealing with the hidden variables z. Because the structure of compositional inference C does not allow dynamic programming, it is intractable to perform exact expectation-maximization style training that requires enumerating all possible values of the hidden variables $\mathbf{z}$. Instead, we propose a simple and tractable training rule based on the creation of a *soft* gold standard for $\mathbf{z}$. In particular, we exploit the fact that in our task, we can automatically construct a reasonably accurate gold standard for $\mathbf{z}$, denoted as $\mathbf{z}^*$: as shown in Figure 4.2, we simply rely on the negator and polarity lexicons. Because $\mathbf{z}^*$ is not always correct, we allow the training procedure to replace $\mathbf{z}^*$ with potentially better assignments as learning proceeds: in the event that the soft gold standard $\mathbf{z}^*$ leads to an incorrect prediction, we search for an assignment that leads to a correct prediction to replace $\mathbf{z}^*$. The exact procedure is given in Figure 4.1, and will be discussed again shortly.

Figure 4.1 shows how we modify the parameter update rule of MIRA (Crammer and Singer, 2003) to reflect the aspect of compositional inference. In the event that the soft gold standard $\mathbf{z}^*$ leads to an incorrect prediction, we search for $\mathbf{z}^{good}$, the assignment with highest score that leads to a correct prediction, and replace $\mathbf{z}^*$ with $\mathbf{z}^{good}$. In the event of no such $\mathbf{z}^{good}$ being found among the K-best assignments of $\mathbf{z}$, we stick with $\mathbf{z}^*$.

The second issue is finding the assignment of $\mathbf{z}$ with the highest score($\mathbf{z}$) =

$\sum_i \mathbf{w} \cdot \mathbf{f}(x, z_i, i)$ that leads to an incorrect prediction $y = C(x, \mathbf{z})$. Because the structure of compositional inference C does not allow dynamic programming, finding such an assignment is again intractable. We resort to enumerating only over K-best assignments instead. If none of the K-best assignments of $\mathbf{z}$ leads to an incorrect prediction y, then we skip the training instance for parameter update.

Features. For each x_i in x, we encode the following features:

- Lexical: We include the current word x_i as well as the lemma of x_i produced by CASS partial parser toolkit (Abney, 1996). We also add a boolean feature to indicate whether the current word is a stop word.
- Dictionary: In order to mitigate the problem with unseen words in the test data, we add features that describe word categories based on the General Inquirer dictionary. We add this feature for each x_i that is not a stop word. We also add a number of boolean features that provide following properties of x_i using the polarity lexicon and the negator lexicon:
  - whether x_i is a function-word negator
  - whether x_i is a content-word negator
  - whether x_i is a negator of any kind
  - the polarity of x_i according to Wilson et al. (Wilson et al., 2005)'s polarity lexicon
  - the polarity of x_i according to the lexicon derived from the General Inquirer dictionary
  - conjunction of the above two features
- Vote: We encode the same vote feature that we use for SC-NEGEX described in § 3.1.

Table 4.3: Performance (in accuracy) on MPQA dataset.

Heuristic-Based							Learning-Based			
VOTE	NEG (1)	NEG (N)	NEG EX (1)	NEG EX (N)	COMPO MC	COMPO PR	SC VOTE	SC NEG EX	CCI COMPO MC	CCI COMPO PR
86.5	82.0	82.2	87.7	87.7	89.7	89.4	88.5	89.1	90.6	90.7

As in the heuristic-based compositional semantics approach (§ 2.2), we experiment with two variations of this learning-based approach: CCI-COMPOPR and CCI-COMPOMC, whose compositional inference rules are COMPOPR and COMPOMC respectively. For brevity, we refer to both variations collectively as CCI-COMPO.

4.4 Experiments

The experiments below evaluate our heuristic- and learning-based methods for subsentential sentiment analysis (§ 4.1). In addition, we explore the role of context by expanding the boundaries of the sentiment-bearing expressions (§ 4.2).

4.4.1 Evaluation with Given Boundaries

For evaluation, we use the Multi-Perspective Question Answering (MPQA) corpus (Wiebe et al., 2005), which consists of 535 newswire documents manually annotated with phrase-level subjectivity information. We evaluate on all strong

Table 4.4: Performance (in accuracy) on MPQA data set with varying boundaries of expressions.

Data	Heuristic-Based							Learning-Based			
	VOTE	NEG (1)	NEG (N)	NEG EX (1)	NEG EX (N)	COMPO MC	COMPO PR	SC VOTE	SC NEG EX	CCI COMPO MC	CCI COMPO PR
[-0,+0]	86.5	82.0	82.2	87.7	87.7	89.7	89.4	88.5	89.1	90.6	90.7
[-1,+1]	86.4	81.0	81.2	87.2	87.2	89.3	89.0	88.3	88.4	89.5	89.4
[-5,+5]	85.9	79.0	79.4	85.7	85.6	88.2	88.0	86.4	87.1	88.7	88.7
[-∞,+∞]	85.3	75.8	76.9	83.9	83.9	87.0	86.9	85.8	85.8	87.3	87.5

(i.e., intensity of expression is 'medium' or higher), sentiment-bearing (i.e., polarity is 'positive' or 'negative') expressions.[6] As a result, we can assume the boundaries of the expressions are given. Performance is reported using 10-fold cross-validation on 400 documents; a separate 135 documents were used as a development set. Based on pilot experiments on the development data, we set parameters for MIRA as follows: slack variable to 0.5, and the number of incorrect labels (constraints) for each parameter update to 1. The number of iterations (epochs) for training is set to 1 for simple classification, and to 4 for classification with compositional inference. We use $K = 20$ for classification with compositional inference.

Results. Performance is reported in Table 4.3. Interestingly, the heuristic-based methods NEG ($\sim$ 82.2%) that only consider function-word negators perform even worse than VOTE (86.5%), which does not consider negators. On the

[6]We discard expressions with confidence marked as 'uncertain'.

other hand, the NEGEX methods (87.7%) that do consider content-word negators as well as function-word negators perform better than VOTE. This confirms the importance of content-word negators for determining the polarities of expressions. The heuristic-based methods motivated by compositional semantics COMPO further improve the performance over NEGEX, achieving up to 89.7% accuracy. In fact, these heuristics perform even better than the SC learning-based methods (~ 89.1%). This shows that heuristics that take into account the compositional structure of the expression can perform better than learning-based methods that do not exploit such structure.

Finally, the learning-based methods that incorporate compositional inference CCI-COMPO (~ 90.7%) perform better than all of the previous methods. The difference between CCI-COMPOPR (90.7%) and SC-NEGEX (89.1%) is statistically significant at the .05 level by paired t-test. The difference between COMPO and any other heuristic that is not based on computational semantics is also statistically significant. In addition, the difference between CCICOMPOPR (learning-based) and COMPOMC (non-learning-based) is statistically significant, as is the difference between NEGEX and VOTE.

4.4.2 Evaluation with Noisy Boundaries

One might wonder whether employing additional context outside the annotated expression boundaries could further improve the performance. Indeed, conventional wisdom would say that it is necessary to employ such contextual information (e.g., Wilson et al. (2005)). In any case, it is important to determine whether our results will apply to more real-world settings where human-

annotated expression boundaries are not available.

To address these questions, we gradually relax our previous assumption that the exact boundaries of expressions are given: for each annotation boundary, we expand the boundary by x words for each direction, up to sentence boundaries, where $x \in \{1, 5, \infty\}$. We stop expanding the boundary if it will collide with the boundary of an expression with a different polarity, so that we can consistently recover the expression-level gold standard for evaluation. This expansion is applied to both the training and test data, and the performance is reported in Table 4.4. From this experiment, we make the following observations:

- Expanding the boundaries hurts the performance for any method. This shows that most of relevant context for judging the polarity is contained within the expression boundaries, and motivates the task of finding the boundaries of opinion expressions.
- The NEGEX methods perform better than VOTE only when the expression boundaries are reasonably accurate. When the expression boundaries are expanded up to sentence boundaries, they perform worse than VOTE. We conjecture this is because the scope of negators tends to be limited to inside of expression boundaries.
- The COMPO methods always perform better than any other heuristic-based methods. And their performance does not decrease as steeply as the NEGEX methods as the expression boundaries expand. We conjecture this is because methods based on compositional semantics can handle the scope of negators more adequately.
- Among the learning-based methods, those that involve compositional inference (CCI-COMPO) always perform better than those that do not (SC) for

any boundaries. And learning with compositional inference tend to perform better than the rigid application of heuristic rules (COMPO), although the relative performance gain decreases once the boundaries are relaxed.

4.5 Related Work

The task focused on in this work is similar to that of Wilson et al. (2005) in that the general goal of the task is to determine the polarity in context at a subsentence level. However, Wilson et al. (2005) formulated the task differently by limiting their evaluation to individual words that appear in their polarity lexicon. Also, their approach was based on a flat bag of features, and only a few examples of what we call content-word negators were employed.

Our use of compositional semantics for the task of polarity classification is preceded by Moilanen and Stephen (2007), but our work differs in that we integrate the key idea of compositional semantics into learning-based methods, and that we perform empirical comparisons among reasonable alternative approaches.

For comparison, we evaluated our approaches on the polarity classification task from SemEval-07 (Strapparava and Mihalcea, 2007). We achieve 88.6% accuracy with COMPOPR, 90.1% with SCNEGEX, and 87.6% with CCICOMPOMC. [7]

There are a number of possible reasons for our lower performance vs. Moilanen and Stephen (2007) on this data set. First, SemEval-07 does not include a

[7]For lack of space, we only report our performance on instances with strong intensities as defined in Moilanen and Stephen (2007), which amounts to only 208 test instances. The cross-validation set of MPQA contains 4.9k instances.

training data set for this task, so we use 400 documents from the MPQA corpus instead. In addition, the SemEval-07 data is very different from the MPQA data in that (1) the polarity annotation is given only at the sentence level, (2) the sentences are shorter, with simpler structure, and not as many negators as the MPQA sentences, and (3) there are many more instances with positive polarity than in the MPQA corpus.

Nairn et al. (2006) also employ a "polarity" propagation algorithm in their approach to the semantic interpretation of implicatives. However, their notion of polarity is quite different from that assumed here and in the literature on sentiment analysis. In particular, it refers to the degree of "commitment" of the author to the truth or falsity of a complement clause for a textual entailment task.

Mcdonald et al. (2007) use a structured model to determine the sentence-level polarity and the document-level polarity simultaneously. But decisions at each sentence level does not consider structural inference within the sentence.

Among the studies that examined content-word negators, Niu et al. (2005) manually collected a small set of such words (referred as "words that change phases"), but their lexicon was designed mainly for the medical domain and the type of negators was rather limited. Wilson et al. (2005) also manually collected a handful of content-word negators (referred as "general polarity shifters"), but not extensively. Moilanen and Stephen (2007) collected a more extensive set of negators semi-automatically using WordNet 2.1, but the empirical effect of such words was not explicitly investigated.

4.6 Summary of Chapter

In this work, we consider the task of determining the polarity of a sentiment-bearing expression, considering the effect of interactions among words or constituents in light of compositional semantics. We presented a novel learning-based approach that incorporates structural inference motivated by compositional semantics into the learning procedure. Our approach can be considered as a small step toward bridging the gap between computational semantics and machine learning methods. Our experimental results suggest that this direction of research is promising. Future research includes an approach that learns the compositional inference rules from data.

CHAPTER 5

LEXICON ADAPTATION AS CONSTRAINT OPTIMIZATION

5.1 Introduction

Polarity lexicons have been a valuable resource for sentiment analysis and opinion mining. In particular, they have been an essential ingredient for fine-grained sentiment analysis (e.g., Kim and Hovy (2004), Kennedy and Inkpen (2005), Wilson et al. (2005)). Even though the polarity lexicon plays an important role, it has received relatively less attention in previous research. In most cases, polarity lexicon construction is discussed only briefly as a preprocessing step for a sentiment analysis task (e.g., Hu and Liu (2004a), Moilanen and Stephen (2007)), but the effect of different alternative polarity lexicons is not explicitly investigated. Conversely, research efforts that focus on constructing a general purpose polarity lexicon (e.g., Takamura et al. (2005), Adreevskaia and Bergler (2006), Esuli and Sebastiani (2006), Rao and Ravichandran (2009)) generally evaluate the lexicon in isolation from any potentially relevant NLP task, and it is unclear how the new lexicon might affect end-to-end performance of a concrete NLP application.

