ENTITY/EVENT-LEVEL SENTIMENT DETECTION AND INFERENCE

by

Lingjia Deng

Bachelor of Engineering

Beijing University of Posts and Telecommunications

2011

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of

Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2015

UNIVERSITY OF PITTSBURGH

INTELLIGENT SYSTEMS PROGRAM

This proposal was presented

by

Lingjia Deng

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

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

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

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

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

ENTITY/EVENT-LEVEL SENTIMENT DETECTION AND INFERENCE

Lingjia Deng, PhD

University of Pittsburgh, 2015

Most of the work in sentiment analysis and opinion mining focuses on extracting explicit sentiments, the sentiments revealed by certain sentiment expressions. However, opinions may be expressed implicitly via inference rules over explicit sentiments. The implicit opinions are indicated in the text, and they are important for a sentiment analysis system to fully understand the opinions in the text. In this thesis, we incorporate several inference rules as constraints in joint prediction models, to develop an entity/event-level sentiment analysis system which aims at detecting both explicit and implicit sentiments expressed from an entity toward an entity or event in the text. The entity/event-level sentiment analysis is a more fine-grained and more difficult task, compared to state-of-the-art sentiment analysis work which mostly are span based. In this work, we propose to work in three aspects: 1) developing an entity/event-level sentiment corpus, MPQA 3.0, where both explicit and implicit sentiments are annotated; 2) developing an annotated corpus and two computational models focusing on inferring sentiments about a particular type of events: +/-effect event, which have positive effect or negative effect on the theme; 3) developing joint prediction models to improve detecting and inferring sentiments expressed from any entity toward any entity or event in the text, and jointly resolving various ambiguities in the entity/event-level sentiment analysis task.

TABLE OF CONTENTS

1.0 INTRODUCTION

There is an increasing number of opinions expressed online in various genres, including reviews, newswire, editorial, blogs, etc. The work in sentiment analysis and opinion mining is continuously moving forward. Early work in opinion mining focus on document-based analysis, e.g., judging the writer's attitude toward a product or a movie by analyzing the writer's review [Pang et al., 2002, Turney, 2002]. To fully understand and utilize the opinions, much work in sentiment analysis and opinion mining in recent years begin to focus on more-fined grained levels, including sentence-based sentiment analysis which detects the sentiment expressed by a sentence [Yu and Hatzivassiloglou, 2003, McDonald et al., 2007], span-based sentiment analysis which recognizes the sentiment expressed by a span [Yang and Cardie, 2013, Johansson and Moschitti, 2013a], and aspect-level sentiment analysis which recognizes the sentiment expressed toward an aspect [Hu and Liu, 2004, Titov and McDonald, 2008], etc. A fine-grained sentiment analysis gives us an opportunity to better understand different opinions the writer expresses throughout the document by extracting the components of the **opinions frames**: the **source** (whose sentiment is it), the **polarity** (positive or negative), and the **target** (what is the sentiment toward). Most of the previous work make the assumption that the opinion frames are revealed by certain opinion expressions. Thus they focus on extracting **explicit opinions** where the sentiments are expressed via sentiment words or phrases. Different from them, this thesis tries to extract **implicit opinions** where the sentiments are not directly expressed toward the target, but it requires **inference** to understand the opinions. Conducting such inferences is easy for humans but difficult for automatic systems. Let's consider several excerpts from the news articles talking about the legislation of gay marriage, listed below.

Ex(1.1) Activists gather on steps of Cleveland City Hall to celebrate legalization of **gay marriage**. [1]
Ex(1.2) Hundreds of thousands of people travelled from all over Italy and Europe yesterday to protest against the proposed legalisation of **gay marriage**, ...[2]
Ex(1.3) The Australian newly weds wanted to protest the country's **same-sex marriage** ban during their wedding.[3]

In each example, the phrases highlighted in red are negative sentiment expressions and the phrases highlighted in green are positive sentiment expressions. All of them are explicit sentiments. The phrases in red are the sources, who express the explicit sentiments. The phrases in blue are the targets of the explicit sentiments, toward which the explicit sentiments are expressed. In Ex(1.1), the activists are positive toward the legalization of gay marriage, revealed by the sentiment phrase *celebrate*. In Ex(1.2), people have negative sentiment toward legalisation of gay marriage, revealed by the sentiment phrase *protest against*. In Ex(1.3), the Australian newly weds are negative toward the the country's same-sex marriage ban, revealed by *protest*. The opinion frames revealed by the expressions above have two things in common: first, they are explicit sentiments because they are revealed by sentiment expressions, highlighted in either red or green. Second, the targets of the explicit sentiments are **events**, i.e., legalization in Ex(1.1) and Ex(1.2) and ban in Ex(1.3). However, there are more opinions