It might even be unrealistic to expect that there can be a general-purpose lexical resource that can be effective across *all* relevant NLP applications, as general-purpose lexicons will not reflect domain-specific lexical usage. Indeed, Blitzer et al. (2007) note that the polarity of a particular word can carry opposite sentiment depending on the domain (e.g., Andreevskaia and Bergler (2008)).

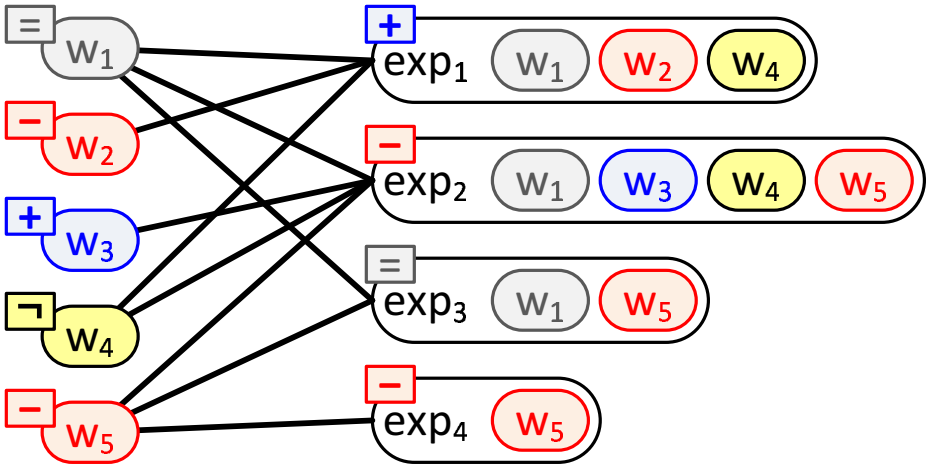

Figure 5.1: The relations among words and expressions. + indicates positive, - indicates negative, = indicates neutral, and ¬ indicates a negator.

5.1.1 Key Ideas

In this work, we propose a novel method based on integer linear programming to adapt an existing polarity lexicon into a new one to reflect the characteristics of the data more directly. In particular, our method considers the relations among words and opinion expressions collectively to derive the most likely polarity of each word for the given domain.

Figure 1 depicts the key insight of our approach using a bipartite graph. On the left hand side, each node represents a word, and on the right hand side, each node represents an opinion expression. There is an edge between a word w_i and an opinion expression e_j, if the word w_i appears in the expression e_j. We assume the possible polarity of each expression is one of the following three values: {*positive, neutral, negative*}, while the possible polarity of each word is one of: {*positive*, *neutral*, *negative* or *negator*}. Strictly speaking, *negator* is not a value for polarity, but we include them in our lexicon, because *valence shifters* or *negators*

have been shown to play an important role for sentiment analysis (e.g., Polanyi and Zaenen (2004), Moilanen and Stephen (2007), Choi and Cardie (2008)).

Typically, the ultimate goal of the sentiment analysis task is to determine the expression-level (or sentiment/ document-level) polarities, rather than the correct word-level polarities with respect to the domain. Therefore, word-level polarities can be considered as latent information. In this work, we show how we can improve the word-level polarities of a general-purpose polarity lexicon by utilizing the expression-level polarities, and in return, how the adapted word-level polarities can improve the expression-level polarities.

In Figure 1, there are two types of relations we could exploit when adapting a general-purpose polarity lexicon into a domain-specific one. The first are word-to-word relations within each expression. That is, if we are not sure about the polarity of a certain word, we can still make a guess based on the polarities of other words within the same expression and knowledge of the polarity of the expression. The second type of relations are word-to-expression relations: e.g., some words appear in expressions that take on a variety of polarities, while other words are associated with expressions of one polarity class or another.

In relation to previous research, analyzing word-to-word (intra-expression) relations is most related to techniques that determine expression-level polarity *in context* (e.g., Wilson et al. (2005)), while exploring word-to-expression (inter-expression) relations has connections to techniques that employ more of a global-view of corpus statistics (e.g., Kanayama and Nasukawa (2006)).[1]

While most previous research exploits only one or the other type of relation,

[1]In case of document-level polarity classification, word-to-expression relations correspond to word-to-document relations.

we propose a unified method that can exploit both types of semantic relation, while adapting a general purpose polarity lexicon into a domain specific one. We formulate our lexicon adaptation task using integer linear programming (ILP), which has been shown to be very effective when solving problems with complex constraints (e.g., Roth and tau Yih (2004), Denis and Baldridge (2007)). And the word-to-word and word-to-expression relations discussed above can be encoded as soft and hard constraints in ILP. Unfortunately, one class of constraint that we would like to encode (see Section 2) will require an exponentially many number of constraints when grounded into an actual ILP problem. We therefore propose an approximation scheme to make the problem more practically solvable.

5.1.2 Summary of Results

We evaluate the effect of the adapted lexicon in the context of a concrete NLP task: expression-level polarity classification. Experimental results show that our lexicon adaptation technique improves the accuracy of two competitive expression-level polarity classifiers from 64.2% - 70.4% to 67.0% - 71.2%.

5.2 An Integer Linear Programming Approach

In this section, we describe how we formulate the lexicon adaptation task using integer linear programming. Before we begin, we assume that we have a general-purpose polarity lexicon $\mathcal{L}$, and a polarity classification algorithm $f(e_l, \mathcal{L})$, that can determine the polarity of the opinion expression e_l based on

the words in e_l and the initial lexicon $\mathcal{L}$. The polarity classification algorithm $f(\cdot)$ can be either a heuristic-based one, or a machine-learning based one – we consider it as a black box for now.

5.2.1 Constraints for Word-level Polarities

For each word x_i, we define four binary variables: $x_i^+, x_i^=, x_i^-, x_i^\neg$ to represent positive, neutral, negative polarity, and negators respectively. If $x_i^\delta = 1$ for some $\delta \in \{+, =, -, \neg\}$, then the word x_i has the polarity δ. The following inequality constraint states that at least one polarity value must be chosen for each word.

$$x_i^+ + x_i^= + x_i^- + x_i^\neg >= 1 \tag{5.1}$$

If we allow only one polarity per word, then the above inequality constraint should be modified as an equality constraint. Although most words tend to associate with a single polarity, some can take on more than one polarity. In order to capture this observation, we introduce an auxiliary binary variable α_i for each word x_i. Then the next inequality constraint states that at most two polarities can be chosen for each word.

$$x_i^+ + x_i^= + x_i^- + x_i^\neg <= 1 + \alpha_i \tag{5.2}$$

Next we introduce the initial part of our objective function.

$$\text{maximize} \sum_i \Big(\; w_i^+ x_i^+ \; + \; w_i^= x_i^= \; + \; w_i^- x_i^- \; + \; w_i^\neg x_i^\neg \; - \; w_\alpha \alpha_i \; \Big) + \; \cdots \tag{5.3}$$

For the auxiliary variable α_i, we apply a constant weight w_α to discourage ILP from choosing more than one polarity for each word. We can allow more than two polarities for each word, by adding extra auxiliary variables and weights. For each variable x_i^δ, we define its weight w_i^δ, which indicates how likely it is that word x_i carries the polarity δ. We define the value of w_i^δ using two different types of information as follows:

$$w_i^\delta \; := \; {}^{\mathcal{L}}w_i^\delta \; + \; {}^{C}w_i^\delta$$

where ${}^{\mathcal{L}}w_i^\delta$ is the degree of polarity δ for word x_i determined by the general-purpose polarity lexicon $\mathcal{L}$, and ${}^{C}w_i^\delta$ is the degree of polarity δ determined by the corpus statistics as follows:[2]

$${}^{C}w_i^\delta \; := \; \frac{\text{\# of } x_i \text{ in expressions with polarity } \delta}{\text{\# of } x_i \text{ in the corpus } C}$$

Note that the occurrence of word x_i in an expression e_j with a polarity δ does not necessarily mean that the polarity of x_i should also be δ, as the interpretation of the polarity of an expression is more than just a linear sum of the word-level polarities (e.g., Moilanen and Stephen (2007)). Nonetheless, not all expressions

[2]If a word x_i is in an expression that is not an opinion, then we count it as an occurrence with neutral polarity.

require a complicated inference procedure to determine their polarity. Therefore, ${}^{C}w_i^{\delta}$ still provides useful information about the likely polarity of each word based on the corpus statistics.

From the perspective of Chomskyan linguistics, the weights ${}^{\mathcal{L}}w_i^{\delta}$ based on the prior polarity from the lexicon can be considered as having a "competence" component , while ${}^{C}w_i^{\delta}$ derived from the corpus counts can be considered as a "performance" component (Chomsky, 1965).

5.2.2 Constraints for Content-word Negators

Next we describe a constraint that exploits knowledge of the typical distribution of *content-word negators* in natural language. Content-word negators are words that are not function words, but act semantically as negators (Choi and Cardie, 2008).[3] Although it is possible to artificially construct a very convoluted sentence with lots of negations, it is unlikely for multiple layers of negations to appear very often in natural language (et al., 1996). Therefore, we allow at most one content-word negator for each expression e_l. Because we do not restrict the number of function-word negators, our constraint still gives room for multiple layers of negations.

$$\sum_{i \in \mu(e_l)} x_i^{\neg} <= 1 \tag{5.4}$$

In the above constraint, $\mu(e_l)$ indicates the set of indices of content words appearing in e_l. For instance, if $i \in \mu(e_l)$, then x_i appears in e_l. This constraint can

[3]Examples of content-word negators are *destroy*, *eliminate*, *prevent* etc.

be polished further to accommodate longer expressions where multiple content-word negators are more likely to appear, by adding a separate constraint with a sliding window.

5.2.3 Constraints for Expression-level Polarities

Before we begin, we introduce $\pi(e_l)$ that will be used often in the remaining section. For each expression e_l, we define $\pi(e_l)$ to be the set of content words appearing in e_l, together with the most likely polarity proposed by a general-purpose polarity lexicon $\mathcal{L}$. For instance, if $x_i^+ \in \pi(e_l)$, then the polarity of word x_i is + according to $\mathcal{L}$.

Next we encode constraints that consider expression-level polarities. If the polarity classification algorithm $f(e_l, \mathcal{L})$ makes an incorrect prediction for e_l using the original lexicon $\mathcal{L}$, then we need to encourage ILP to fix the error by suggesting different word-level polarities. We capture this idea by the following constraint:

$$\sum_{x_i^\delta \in \pi(e_l)} x_i^\delta \ <= \ |\pi(e_l)| - 1 + \beta_l \tag{5.5}$$

The auxiliary binary variable β_l is introduced for each e_l so that the assignment $\pi(e_l)$ does not have to be changed if paying for the cost w_β in the objective function. (See equation (5.10).) That is, suppose the ILP solver assigns '1' to all variables in $\phi(e_l)$, (which corresponds to keeping the original lexicon as it is for all words in the given expression e_l), then the auxiliary variable β_l must be also set as '1' in order to satisfy the constraint (5). Because β_l is associated with a

negative weight in the objective function, doing so will act against maximizing the objective function. This way, we discourage the ILP solver to preserve the original lexicon as it is.

To verify the constraint (5) further, suppose that the ILP solver assigns '1' for all variables in $\phi(e_l)$ except for one variable. (Notice that doing so corresponds to proposing a new polarity for one of the words in the given expression e_l.) Then the constraint (5) will hold regardless of whether the ILP solver assigns '0' or '1' to β_l. Because β_l is associated with a negative weight in the objective function, the ILP solver will then assign '0' to β_l to maximize the objective function. In other words, we encourage the ILP solver to modify the original lexicon for the given expression e_l.

We use this type of *soft* constraint in order to cope with the following two noise factors: first, it is possible that some annotations are noisy. Second, $f(e_l, \mathcal{L})$ is not perfect, and might not be able to make a correct prediction even with the correct word-level polarities.

Next we encode a constraint that is the opposite of the previous one. That is, if the polarity classification algorithm $f(e_l, \mathcal{L})$ makes a correct prediction on e_l using the original lexicon $\mathcal{L}$, then we encourage ILP to keep the original word-level polarities for words in e_l.

$$\sum_{x_i^\delta \in \pi(e_l)} x_i^\delta \;>=\; |\pi(e_l)| - |\pi(e_l)|\beta_l \tag{5.6}$$

Interpretation of constraint (6) with the auxiliary binary variable β_l is similar to that of constraint (5) elaborated above.

Notice that in equation (5.5), we encouraged ILP to fix the current lexicon $\mathcal{L}$ for words in e_l, but we have not specified the consequence of a modified lexicon ($\mathcal{L}'$) in terms of expression-level polarity classification $f(e_l, \mathcal{L}')$. Certain changes to $\mathcal{L}$ might not fix the prediction error for e_l, and those might even cause extra incorrect predictions for other expressions. Then it would seem that we need to replicate constraints (5.5) & (5.6) for all permutations of word-level polarities. However, doing so would incur exponentially many number of constraints ($4^{|e_l|}$) for each expression.[4]

To make the problem more practically solvable, we only consider changes to the lexicon that are within edit-one distance with respect to $\pi(e_l)$. More formally, let us define $\pi'(e_l)$ to be the set of content words appearing in e_l, together with the most likely polarity proposed by a modified polarity lexicon $\mathcal{L}'$. Then we need to consider all $\pi'(e_l)$ such that $|\pi'(e_l) \cap \pi(e_l)| = |\pi(e_l)| - 1$. There are $(4-1)|e_l|$ number of different $\pi'(e_l)$, and we index them as $\pi'_k(e_l)$. We then add following constraints similarly as equation (5.5) & (5.6):

$$\sum_{x_i^\delta \in \pi'_k(e_l)} x_i^\delta \;<=\; |\pi'_k(e_l)| - 1 + \beta_{(l,k)} \tag{5.7}$$

if the polarity classification algorithm $f(\cdot)$ makes an incorrect prediction based on $\pi'_k(e_l)$. And,

$$\sum_{x_i^\delta \in \pi'_k(e_l)} x_i^\delta \;>=\; |\pi'_k(e_l)| - |\pi'_k(e_l)|\beta_{(l,k)} \tag{5.8}$$

[4]For certain simple polarity classification algorithm $f(e_l, \mathcal{L})$, it is possible to write polynomially many number of constraints. However our approach intends to be more general by treating $f(e_l, \mathcal{L})$ as a black box, so that algorithms that do not factor nicely can also be considered as an option.

if the polarity classification algorithm $f(\cdot)$ makes a correct prediction based on $\pi'_k(e_l)$. Remember that none of the constraints (5.5) - (5.8) enforces assignment $\pi(e_l)$ or $\pi'_k(e_l)$ as a hard constraint. In order to enforce at least one of them to be chosen, we add the following constraint:

$$\sum_{x_i^\delta \in \pi(e_l)} x_i^\delta \;>=\; |\pi(e_l)| - 1 \tag{5.9}$$

This constraint ensures that the modified lexicon $\mathcal{L}'$ is not drastically different from $\mathcal{L}$. Assuming that the initial lexicon $\mathcal{L}$ is a reasonably good one, constraining the search space for $\mathcal{L}'$ will regulate that $\mathcal{L}'$ does not turn into a degenerative one that overfits to the current corpus C.

5.2.4 Objective Function

Finally, we introduce our full objective function.

$$\begin{aligned} \text{maximize} \sum_i \Big(& w_i^+ x_i^+ + w_i^= x_i^= \\ & + w_i^- x_i^- + w_i^\neg x_i^\neg \\ & - w_\alpha \alpha_i \Big) \\ & - \sum_l w_\beta \rho_l \beta_l \\ & - \sum_{l,k} w_\beta \rho_{(l,k)} \beta_{(l,k)} \end{aligned} \tag{5.10}$$

We have already described the first part of the objective function (equation (5.3)), thus we only describe the last two terms here. w_β is defined similarly as

w_α; it is a constant weight that applies for any auxiliary binary variable β_l and $\beta_{(l,k)}$.

We further define ρ_l and $\rho_{(l,k)}$ as secondary weights, or *amplifiers* to adjust the constant weight w_β. To enlighten the motivation behind the amplifiers ρ_l and $\rho_{(l,k)}$, we bring out the following observations:

1. Among the incorrect predictions for expression-level polarity classification, some are more incorrect than the other. For instance, classifying positive class to negative class is more wrong than classifying positive class to neutral class. Therefore, the cost of not fixing very incorrect predictions should be higher than the cost of not fixing less incorrect predictions. (See [R2] and [R3] in Table 5.1.)
2. If the current assignment $\pi(e_l)$ for expression e_l yields a correct prediction using the classifier $y(e_l, \mathcal{L})$, then there is not much point in changing $\mathcal{L}$ to $\mathcal{L}'$, even if $y(e_l, \mathcal{L}')$ also yields a correct prediction. In this case, we would like to assign slightly higher confidence in the original lexicon $\mathcal{L}$ then the new one $\mathcal{L}'$. (See [R1] in Table 5.1.)
3. Likewise, if the current assignment $\pi(e_l)$ for expression e_l yields an incorrect prediction using the classifier $y(e_l, \mathcal{L})$, then there is not much point in changing $\mathcal{L}$ to $\mathcal{L}'$, if $y(e_l, \mathcal{L}')$ also yields an equally incorrect prediction. Again we assign slightly higher confidence in the original lexicon $\mathcal{L}$ than the new one $\mathcal{L}'$ in such cases. (Compare each row in [R2] with a corresponding row in [R3] in Table 5.1.)

To summarize, for correct predictions, the degree of ρ determines the degree of cost of (undesirably) altering the current lexicon for e_l. For incorrect predic-

Table 5.1: The value of amplifiers ρ_l and $\rho_{(l,k)}$.

[R1]	If $\pi(e_l)$ correct	ρ_l	$\leftarrow 1.5$
	If $\pi'_k(e_l)$ correct	$\rho_{(l,k)}$	$\leftarrow 1.0$
[R2]	If $\pi(e_l)$ very incorrect	ρ_l	$\leftarrow 1.0$
	If $\pi(e_l)$ less incorrect	ρ_l	$\leftarrow 0.5$
[R3]	If $\pi'_k(e_l)$ very incorrect	$\rho_{(l,k)}$	$\leftarrow 1.5$
	If $\pi'_k(e_l)$ less incorrect	$\rho_{(l,k)}$	$\leftarrow 1.0$

tions, the degree of ρ determines the degree of cost of not fixing the current lexicon for e_l.

5.3 Experiments

In the experiment section, we seek for answers for the following questions:

Q1 What is the effect of a polarity lexicon on the expression-level polarity classification task? In particular, is it useful when using a machine learning technique that might be able to learn the necessary polarity information just based on the words in the training data, without consulting a dictionary? (Section 3.1)

Q2 What is the effect of an adapted polarity lexicon on the expression-level polarity classification task? (Section 3.2)

Notice that we include the *neutral* polarity in the polarity classification. It makes our task much harder (e.g., Wilson et al. (2009)) than those that assume inputs

are guaranteed to be either strongly positive or negative (e.g., Pang et al. (2002), Choi and Cardie (2008)). But in practice, one cannot expect that a given input is strongly polar, as automatically extracted opinions are bound to be noisy. Furthermore, Wiebe et al. (2005) discuss that some opinion expressions do carry a neutral polarity.

We experiment with the Multi-Perspective Question Answering (MPQA) corpus (Wiebe et al., 2005) for evaluation. It contains 535 newswire documents annotated with phrase-level subjectivity information. We evaluate on all opinion expressions that are known to have high level of inter-annotator agreement. That is, we include opinions with intensity marked as 'medium' or higher, and exclude those with annotation confidence marked as 'uncertain'. To focus our study on the direct influence of the polarity lexicon upon the sentiment classification task, we assume the boundaries of the expressions are given. However, our approach can be readily used in tandem with a system that extracts opinion expressions (e.g., Kim and Hovy (2005a), Breck et al. (2007)). Performance is reported using 10-fold cross-validation on 400 documents, and a separate 135 documents were used as a development set. For the general-purpose polarity lexicon, we expand the polarity lexicon of Wilson et al. (2005) with General Inquirer dictionary as suggested by Choi and Cardie (2008).

We report the performance in two measures: *accuracy* for 3-way classification, and *average error distance*. The reason why we consider *average error distance* is because classifying a positive class into a negative class is worse than classifying a positive class into a neutral one. We define the error distance between 'neutral' class and any other class as 1, while the error distance between 'positive' class and 'negative' class as 2. If a predicted polarity is correct, then the

error distance is 0. We compute the error distance of each prediction and take the average over all predictions in the test data.

5.3.1 Effect of a Polarity Lexicon

To verify the effect of a polarity lexicon on the expression-level polarity classification task, we experiment with simple classification-based machine learning technique. We use the Mallet (McCallum, 2002) implementation of Conditional Random Fields (CRFs) (Lafferty et al., 2001).[5] To highlight the influence of a polarity lexicon, we compare the performance of CRFs with and without features derived from polarity lexicons.

Features: We encode basic features as words and lemmas for all content words in the given expression. The performance of CRFs using only the basic features are given in the first row of the Table 5.2. Next we encode features derived from polarity lexicons as follows.

- The output of Vote & Flip algorithm. (Section 3.2 & Figure 5.2.)
- Number of positive, neutral, negative, and negators in the given expression.
- Number of positive (or negative) words in conjunction with number of negators.
- (boolean) Whether the number of positive words dominates negative ones.

[5]We use the CRF implementation of Mallet (McCallum, 2002) with Markov-order 0, which is equivalent to Maximum Entropy models (Berger et al., 1996).

Table 5.2: Effect of a polarity lexicon on expression-level classification using CRFs

	Accuracy	Avg. Error Distance
Without Lexicon	63.9	0.440
With Lexicon	70.4	0.334

- (boolean) Whether the number of negative words dominates positive ones.
- (boolean) None of the above two cases
- Each of the above three boolean values in conjunction with the number of negators.

Results: Table 5.2 shows the performance of CRFs with and without features that consult the general-purpose lexicon. As expected, CRFs can perform reasonably well (accuracy = 63.9%) even without consulting the dictionary, by learning directly from the data. However, having the polarity lexicon boosts the performance significantly (accuracy = 70.4%), demonstrating that lexical resources are very helpful for fine-grained sentiment analysis. The difference in performance is statistically significant by paired t-test for both accuracy ($p < 0.01$) and average error distance ($p < 0.01$).

5.3.2 Effect of Adapting a Polarity Lexicon

In this section, we assess the quality of the adapted lexicon in the context of an expression-level polarity classification task. In order to perform the lexicon

For each expression e_i,
  $nPositive \leftarrow$ # of positive words in e_i
  $nNeutral \leftarrow$ # of neutral words in e_i
  $nNegative \leftarrow$ # of negative words in e_i
  $nNegator \leftarrow$ # of negating words in e_i

  if ($nNegator$ % 2 = 0)
    then $fFlipPolarity \leftarrow false$
  else
    then $fFlipPolarity \leftarrow true$

  if ($nPositive > nNegative$) & $\neg\, fFlipPolarity$
    then Polarity(e_i) $\leftarrow$ *positive*
else if ($nPositive > nNegative$) & $fFlipPolarity$
    then Polarity(e_i) $\leftarrow$ *negative*
else if ($nPositive < nNegative$) & $\neg\, fFlipPolarity$
    then Polarity(e_i) $\leftarrow$ *negative*
else if ($nPositive < nNegative$) & $fFlipPolarity$
    then Polarity(e_i) $\leftarrow$ *neutral*
else if $nNeutral > 0$
    then Polarity(e_i) $\leftarrow$ *neutral*
else
    then Polarity(e_i) $\leftarrow$ $default\_polarity$ (the most prominent polarity in the corpus)

Figure 5.2: Vote & Flip Algorithm

adaptation via ILP, we need an expression-level polarity classification algorithm $f(e_l, \mathcal{L})$ as described in Section 2. According to Choi and Cardie (2008), voting algorithms that recognize content-word negators achieve a competitive performance, so we will use a variant of it for simplicity. Because none of the algorithms proposed by Choi and Cardie (2008) is designed to handle the neutral polarity, we invent our own version as shown in Figure 5.2.

It might look a bit complex at first glance, but the intuition is simple. The variable $fFlipPolarity$ determines whether we need to flip the overall majority polarity based on the number of negators in the given expression. If the positive

(or negative) polarity words dominate the given expression, and if there is no need to flip the majority polarity, then we take the positive (or negative) polarity as the overall polarity. If the positive (or negative) polarity words dominate the given expression, and if we need to flip the majority polarity, then we take the negative (or neutral) polarity as the overall polarity.

Notice that the result of flipping the negative polarity is *neutral*, not *positive*. In our pilot study, we found that this strategy works better than flipping the negative polarity to positive.[6] Finally, if the number of positive words and the negative words tie, and there is any neutral word, then we assign the neutral polarity. In this case, we don't worry if there is a negator, because flipping a neutral polarity would still result in a neutral polarity. If none of above condition is met, than we default to the most prominent polarity of the data, which is the negative polarity in the MPQA corpus. We name this simple algorithm as Vote & Flip algorithm. The performance is shown in the first row in Table 2.

Next we describe the implementation part of the ILP. For 10 fold-cross validation, we formulate the ILP problem using the training data (360 documents), and then test the effect of the adapted lexicon on the remaining 40 documents. We include only those content words that appeared more than 3 times in the training data. From the pilot test using the development set, we picked the value of w_β as 0.1. We found that having the auxiliary variables α_l which allow more than one polarity per word does not necessarily help with the performance, so we omitted them. We suspect it is because the polarity classifiers we experimented with is not highly capable of disambiguating different lexical usages and select the right polarity for a given context. We use CPLEX integer

[6]This finding is not surprising. For instance, if we consider the polarity of *"She did not get hurt much from the accident."*, it can be viewed as neutral; although it is good that one did not hurt much, it is still bad that there was an accident. Hence it gives a mixed feeling, which corresponds to the neutral polarity.

programming solver to solve our ILP problems. On a machine with 4GHz CPU, it took several minutes to solve each ILP problem.

In order to assess the effect of the adapted lexicon using CRFs, we need to first train the CRFs model. Using the same training set used for the lexicon adaptation would be suboptimal, because the features generated from the adapted lexicon will be unrealistically good in that particular data. Therefore, we prepared a separate training data for CRFs using 135 documents from the development set.

Results: Table 5.3 shows the comparison of the original lexicon and the adapted lexicon in terms of polarity classification performance using the Vote & Flip algorithm. The adapted lexicon improves the accuracy as well as reducing the average error distance. The difference in performance is statistically significant by paired t-test for both accuracy ($p < 0.01$) and average error distance ($p < 0.01$).

Table 5.4 shows the comparison of the original lexicon and the adapted lexicon using CRFs. The improvement is not as substantial as that of Vote & Flip algorithm but the difference in performance is also statistically significant for both accuracy ($p = 0.03$) and average error distance ($p = 0.04$).

5.4 Related Work

There are a number of previous work that focus on building polarity lexicons (e.g., Takamura et al. (2005), Kaji and Kitsuregawa (2007), Rao and Ravichan-

Table 5.3: Effect of an adapted polarity lexicon on expression-level classification using the Vote & Flip Algorithm

	Accuracy	Avg. Error Distance
Original Lexicon	64.2	0.395
Adapted Lexicon	67.0	0.365

Table 5.4: Effect of an adapted polarity lexicon on expression-level classification using CRFs

	Accuracy	Avg. Error Distance
Original Lexicon	70.4	0.334
Adapted Lexicon	71.2	0.327

dran (2009)). But most of them evaluated their lexicon in isolation from any potentially relevant NLP task, and it is unclear how the new lexicon might affect end-to-end performance of a concrete NLP application. Our work differs in that we try to draw a bridge between general purpose lexical resources and a domain-specific NLP application.

Kim and Hovy (2005a) and Banea et al. (2008) present bootstrapping methods to construct a subjectivity lexicon and measure the effect of the new lexicon for sentence-level subjectivity classification. However, their lexicons only tell whether a word is a subjective one, but not the polarity of the sentiment. Furthermore, the construction of lexicon is still an isolated step from the classification task. Our work on the other hand allows the classification task to directly influence the construction of lexicon, enabling the lexicon to be adapted for a concrete NLP application and for a specific domain.

Wilson et al. (2005) pioneered the expression-level polarity classification task using the MPQA corpus. The experimental results are not directly comparable to ours, because Wilson et al. (2005) limit the evaluation only for the words that appeared in their polarity lexicon. Choi and Cardie (2008) also focus on the expression-level polarity classification, but their evaluation setting is not as practical as ours in that they assume the inputs are guaranteed to be either strongly positive or negative.

5.5 Summary of Chapter

In this work, we present a novel lexicon adaptation technique based on integer linear programming to reflect the characteristics of the domain more directly. In particular, our method collectively considers the relations among words and opinion expressions to derive the most likely polarity of each lexical item for the given domain. We evaluate the effect of our lexicon adaptation technique in the context of a concrete NLP application: expression-level polarity classification. The positive results from our experiments encourage further research for lexical resource adaptation techniques.

CHAPTER 6

COREFERENCE RESOLUTION

6.1 Introduction

Undirected graphical models such as Conditional Random Fields (CRFs) (Lafferty et al., 2001) have shown great success for problems involving structured output variables (e.g. Wellner et al. (2004), Finkel et al. (2005)). For many real-world NLP applications, however, the required graph structure can be very complex, and computing the global normalization factor even approximately can be extremely hard. Previous approaches for training CRFs have either (1) opted for a training method that no longer maximizes the likelihood, (e.g. McCallum and Wellner (2004), Roth and Yih (2005)) [1], or (2) opted for a simplified graph structure to avoid intractable global normalization (e.g. (Roth and Yih, 2005), (Wellner et al., 2004)).

Solutions of the first type replace the computation of the global normalization factor $\sum_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$ with $\text{argmax}_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$ during training, since finding an argmax of a probability distribution is often an easier problem than finding the entire probability distribution. Training via the voted perceptron algorithm (Collins, 2002) or using a max-margin criterion also correspond to the first option (e.g. McCallum and Wellner (2004), Finley and Joachims (2005)). But without the global normalization, the maximum-likelihood criterion motivated by the maximum entropy principle (Berger et al., 1996) is no longer a feasible option as an optimization criterion.

[1] Both McCallum and Wellner (2004) and Roth and Yih (2005) used the voted perceptron algorithm (Collins, 2002) to train intractable CRFs.

The second solution simplifies the graph structure for training, and applies complex global inference only for testing. In spite of the discrepancy between the training model and the testing model, it has been empirically shown that (1) performing global inference only during testing can improve performance (e.g. Finkel et al. (2005), Roth and Yih (2005)), and (2) full-blown global training can often perform worse due to insufficient training data (e.g. Punyakanok et al. (2005)). Importantly, however, attempts to reduce the discrepancy between the training and test models — by judiciously adding the effect of global inference to the training — have produced substantial performance improvements over locally trained models (e.g. Cohen and Carvalho (2005), Sutton and McCallum (2005a)).

In this work, we present *structured local training*, a novel training procedure for maximum-likelihood training of undirected graphical models, such as CRFs. The procedure maximizes likelihood while exploiting the benefits of global inference during training by capturing the interactions between local inference and global inference via hidden variables.

A Motivating Example for Coreference Resolution

In this section, we present an example of the coreference resolution problem to motivate our approach. It has been shown that global inference-based training for coreference resolution outperforms training with local inference only (e.g. Finley and Joachims (2005), McCallum and Wellner (2004)). In particular, the output of coreference resolution must obey equivalence relations, and exploiting such structural constraints on the output space during training can improve performance. Consider the coreference resolution task for the following text.

> It was after the passage of this act, that Mary$^{(1)}$'s attitude towards Elizabeth$^{(1)}$ became overtly hostile. The deliberations surrounding the act seem to have revived all Mary's memories of the humiliations she had suffered at the hands of Anne Boleyn. At the same time, Elizabeth$^{(2)}$'s continuing prevarications over religion confirmed that **she** was indeed her mother's daughter.

In the above text, the "*she*" in the last sentence is coreferent with both mentions of "*Elizabeth*". However, when we consider "*she*" and "*Elizabeth*$^{(1)}$" in isolation from the remaining coreference chain, it can be difficult for a machine learning method to determine whether the pair is coreferent or not. Indeed, such a pair may not look very different from the pair "*she*" and "*Mary*$^{(1)}$" in terms of feature vectors. It is much easier, however, to determine that "*she*" and "*Elizabeth*$^{(2)}$" are coreferent, or that "*Elizabeth*$^{(1)}$" and "*Elizabeth*$^{(2)}$" are coreferent. Only by taking the transitive closure of these pairwise coreference relations does it become clear that "*she*" and "*Elizabeth*$^{(1)}$" are coreferent. In other words, global training might handle potentially confusing coreference cases better because it allows parameter learning (for each pairwise coreference decision) to be informed by global inference.

6.1.1 Key Ideas

We argue that, with appropriate modification to the learning instances, local training is adequate for the coreference resolution task. Specifically, we propose that confusing pairs in the training data — such as "*she*" and "*Elizabeth*$^{(1)}$" — be learned as *not-coreferent*, so long as the global inference step can fix this error by

exploiting the structure of the output space, i.e. by exploiting the equivalence relations. This is the key idea of *structured local training*, a novel training procedure for maximum-likelihood training of undirected graphical models, such as CRFs. The procedure maximizes likelihood while exploiting the benefits of global inference during training by capturing the interactions between local inference and global inference via hidden variables.

Furthermore, we introduce *biased potential functions* that redefine the likelihood for CRFs so that the performance of CRFs trained under the maximum likelihood criterion correlates better empirically with the preferred evaluation measures such as F-score and MUC-score.

We focus on the problem of coreference resolution; however, our approaches are general and can be extended to other NLP applications with structured output. Our approaches also extend to non-conditional graphical models such as Markov Random Fields.

6.1.2 Summary of Results

In experiments on two coreference data sets, structured local training reduces the error rate significantly (3.5%) for one coreference data set and minimally ($\leq$ 1%) for the other. Experiments using biased potential functions increase recall uniformly and significantly for both data sets and both task-specific evaluation measures. Results for the combination of the two techniques are promising, but mixed: pairwise F1 increases by 0.8-5.5% for both data sets; MUC F1 increases by 3.5% for one data set, but slightly hurts performance for the second data set.

6.2 Structured Local Training

6.2.1 Definitions

For clarity, we define the following terms that will be used throughout this chapter.

- **local inference:** [2] Inference factored into smaller independent pieces, without considering the structure of the output space.
- **global inference:** Inference applied on the entire set of output variables, considering the structure of the output space.
- **local training:** Training that does not invoke global inference at each iteration.

- **global training:** Training that does invoke global inference at each iteration.

6.2.2 A Hidden-Variable Model

We now present a general description of *structured local training*. Let $\mathbf{y}$ be a vector of output variables for structured output, and let $\mathbf{x}$ be a vector of input variables. In order to capture the interactions between global inference and local inference, we introduce hidden variables $\mathbf{h}$, $|\mathbf{h}| = |\mathbf{y}|$, so that the global inference for $p(\mathbf{y}, \mathbf{h}|\mathbf{x})$ can be factored into two components using the product rule, as follows:

$$\begin{aligned} p(\mathbf{y}, \mathbf{h}|\mathbf{x}) &= p(\mathbf{y}|\mathbf{h}, \mathbf{x})\, p(\mathbf{h}|\mathbf{x}) \\ &= p(\mathbf{y}|\mathbf{h})\, p(\mathbf{h}|\mathbf{x}) \end{aligned}$$

[2] In this work, inference refers to the operation of finding the *argmax* in particular.

The second component $p(\mathbf{h}|\mathbf{x})$ on the right hand side corresponds to the local model, for which the inference factorizes into smaller independent pieces, e.g. $\text{argmax}_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}) = \{\text{argmax}_{h_i} \phi(h_i, \mathbf{x})\}$. And the first component $p(\mathbf{y}|\mathbf{h}, \mathbf{x})$ on the right hand side corresponds to the global model, whose inference may not factorize nicely. Further, we assume that $\mathbf{y}$ is independent of $\mathbf{x}$ given $\mathbf{h}$, so that $p(\mathbf{y}|\mathbf{h}, \mathbf{x}) = p(\mathbf{y}|\mathbf{h})$. That is to say, $\mathbf{h}$ captures sufficient information from $\mathbf{x}$, so that given $\mathbf{h}$, global inference of $\mathbf{y}$ only depends on $\mathbf{h}$. The quantity of $p(\mathbf{y}|\mathbf{x})$ then is given by marginalizing out $\mathbf{h}$ as follows:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{y}, \mathbf{h}|\mathbf{x})$$

Intuitively, the hidden variables $\mathbf{h}$ represent the local decisions that can lead to a good $\mathbf{y}$ after global inference is applied. In the case of coreference resolution, one natural factorization would be that global inference is a clustering algorithm, and local inference is a classification decision on each pair of noun phrases (or mentions).[3] In this work, we assume that we only parameterize the local model $p(\mathbf{h}|\mathbf{x})$, although it would be possible to extend the parameterization to the global model as well, depending on the particular application under consideration. The similarity between a pair of mentions is parameterized via log-linear models. However, once we have the similarity scores extracted via local inference, the clustering algorithm does not require further parameterization.

For training, we apply the standard Expectation-Maximization (EM) algorithm (Dempster et al., 1977) as follows:

[3]Formally, we define each $y_i \in \mathbf{y}$ to be the coreference decision for the ith pair of mentions, and $x_i \in \mathbf{x}$ be the input regarding the ith pair of mentions. Then h_i corresponds to the local coreference decision that can lead to a good coreference decision y_i after the clustering algorithm has been applied.

- E Step: Compute a distribution

$$\widetilde{P}^{(t)} = P(\mathbf{h}|\mathbf{y}, \mathbf{x}, \theta^{(t-1)})$$

- M Step: Set $\theta^{(t)}$ to θ that maximizes

$$E_{\widetilde{P}^{(t)}} [\log P(\mathbf{y}, \mathbf{h}|\mathbf{x}, \theta)]$$

By repeatedly applying the above two steps for $t = 1, 2, ...,$ the value of θ converges to the local maxima of the conditional log likelihood $L(\theta) = \log P(\mathbf{y}|\mathbf{x}, \theta)$.

6.2.3 Application to Coreference Resolution

For $y_i \in \mathbf{y}$ (and $h_i \in \mathbf{h}$) in the coreference resolution task, $y_i = 1$ (and $h_i = 1$) corresponds to ith pair of mentions being coreferent, and $y_i = 0$ (and $h_i = 0$) corresponds to ith pair being not coreferent.

[Local Model $P(\mathbf{h}|\mathbf{x})$] For the local model, we define cliques as individual nodes,[4] and parameterize each clique potential as

$$\phi(h_i, \mathbf{x}) = \phi(h_i, x_i) = \exp \sum_k \lambda_k f_k(h_i, x_i)$$

Let $\Phi(\mathbf{h}|\mathbf{x}) \equiv \prod_i \phi(h_i, x_i)$. Then,

$$P(\mathbf{h}|\mathbf{x}) = \frac{\Phi(\mathbf{h}, \mathbf{x})}{\sum_{\mathbf{h}} \Phi(\mathbf{h}, \mathbf{x})}$$

Notice that in this model, finding $\text{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{x})$ corresponds to simply finding $\text{argmax}_{h_i} \phi(h_i, x_i)$ independently for each $h_i \in \mathbf{h}$.

[4] Each node in the graphical representation of CRFs corresponds to the coreferent decision for each pair of mentions. This corresponds to the "Model 3" of McCallum and Wellner (2004).

ALGORITHM-1

INPUT: $\mathbf{x}$, true labeling $\mathbf{y}^*$, current local model $P(\mathbf{h}|\mathbf{x})$
GOAL: Find the highest confidence labeling $\mathbf{y}'$
such that $\mathbf{y}^*$ = single-link-clustering($\mathbf{y}'$)

$\mathbf{h}^* \leftarrow \text{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{x})$
$\mathbf{h}' \leftarrow$ single-link-clustering($\mathbf{h}^*$)
construct a graph $G = (V, E)$, where
  $E = \{h'_i : h'_i \in \mathbf{h}' \text{ s.t. } y^*_i = 1\}$
  $V = \{v : v \text{ is a NP referred by a } h'_i \in E\}$
  with edge cost $cost_{h'_i} = \phi(h'_i, x_i)$ if $h'_i \neq y^*_i$
  with edge cost $cost_{h'_i} = 0$ if $h'_i = y^*_i$
find a minimum spanning tree(or forest) M of G
for each $h'_i \in \mathbf{h}'$
  if $h'_i = y^*_i$
    $y'_i \leftarrow h^*_i$
  else if $h'_i \in M$
    $y'_i \leftarrow 1$
  else $y'_i \leftarrow 0$
end for
return $\mathbf{y}'$

Figure 6.1: Algorithm to find the highest confidence labeling $\mathbf{y}'$ that can be clustered to the true labeling $\mathbf{y}^*$

[Global Model $P(\mathbf{y}|\mathbf{h})$] For the global model, we assume a deterministic clustering algorithm is given. In particular, we focus on single-link clustering, as it has been shown to be effective for coreference resolution (e.g. Ng and Cardie (2002)). With single-link clustering, $P(\mathbf{y}|\mathbf{h}) = 1$ if $\mathbf{h}$ can be clustered to $\mathbf{y}$, and $P(\mathbf{y}|\mathbf{h}) = 0$ if $\mathbf{h}$ cannot be clustered to $\mathbf{y}$.[5]

[Computation of the E-step] The E-step requires computation of the distribution of $P(\mathbf{h}|\mathbf{y}, \mathbf{x}, \theta^{(t-1)})$, which we will simply denote as $P(\mathbf{h}|\mathbf{y}, \mathbf{x})$, since all our distributions are implicitly conditioned on the model parameters θ.

$$P(\mathbf{h}|\mathbf{y}, \mathbf{x}) = \frac{P(\mathbf{h}, \mathbf{y}|\mathbf{x})}{P(\mathbf{y}|\mathbf{x})} \propto P(\mathbf{y}|\mathbf{h})\, P(\mathbf{h}|\mathbf{x})$$

[5]Single-link clustering simply takes the transitive closure, and does not consider the distance metric. In a pilot study, we also tried a variant of a stochastic clustering algorithm that takes into account the distance metric (set as the probabilities from the local model) for the global model, but the performance was worse.

ALGORITHM-2
INPUT: $\mathbf{x}$, true labeling $\mathbf{y}^*$, current local model $P(\mathbf{h}|\mathbf{x})$
GOAL: Find a high confidence labeling $\mathbf{y}'$ that is close to the true labeling $\mathbf{y}^*$
$\mathbf{h}^* \leftarrow \text{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{x})$
$\mathbf{h}' \leftarrow$ single-link-clustering($\mathbf{h}^*$)
for each $h'_i \in \mathbf{h}'$
  if $h'_i = y^*_i$
    $y'_i \leftarrow h^*_i$
  else $y'_i \leftarrow y^*_i$
end for
return $\mathbf{y}'$

Figure 6.2: Algorithm to find a high confidence labeling $\mathbf{y}'$ that is close to the true labeling $\mathbf{y}^*$

Notice that when computing $P(\mathbf{h}|\mathbf{y}, \mathbf{x})$, the denominator $P(\mathbf{y}|\mathbf{x})$ stays as a constant for different values of $\mathbf{h}$. The E-step requires enumeration of all possible values of $\mathbf{h}$, but it is intractable with our formulation, because inference for the global model $P(\mathbf{y}|\mathbf{h})$ does not factor out nicely. Therefore, we must resort to an approximation method. Neal and Hinton (1999) analyze and motivate various approximate EM training methods. One popular choice in practice is called "Viterbi training", a variant of the EM algorithm, which has been shown effective in many NLP applications. Viterbi training approximates the distribution by assigning all probability mass to a single best assignment. The algorithm for this is shown in Figure 6.1.

We propose another approximation option for the E-step that is given by Figure 6.2. Intuitively, when the current local model misses positive coreference decisions, the first algorithm constructs a $\mathbf{y}'$ that is closest to $\mathbf{h}'$ for single-link clustering to recover the true labeling $\mathbf{y}^*$, while the second algorithm constructs a $\mathbf{y}'$ that is closer to $\mathbf{y}^*$ by preserving all of the missing positive coreference decisions. [6]

[6]In a pilot study, we found that ALGORITHM-2 performs slightly better than ALGORITHM-1. We also

[Computation of M-step] Because $P(\mathbf{y}|\mathbf{h})$ is not parameterized, finding $\text{argmax}_\theta\ P(\mathbf{y}, \mathbf{h}|\mathbf{x})$ reduces to finding $\text{argmax}_\theta\ P(\mathbf{h}|\mathbf{x})$, which is standard CRF training. In order to speed up the training, we start convex optimization for CRFs using the parameter values $\theta^{(t-1)}$ from the previous M-step. For the very first iteration of EM, we start by setting $P(\mathbf{y}^*|\mathbf{x}) = 1$ for E-step, so that the first M-step will finds $\text{argmax}_\theta\ P(\mathbf{y}^*|\mathbf{x})$.

[Inference on the test data] It is intractable to marginalize out $\mathbf{h}$ from $P(\mathbf{y}, \mathbf{h}|\mathbf{x})$. Therefore, similar to the Viterbi-training in the E-step, we approximate the distribution of $\mathbf{h}$ by $\text{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{X})$.

6.3 Experiments – Effect of Structured Local Training

Data set: We evaluate our approach with two coreference data sets: MUC6 (muc, 1995) and MPQA[7](Wiebe et al., 2005). For the MUC6 data set, we extract noun phrases (mentions) automatically, but for MPQA, we assume mentions for coreference resolution are given as in Stoyanov and Cardie (2006). For MUC6, we use the standard training/test data split. For MPQA, we use 150 documents for training, and 50 documents for testing.

tried two other approximation options, but none performed as well as ALGORITHM-2. One of them removes the confusing sub-instances and has the effect of setting a uniform distribution on those sub-instances. The other computes the actual distribution on a subset of sub-instances. For brevity, we only present experimental results using ALGORITHM-2 in this work.

[7]Available at http://nrrc.mitre.org/NRRC/publications.htm.

Configuration: We follow Ng and Cardie (2002) for feature vector construction for each pair of mentions,[8] and Finley and Joachims (2005) for constructing a training/testing instance for each document: a training/testing instance consists of all pairs of mentions in a document. Then, a single pair of mentions is a *sub-instance*. We use the Mallet[9] implementation of CRFs, and set a Gaussian prior of 1.0 for all experiments. At each M-step, we train CRFs starting from the parameters from the previous M-step. We train CRFs up to 200 iterations, but because we start training CRFs from the previous parameters, the convergence from the second M-step becomes much faster. We apply up to 5 EM iterations, and choose best performing $\theta^{(t)}, 2 \leq t \leq 5$ based on the performance on the training data.[10]

Hypothesis: For the baseline (BASE) we employ the locally trained model for pairwise decisions without global inference. Clustering is applied only at test time, in order to make the assignment on the output variables coherent. We hypothesize that for the baseline, maximizing the likelihood for training will correlate more with the pairwise accuracy of the incoherent decisions before clustering than the pairwise accuracy of the coherent decisions after clustering. We also hypothesize that by performing structured local training (SLT), maximizing the likelihood will correlate more with the pairwise accuracy after clustering.

[8] In particular, our feature set corresponds to "All Features" in Ng and Cardie (2002), and we discretized numeric values.

[9] Available at http://mallet.cs.umass.edu.

[10] Selecting $\theta^{(t)}$ on a separate tuning data would be better, but the data for MUC6 in particular is very limited. Notice that we don't pick θ^1 when reporting the performance of SLT, because it is identical to the baseline.

Table 6.1: Performance of Structured Local Training: SLT reduces error rate (e %) after applying single-link clustering.

MUC6								
	after clustering				before clustering			
	e %	R %	P %	F %	e %	R %	P %	F %
BASE	1.50	**59.2**	56.2	**57.7**	1.18	38.0	85.6	52.6
SLT	**1.28**	49.8	**67.3**	57.2	1.35	26.4	84.3	40.2
MPQA								
	after clustering				before clustering			
	e %	R %	P %	F %	e %	R %	P %	F %
BASE	9.83	**75.8**	57.0	65.1	7.05	52.1	83.4	64.1
SLT	**6.39**	62.1	**80.6**	**70.2**	7.39	43.7	90.1	58.9

Results: Experimental results are shown in Table 6.1. We report error rate (error rate = 100 – accuracy) on the pairwise decisions (e %), and F1-score (F %) on the coreferent pairs.[11] For comparison, we show numbers from both after and before single-link clustering is applied. As hypothesized, the error rate of BASE increases after clustering, while the error rate of SLT decreases after clustering. Moreover, the error rate of SLT is considerably lower than that of BASE after clustering. However, the F1-score does not correlate with the error rate. That is, a lower error rate does not always lead to a higher F1-score, which motivates the *Biased Potential Functions* that we introduce in the next section. Notice that when we compare the precision/recall breakdown after clustering, SLT has higher precision and lower recall than BASE.

[11] Error rate and F1-score on the coreferent pairs are not ideal measures for the quality of clustering, however, we show them here in order to contrast the effect of SLT. We present MUC-scores for the same experimental settings in Table 6.3.

6.4 Biased Potential Functions

We introduce *biased potential functions* for training CRFs to empirically favor preferred evaluation measures for the learning task, such as F-score and MUC-score that have been considered hard for traditional likelihood-based methods to optimize for. Intuitively, biased potential functions emphasize those subcomponents of an instance that can be of greater importance than the rest of an instance.

6.4.1 Definitions

The conditional probability of $P(\mathbf{y}|\mathbf{x})$[12] for CRFs is given by Lafferty et al. (2001)

$$P(\mathbf{y}|\mathbf{x}) = \frac{\prod_i \phi(C_i, \mathbf{x})}{\sum_{\mathbf{y}} \prod_i \phi(C_i, \mathbf{x})}$$

where $\phi(C_i, \mathbf{x})$ is a potential function defined over each clique C_i. Potential functions are typically parameterized in an exponential form as follows.

$$\phi(C_i, \mathbf{x}) = \exp \sum_k \lambda_k f_k(C_i, \mathbf{x})$$

where λ_k are the parameters and $f_k(\cdot)$ are feature indicator functions. Because the Hammersley-Clifford theorem (Hammersley and Clifford, 1971) for undirected graphical models holds for any non-negative potential functions, we propose alternative potential functions as follows.

$$\psi(C_i, \mathbf{x}) = \begin{cases} \beta\phi(C_i, \mathbf{x}) & \text{if } \mu(C_i, \mathbf{x}) = \mathit{true} \\ \phi(C_i, \mathbf{x}) & \text{otherwise} \end{cases}$$

[12]For the local model described in Section 2, $\mathbf{y}$ should be replaced with $\mathbf{h}$. We use $\mathbf{y}$ in this section however, as it is a more conventional notation in general.

where β is a non-negative bias factor, and $\mu(C_i, \mathbf{x})$ is a predicate (or an indicator function) to check certain properties on $(C_i, \mathbf{x})$.[13] Examples of possible $\mu(\cdot)$ would be whether the true assignment for C_i in the training data contains certain class values, or whether the current observation indexed by C_i has particular characteristics. More specific details will be given in §4.2.

Training and testing with biased potential functions is mostly identical to the traditional log-linear formulations by $\phi(\cdot)$ as defined above, except for small and straightforward modifications to the computation of the likelihood and the derivative of the likelihood.

The key idea for biased potential functions is nothing new, as it is conceptually similar to instance weighting for problems with non-structured output (e.g. Aha and Goldstone (1992), Cardie and Nowe (1997)). However, biased potential functions differ technically in that they emphasize desired subcomponents without altering the i.i.d. assumption, and still weight each instance alike. Despite the conceptual simplicity, we are not aware of any previous work that explored biased potential functions for problems with structured output.

6.4.2 Applications to Coreference Resolution

[Bias on Coreferent Pairs] For coreference resolution, pairs that are coreferent are in a minority class[14], and biased potential functions can mitigate this skewed data problem, by amplifying the clique potentials that correspond to coreferent

[13] In our problem formulation, cliques are individual nodes, and potential functions are defined over the observations indexed by the current i only: i.e. $\phi(C_i, \mathbf{x}) = \phi(y_i, x_i)$, $\mu(C_i, \mathbf{x}) = \mu(y_i, x_i)$ and $\psi(C_i, \mathbf{x}) = \psi(y_i, x_i)$.

[14] Only 1.72% of the pairs are coreferent in the MUC6 data, and about 12% are coreferent in the MPQA data.

pairs. We define $\mu(y_i, x_i)$ to be true if and only if the true assignment for y_i in the training data is *'coreferent'*. Notice that $\mu(\cdot)$ does not depend on what particular value y_i might take, but only depends on the true value of y_i in the training data. For testing, $\mu(y_i, x_i)$ will be always false.[15]

[Bias on Closer Coreferent Pairs] For coreference resolution, we hypothesize that coreferent pairs for closer mentions have more significance, because they tend to have clearer linguistic clues to determine coreference. We further hypothesize that by emphasizing only close coreferent pairs, we can have our model favor the MUC score. For this, we define $\mu(y_i, x_i)$ to be true if and only if x_i is for a pair of mentions that are the closest coreferent pair.

6.5 Experiments – Effect of Biased Potential Functions

Data sets and configurations for experiments are identical to those used in §3.

Hypothesis: We hypothesize that using biased potential functions, maximizing the likelihood for training can correlate better with F1-score or MUC-score than the pairwise accuracy. In particular, we hypothesize that biasing on every coreferent pair will correlate more with F1-score, and biasing on close coreferent pairs will correlate more with MUC-score. In general, we expect that biasing on coreferent pairs will boost recall, potentially decreasing precision.

[15] Notice that $\mu(y_i, x_i)$ changes the surface of the likelihood for training, but does not affect the inference of finding the argmax in our local model. That is, $\text{argmax}_{y_i} \phi(y_i, x_i) = \text{argmax}_{y_i} \psi(y_i, x_i)$ (with y_i replaced with h_i).

Table 6.2: Performance of Biased Potential Functions: pairwise scores are taken before single-link-clustering is applied.

MUC6							
	pairwise				MUC		
	e %	R %	P %	F %	R %	P %	F %
BASE	1.18	38.0	**85.6**	52.6	59.0	**75.8**	66.4
BASIC-P$1^{1.5}$	1.20	38.9	82.1	52.8	64.2	71.8	**67.8**
BASIC-P$1^{3.0}$	1.32	46.9	71.3	56.6	68.9	64.3	66.5
BASIC-P$a^{1.5}$	**1.15**	44.2	79.9	56.9	62.1	68.7	65.2
BASIC-P$a^{3.0}$	1.44	**52.5**	62.9	**57.2**	**70.9**	60.5	65.3
MPQA							
	pairwise				MUC		
	e %	R %	P %	F %	R %	P %	F %
BASE	**7.05**	52.1	**83.4**	64.1	75.6	**81.5**	**78.4**
BASIC-P$1^{1.5}$	7.18	54.6	79.6	64.8	77.7	76.5	77.1
BASIC-P$1^{3.0}$	7.22	59.9	75.4	66.8	83.3	71.7	77.1
BASIC-P$a^{1.5}$	7.65	59.7	72.2	65.4	79.8	73.2	76.4
BASIC-P$a^{3.0}$	8.22	**69.2**	65.1	**67.1**	**85.8**	67.8	75.7

Results [BPF]: Experimental results for biased potential functions, without structured local training, are shown in Table 6.2. BASIC-P1^{β} denotes local training with biased potential on the closest coreferent pairs with bias factor β, and BASIC-Pa^{β} denotes local training with biased potential on the all coreferent pairs with bias factor β, where β = 1.5 or 3.0. For brevity, we only show pairwise numbers before applying single-link-clustering.[16] As hypothesized, biased potential

[16]This is because we showed in §3 that basic local training does not correlate well with pairwise scores after clustering, and in order to see the direct effect of biased potential functions, we examine pairwise numbers before clustering.

Table 6.3: Performance of Biased Potential Functions with Structured Local Training: All numbers are taken after single-link clustering.

MUC6							
	pairwise				MUC		
	e %	R %	P %	F %	R %	P %	F %
BASE	1.50	59.2	56.2	57.7	59.0	75.8	66.4
SLT	1.28	49.8	67.3	57.2	56.3	**77.8**	65.3
SLT-P$1^{1.5}$	**1.19**	52.8	**70.6**	60.4	59.3	74.6	66.1
SLT-P$1^{3.0}$	1.42	63.5	57.9	**60.6**	67.5	70.7	69.1
SLT-P$a^{1.5}$	1.43	58.6	58.5	58.5*	64.0	73.6	68.5
SLT-P$a^{3.0}$	1.71	**65.2**	50.3	56.8	**70.5**	69.3	**69.9***
MPQA							
	pairwise				MUC		
	e %	R %	P %	F %	R %	P %	F %
BASE	9.83	75.8	57.0	65.1	75.6	81.5	78.4
SLT	**6.39**	62.1	**80.6**	70.2	69.1	**88.2**	77.5
SLT-P$1^{1.5}$	6.54	64.9	77.4	**70.6***	72.2	84.5	77.9*
SLT-P$1^{3.0}$	9.09	77.2	59.6	67.3	78.4	79.5	78.9
SLT-P$a^{1.5}$	6.74	65.2	75.7	70.1	72.4	87.2	**79.1**
SLT-P$a^{3.0}$	14.71	**78.2**	43.9	56.2	**80.5**	73.8	77.0

functions in general boost recall at the cost of precision. Also, for a fixed value of β, BASIC-P1^{β} gives better MUC-F1 than BASIC-Pa^{β}, and BASIC-Pa^{β} gives better pairwise-F1 than BASIC-P1^{β} for both data sets.

Results [SLT+BPF]: Experimental results that combine SLT and BPF are shown in Table 6.3. Similarly as before, SLT-Px^{β} denotes SLT with biased po-

tential scheme Px, with bias factor β. For brevity, we only show numbers after applying single-link-clustering. Unlike the results shown in Table 6.2, for a fixed value of β, SLT-P1$^{\beta}$ correlates better with pairwise-F1, and SLT-Pa^{β} correlates better with MUC-F1. This indicates that when biased potential functions are used in conjunction with SLT, the effect of biased potential functions can be different from the case without SLT. Comparing F1-scores in Table 6.2 and Table 6.3, we see that the combination of biased potential functions with SLT improves performance in general. In particular, SLT-P1$^{3.0}$ and SLT-P$a^{1.5}$ consistently improve performance over BASE on both data sets, for both pairwise-F1 and MUC-F1. We present performance scores for all variations of configurations for reference, but we also mark the particular configuration SLT-Px^{β} (by '*' on F1-scores) that is chosen when selecting the configuration based on the performance on the training data for each performance measure. To conclude, structured local training with biased potential functions bring a substantial improvement for MUC-F1 score, from 66.4% to 69.9% for MUC6 data set. For pairwise-F1, the performance increase from 57.7% to 58.5% for MUC6, and from 65.1% to 70.6% for MPQA.[17]

6.6 Related Work

Structured local training is motivated by recent research that has shown that reducing the discrepancy between the training model and testing model can improve the performance without incurring the heavy computational overhead of full-blown global inference-based training. [18] (e.g. Cohen and Carvalho (2005),

[17] Performance on the MPQA data for MUC-F1 is slightly decreased from 78.4% to 77.9%. Note the MUC scores for the MPQA baseline are already quite high to begin with.

[18] The computational cost for SLT in our experiments were about twice of the cost for the local training of the baseline. This is the case because M-step converges very fast from the second EM iteration, by initializing CRFs using parameters from the previous M-step. Biased potential functions hardly adds extra

Sutton and McCallum (2005a), Sutton and McCallum (2005b)). Our work differs in that (1) we use hidden variables to capture the interactions between local inference and global inference, (2) we present an application to coreference resolution, while previous work has shown applications for variants of sequence tagging. McCallum and Wellner (2004) showed a global training approach with CRFs for coreference resolution, but they used the voted perceptron algorithm for training, which no longer maximizes the likelihood. In addition, they assume that all and only those noun phrases involved in coreference resolution are given.

The performance of our system on MUC6 data set is comparable to previously reported systems. Using the same feature set, Ng and Cardie (2002) reports 64.5% of MUC-score, while our system achieved 69.9%. Ng and Cardie (2002) reports 70.4% of MUC-score using hand-selected features. With an additional feature selection or feature induction step, the performance of our system might further improve. McCallum and Wellner (2004) reports 73.42% of MUC-score on MUC6 data set, but their experiments assumed perfect identification of all and only those noun phrases involved in a coreference relation, thus substantially simplifying the task.

6.7 Summary of Chapter

We present a novel training procedure, *structured local training*, that maximizes likelihood while exploiting the benefits of global inference during training. This is achieved by incorporating hidden variables to capture the interactions be-

computational cost. In practice, BPFs reduce training time substantially: we observed that the higher the bias is, the quicker CRFs converge.

tween local inference and global inference. In addition, we introduce *biased potential functions* that allow CRFs to empirically favor performance measures such as F1-score or MUC-score. We focused on the application of coreference resolution in this work, but the key ideas of our approaches can be extended to other applications, and other machine learning techniques motivated by Markov networks.

CHAPTER 7

CONCLUSIONS

7.1 Summary of Contributions

In this dissertation, we explored statistical and computational approaches to fine-grained opinion analysis. In particular, we tackled the task of extracting fine-grained opinion elements, extracting fine-grained opinion expressions together with their attributes, determining the polarity of the fine-grained opinion expressions in light of compositional semantics, adapting a general-purpose polarity lexicon into a domain specific one, and resolving the coreferent entities. A common ground in our approaches is that we recognized and exploited task-specific linguistic structure into the learning and/or inference procedures.

7.2 Future Research Direction

The study performed in this dissertation proves the viability of fine-grained opinion analysis, and opens new challenges for future research, some of which are discussed in what follows.

7.2.1 Summarizing Opinions at Web-scale

Opinion-laden text, such as is found in blogs and forums, is ever more prevalent than before. Consequently, there are growing practical needs for statistical methods to automatically generate summaries of prevailing opinions across

multiple documents, toward a given topic. Some researchers have explored opinion-oriented summarization (e.g., Titov and McDonald (2008), Lerman and McDonald (2009), Cheung et al. (2009)), however most of previous work has focused on the domain of product reviews, and handled a manageable size of corpus. There has been ample research on extractive summarization techniques (e.g., Carbonell and Goldstein (1998), Nenkova et al. (2006), Erkan and Radev (2004)) for summarizing general information without concerning opinions. One of the recent techniques is based on efficient approximation algorithms exploiting submodularity (e.g., Lin et al. (2009), Lin and Bilmes (2010)), which could be particularly suitable when handling a very large amount of data.

7.2.2 Objective Subjectivity

Much research for affect and opinion to date has focused on the explicit use of subjective language (e.g., Wilson et al. (2005)). That is, subjectivity has been recognized via lexical cues that are strongly indicative of emotions or opinions, such as "good", "enthusiastic" or "perplexed". However, the way humans express affect and opinion is not limited to the explicit use of subjective lexical items. More often than not, opinions are expressed via seemingly objective statements that support one's affective state of mind. For example, when people debate whether global warming is actually occurring or not, it is a collection of factual statements that builds an actual opinion. That is, a particular selection of objective statements can be already an act of expressing an opinion. But can we teach computers to recognize this type of *subjectivity in disguise as objectivity*? This might be possible, due to the sheer volume of user-created opinionated text available today. Not only it is an interesting research problem in its own right,

it is also a problem with practical impact; *subjectivity in disguise as objectivity* is particularly prevalent in political blogs and forums. And a system that can recognize objective statements that form an opinion toward a given topic would be much more useful than a system that simply judges the polarity of an opinion appearing in text.

7.2.3 Compositional Rule Induction for Polarity Inference

Compositionality explains a good deal of semantic interpretation of text (Montague, 1974), however, the challenge is how to design a computational model that mimics the way human infers the semantic meaning of the text. The work presented in this dissertation in Chapter 2 exploits the idea of compositionality for fine-grained polarity inference and demonstrates positive results in empirical study. However, the limitation is that only a handful of hand-coded compositional rules were employed, which is a crude subset of what would be used by human. A next step departing from this work would be inducting the rules automatically from the raw text.

7.2.4 Affect Beyond Opinion

The research pursued in this dissertation has focused on making a rather coarse categorization of affect and opinion - positive/neutral/negative. However, human affect is much more intricate than that. Therefore, a desired next step would be to investigate computational models that can interpret more sophisticated types of affect appearing in text (e.g., Strapparava and Mihalcea (2007)).

For instance, can we teach computers to detect deception in text (e.g., Mihalcea and Strapparava (2009))? Such research can benefit much from other disciplines, such as psycholinguistics, psychology, and cognitive science, in developing computational models that look for evidence of deception. Conversely, computational models designed by computer scientists can help discovering new insights that have not been previously recognized in other disciplines.

BIBLIOGRAPHY

[Abney1996] Steven Abney. 1996. Partial parsing via finite-state cascades. *Nat. Lang. Eng.*, 2(4):337–344.

[Adamic and Glance2005] Lada A. Adamic and Natalie Glance. 2005. The political blogosphere and the 2004 U.S. election: Divided they blog. In *Proceedings of LinkKDD*.

[Adreevskaia and Bergler2006] Alina Adreevskaia and Sabine Bergler. 2006. Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 209–216.

[Aha and Goldstone1992] David W. Aha and Robert L. Goldstone. 1992. Concept learning and flexible weighting. In *In Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pages 534–539. Erlbaum.

[Andreevskaia and Bergler2008] Alina Andreevskaia and Sabine Bergler. 2008. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *Proceedings of ACL-08: HLT*, pages 290–298, Columbus, Ohio.

[Banea et al.2008] Carmen Banea, Rada Mihalcea, and Janyce Wiebe. 2008. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In European Language Resources Association (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may.

[Bautin et al.2008] Mikhail Bautin, Lohit Vijayarenu, and Steven Skiena. 2008. International sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

[Berger et al.1996] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics 22-1*, pages 39–71.

[Bethard et al.2004] Steven Bethard, Hong, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. 2004. Automatic extraction of opinion propositions and their holders. In *In 2004 AAAI Spring Symposium on Exploring Attitude and Affect in Text*, pages 22–24.

[Blitzer et al.2007] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association for Computational Linguistics*, Prague, Czech Republic.

[Breck et al.2007] Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2683–2688, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Bunescu et al.2004] R. Bunescu, Razvan Bunescu, and R. J. Mooney. 2004. Collective information extraction with relational markov networks.

[Cai and Hofmann2004] Lijuan Cai and Thomas Hofmann. 2004. Hierarchical document categorization with support vector machines. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87, New York, NY, USA. ACM.

[Carbonell and Goldstein1998] Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, New York, NY, USA. ACM.

[Cardie and Nowe1997] Claire Cardie and Nicholas Nowe. 1997. Improving minority class prediction using case-specific feature weights. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 57–65, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Cardie et al.2003] Claire Cardie, Janyce Wiebe, Theresa Wilson, and Diane Litman. 2003. Combining low-level and summary representations of opinions for multi-perspective question answering. In *Proceedings of the AAAI Spring Symposium on New Directions in Question Answering*, pages 20–27.

[Cardie et al.2004] Claire Cardie, Janyce Wiebe, Theresa Wilson, and Diane J. Litman. 2004. Low-level annotations and summary representations of opinions for multiperspective qa. In *New Directions in Question Answering*, pages 87–98.

[Chesley et al.2006] Paula Chesley, Bruce Vincent, Li Xu, and Rohini Srihari. 2006. Using verbs and adjectives to automatically classify blog sentiment. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 27–29.

[Cheung et al.2009] Jackie Chi Kit Cheung, Giuseppe Carenini, and Raymond T. Ng. 2009. Optimization-based content selection for opinion summarization. In *UCNLG+Sum '09: Proceedings of the 2009 Workshop on Language Generation and Summarisation*, pages 7–14, Morristown, NJ, USA. Association for Computational Linguistics.

[Choi and Cardie2007] Yejin Choi and Claire Cardie. 2007. Structured local training and biased potential functions for conditional random fields with application to coreference resolution. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 65–72, Rochester, New York, April. Association for Computational Linguistics.

[Choi and Cardie2008] Yejin Choi and Claire Cardie. 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, October. Association for Computational Linguistics.

[Choi and Cardie2009] Yejin Choi and Claire Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 590–598, Singapore, August. Association for Computational Linguistics.

[Choi and Cardie2010] Yejin Choi and Claire Cardie. 2010. Hierarchical sequential learning for extracting opinions and their attributes. In *Association for Computational Linguistics (ACL)*.

[Choi et al.2005] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 355–362, Morristown, NJ, USA. Association for Computational Linguistics.

[Choi et al.2006] Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Morristown, NJ, USA. Association for Computational Linguistics.

[Chomsky1965] Noam Chomsky. 1965. Aspects of the theory of syntax.

[Claire Cardie and Litman.2004] Theresa Wilson Claire Cardie, Janyce Wiebe

and Diane Litman. 2004. Low-level annotations and summary representations of opinions for multiperspective qa. In *New Directions in Question Answering. AAAI Press/MIT Press.*

[Cohen and Carvalho2005] William W. Cohen and Vitor R. Carvalho. 2005. Stacked sequential learning. In *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence*, pages 671–676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Collins2002] Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

[Conrad and Schilder2007] Jack G. Conrad and Frank Schilder. 2007. Opinion mining in legal blogs. In *Proceedings of the International Conference on Artificial Intelligence and Law (ICAIL)*, pages 231–236, New York, NY, USA. ACM.

[Crammer and Singer2003] Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991.

[Cunningham et al.2002] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA*.

[Das and Chen.2001] Sanjiv Das and Mike Chen. 2001. Yahoo for amazon: Extracting market sentiment from stock message boards. In *Proceedings of the 8th Asia Pacific Finance Association Annual Conference.*

[Dave et al.2003] Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 519–528, New York, NY, USA. ACM.

[Dempster et al.1977] A. Dempster, N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Statistical Society, Series B*, 39(1):1–38.

[Denis and Baldridge2007] Pascal Denis and Jason Baldridge. 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.

[Deschacht and Moens2006] Koen Deschacht and Marie-Francine Moens. 2006. Efficient hierarchical entity classifier using conditional random fields. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 33–40, Sydney, Australia, July. Association for Computational Linguistics.

[Dowty et al.1981] David R. Dowty, Robert E. Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*. D. Reidel.

[Eguchi and Shah2006] Koji Eguchi and Chirag Shah. 2006. Opinion retrieval experiments using generative models: Experiments for the TREC 2006 blog track. In *Proceedings of TREC*.

[Erkan and Radev2004] Günes Erkan and Dragomir R. Radev. 2004. Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

[Esuli and Sebastiani2006] Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, pages 417–422.

[et al.1996] Joseph Pickett et al. 1996. The american heritage book of english usage: A practical and authoritative guide to contemporary english.

[Fine et al.1998] Shai Fine, Yoram Singer, and Naftali Tishby. 1998. The hierarchical hidden markov model: Analysis and applications. In *Machine Learning, vol. 32, p. 41-62*.

[Finkel et al.2005] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Morristown, NJ, USA. Association for Computational Linguistics.

[Finkel et al.2006] Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. 2006. Solving the problem of cascading errors: approximate bayesian inference for linguistic annotation pipelines. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 618–626, Morristown, NJ, USA. Association for Computational Linguistics.

[Finley and Joachims2005] Thomas Finley and Thorsten Joachims. 2005. Supervised clustering with support vector machines. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 217–224, New York, NY, USA. ACM.

[Fukuhara et al.2007] Tomohiro Fukuhara, Hiroshi Nakagawa, and Toyoaki Nishida. 2007. Understanding sentiment of people from news articles: Temporal sentiment analysis of social events. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

[Greene and Resnik2009] Stephan Greene and Philip Resnik. 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, June. Association for Computational Linguistics.

[GuoDong et al.2006] Zhou GuoDong, Su Jian, and Zhang Min. 2006. Modeling commonality among related classes in relation extraction. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 121–128, Morristown, NJ, USA. Association for Computational Linguistics.

[Hammersley and Clifford1971] John Hammersley and Peter Clifford. 1971. Markov fields on finite graphs and lattices.

[Hu and Liu2004a] Minqing Hu and Bing Liu. 2004a. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA. ACM.

[Hu and Liu2004b] Minqing Hu and Bing Liu. 2004b. Mining opinion features in customer reviews. In *Proceedings of AAAI*, pages 755–760.

[Kaji and Kitsuregawa2007] Nobuhiro Kaji and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of the Joint Conference on Empirical Methods in Natu-*

ral Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 1075–1083.

[Kanayama and Nasukawa2006] Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–363. Association for Computational Linguistics.

[Kennedy and Inkpen2005] Alistair Kennedy and Diana Inkpen. 2005. Sentiment classification of movie and product reviews using contextual valence shifters. In *Proceedings of FINEXIN 2005, Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations*, pages 110–125.

[Kim and Hovy2004] Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1367, Morristown, NJ, USA. Association for Computational Linguistics.

[Kim and Hovy2005a] Soo-Min Kim and Eduard Hovy. 2005a. Automatic detection of opinion bearing words and sentences. In *Companion Volume to the Proceedings of IJCNLP-05, the Second International Joint Conference on Natural Language Processing*, pages 61–66, Jeju Island, KR.

[Kim and Hovy2005b] Soo-Min Kim and Eduard Hovy. 2005b. Identifying opinion holders for question answering in opinion texts. In *Proceedings of the AAAI Workshop on Question Answering in Restricted Domains*.

[Kim and Hovy2006] Soo-Min Kim and Eduard Hovy. 2006. Identifying and analyzing judgment opinions. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 200–207, Morristown, NJ, USA. Association for Computational Linguistics.

[Koomen et al.2005] Peter Koomen, Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2005. Generalized inference with multiple semantic role labeling systems. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 181–184, Ann Arbor, Michigan, June. Association for Computational Linguistics.

[Ku et al.2006] Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI*

Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW), pages 100–107.

[Lafferty et al.2001] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Lerman and McDonald2009] Kevin Lerman and Ryan McDonald. 2009. Contrastive summarization: an experiment with consumer reviews. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 113–116, Morristown, NJ, USA. Association for Computational Linguistics.

[Liang et al.2008] Percy Liang, Hal Daumé, III, and Dan Klein. 2008. Structure compilation: trading structure for features. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 592–599, New York, NY, USA. ACM.

[Lin and Bilmes2010] Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *North American chapter of the Association for Computational Linguistics/Human Language Technology Conference (NAACL/HLT-2010)*, Los Angeles, CA, June.

[Lin et al.2009] Hui Lin, Jeff Bilmes, and Shasha Xie. 2009. Graph-based submodular selection for extractive summarization. In *Proc. IEEE Automatic Speech Recognition and Understanding (ASRU)*, Merano, Italy, December.

[Liu et al.2005] Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the web. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 342–351, New York, NY, USA. ACM.

[Liu et al.2007] Yang Liu, Jimmy Huang, Aijun An, and Xiaohui Yu. 2007. ARSA: A sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*.

[Mao and Lebanon2007] Yi Mao and Guy Lebanon. 2007. Isotonic conditional random fields and local sentiment flow. In *Advances in Neural Information Processing Systems*.

[McCallum and Wellner2004] Andrew McCallum and Ben Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Neural Information Processing Systems (NIPS).*

[McCallum2002] Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

[Mcdonald et al.2007] Ryan Mcdonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

[Mihalcea and Strapparava2009] Rada Mihalcea and Carlo Strapparava. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 309–312, Suntec, Singapore, August. Association for Computational Linguistics.

[Miller1995] George A. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41.

[Moilanen and Stephen2007] Karo Moilanen and Pulman Stephen. 2007. Sentiment composition. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007)*, pages 378–382, September 27-29.

[Montague1974] Richard Montague. 1974. *Formal Philosophy: Selected Papers of Richard Montaguetextup, ed. Richmond Thomason*. Yale University Press, New Haven.

[Mooney and Bunescu2005] Raymond J. Mooney and Razvan Bunescu. 2005. Mining knowledge from text using information extraction. *SIGKDD Explor. Newsl.*, 7(1):3–10.

[Morinaga et al.2002] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. 2002. Mining product reputations on the web. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 341–349, New York, NY, USA. ACM.

[muc1995] 1995. Muc-6. In *In Proc. of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufmann.

[Munson et al.2005] Art Munson, Claire Cardie, and Rich Caruana. 2005. Op-

timizing to arbitrary nlp metrics using ensemble selection. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 539–546, Morristown, NJ, USA. Association for Computational Linguistics.

[Nairn et al.2006] Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *In Proceedings of ICoS-5 (Inference in Computational Semantics*.

[Neal and Hinton1999] Radford M. Neal and Geoffrey E. Hinton. 1999. A view of the em algorithm that justifies incremental, sparse, and other variants. pages 355–368.

[Nenkova et al.2006] Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580, New York, NY, USA. ACM.

[Ng and Cardie2002] Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 104–111, Morristown, NJ, USA. Association for Computational Linguistics.

[Niu et al.2005] Yun Niu, Xiaodan Zhu, Jianhua Li, and Graeme Hirst. 2005. Analysis of polarity information in medical text. In *In: Proceedings of the American Medical Informatics Association 2005 Annual Symposium*, pages 570–574.

[Pang and Lee2004] Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 271, Morristown, NJ, USA. Association for Computational Linguistics.

[Pang and Lee2005] Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 115–124.

[Pang and Lee2008] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135.

[Pang et al.2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA. Association for Computational Linguistics.

[Pinker2007] Steven Pinker. 2007. *The Stuff of Thought: Language as a Window into Human Nature*. Viking Adult.

[Polanyi and Zaenen2004] Livia Polanyi and Annie Zaenen. 2004. Contextual lexical valence shifters. In Yan Qu, James Shanahan, and Janyce Wiebe, editors, *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*. AAAI Press. AAAI technical report SS-04-07.

[Popescu and Etzioni2005a] Ana-Maria Popescu and Oren Etzioni. 2005a. Extracting product features and opinions from reviews. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.

[Popescu and Etzioni2005b] Ana-Maria Popescu and Oren Etzioni. 2005b. Extracting product features and opinions from reviews. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 339–346, Morristown, NJ, USA. Association for Computational Linguistics.

[Prager et al.2000] John Prager, Eric Brown, Anni Coden, and Dragomir Radev. 2000. Question-answering by predictive annotation. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 184–191, New York, NY, USA. ACM.

[Punyakanok et al.2004] Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic role labeling via integer linear programming inference. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1346, Morristown, NJ, USA. Association for Computational Linguistics.

[Punyakanok et al.2005] Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dav Zimak. 2005. Learning and inference over constrained output. In *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence*, pages 1124–1129, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Rao and Ravichandran2009] Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *EACL '09: Proceedings of the*

12th Conference of the European Chapter of the Association for Computational Linguistics, pages 675–682, Morristown, NJ, USA. Association for Computational Linguistics.

[Riloff and Wiebe2003] Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112, Morristown, NJ, USA. Association for Computational Linguistics.

[Roth and tau Yih2004] Dan Roth and Wen tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 1–8, Boston, Massachusetts, USA, May 6 - May 7. Association for Computational Linguistics.

[Roth and Yih2002] Dan Roth and Wen-tau Yih. 2002. Probabilistic reasoning for entity & relation recognition. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

[Roth and Yih2005] Dan Roth and Wen-tau Yih. 2005. Integer linear programming inference for conditional random fields. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 736–743, New York, NY, USA. ACM.

[Shaikh et al.2007] Mostafa Al Shaikh, Helmut Prendinger, and Ishizuka Mitsuru. 2007. Assessing sentiment of text by semantic dependency and contextual valence analysis. In *ACII '07: Proceedings of the 2nd international conference on Affective Computing and Intelligent Interaction*, pages 191–202, Berlin, Heidelberg. Springer-Verlag.

[Skounakis et al.2003] Marios Skounakis, Mark Craven, and Soumya Ray. 2003. Hierarchical hidden markov models for information extraction. In *IJCAI'03: Proceedings of the 18th international joint conference on Artificial intelligence*, pages 427–433, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Snyder and Barzilay2007] Benjamin Snyder and Regina Barzilay. 2007. Multiple aspect ranking using the Good Grief algorithm. In *Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL)*, pages 300–307.

[Somasundaran et al.2007a] Swapna Somasundaran, Josef Ruppenhofer, and

Janyce Wiebe. 2007a. Detecting arguing and sentiment in meetings. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*.

[Somasundaran et al.2007b] Swapna Somasundaran, Theresa Wilson, Janyce Wiebe, and Veselin Stoyanov. 2007b. QA with attitude: Exploiting opinion type analysis for improving question answering in on-line discussions and the news. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

[Stepinski and Mittal2007] Adam Stepinski and Vibhu Mittal. 2007. A fact/opinion classifier for news articles. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 807–808, New York, NY, USA. ACM Press.

[Stoyanov and Cardie2006] Veselin Stoyanov and Claire Cardie. 2006. Partially supervised coreference resolution for opinion summarization through structured rule learning. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 336–344, Morristown, NJ, USA. Association for Computational Linguistics.

[Stoyanov and Cardie2008a] Veselin Stoyanov and Claire Cardie. 2008a. Annotating topics of opinions. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

[Stoyanov and Cardie2008b] Veselin Stoyanov and Claire Cardie. 2008b. Topic identification for fine-grained opinion analysis. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 817–824, Morristown, NJ, USA. Association for Computational Linguistics.

[Stoyanov et al.2005] Veselin Stoyanov, Claire Cardie, and Janyce Wiebe. 2005. Multi-perspective question answering using the opqa corpus. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 923–930, Morristown, NJ, USA. Association for Computational Linguistics.

[Strapparava and Mihalcea2007] Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: affective text. In *SemEval '07: Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74, Morristown, NJ, USA. Association for Computational Linguistics.

[Sutton and McCallum2005a] Charles Sutton and Andrew McCallum. 2005a. Fast, piecewise training for discriminative finite-state and parsing models.

[Sutton and McCallum2005b] Charles Sutton and Andrew McCallum. 2005b. Piecewise training of undirected models. In *In Proc. of UAI*.

[Sutton et al.2007] Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *J. Mach. Learn. Res.*, 8:693–723.

[Takamura et al.2005] Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of ACL-05, 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, US. Association for Computational Linguistics.

[Tateishi et al.2001] Kenji Tateishi, Yoshihide Ishiguro, and Toshikazu Fukushima. 2001. Opinion information retrieval from the Internet. *Information Processing Society of Japan (IPSJ) SIG Notes*, 2001(69(20010716)):75–82. Also cited as "A reputation search engine that gathers people's opinions from the Internet", IPSJ Technical Report NL-14411. In Japanese.

[Thomas et al.2006] Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 327–335.

[Titov and McDonald2008] Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, Columbus, Ohio, June. Association for Computational Linguistics.

[Tong2001] Richard M. Tong. 2001. An operational system for detecting and tracking opinions in on-line discussion. In *Proceedings of the Workshop on Operational Text Classification (OTC)*.

[Turney2002] Peter D. Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Morristown, NJ, USA. Association for Computational Linguistics.

[Wellner et al.2004] Ben Wellner, Andrew McCallum, Fuchun Peng, and Michael Hay. 2004. An integrated, conditional model of information extraction and coreference with application to citation matching. In *UAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 593–601, Arlington, Virginia, United States. AUAI Press.

[White et al.2001] Michael White, Tanya Korelsky, Claire Cardie, Vincent Ng, David Pierce, and Kiri Wagstaff. 2001. Multidocument summarization via information extraction. In *HLT '01: Proceedings of the first international conference on Human language technology research*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

[Wiebe et al.2002] Janyce Wiebe, Eric Breck, Chris Buckley, Claire Cardie, Paul Davis, Bruce Fraser, Diane Litman, David Pierce, Ellen Riloff, and Theresa Wilson. 2002. NRRC Summer Workshop on Multiple-Perspective Question Answering Final Report. Tech report, Northeast Regional Research Center, Bedford, MA.

[Wiebe et al.2005] Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation (formerly Computers and the Humanities)*, 39(2/3):164–210.

[Wilson et al.2005] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, Morristown, NJ, USA. Association for Computational Linguistics.

[Wilson et al.2006] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. 2006. Recognizing strong and weak opinion clauses. *Computational Intelligence*, 22(2):73–99.

[Wilson et al.2009] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Comput. Linguist.*, 35(3):399–433.

[Yu and Hatzivassiloglou2003] Hong Yu and Vasileios Hatzivassiloglou. 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, Morristown, NJ, USA. Association for Computational Linguistics.

[Zhao et al.2008] Jun Zhao, Kang Liu, and Gen Wang. 2008. Adding redundant features for CRFs-based sentence sentiment classification. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 117–126, Honolulu, Hawaii, October. Association for Computational Linguistics.

[Zhou and Hovy2006] Liang Zhou and Eduard Hovy. 2006. On the summarization of dynamically introduced information: Online discussions and blogs. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 237–242.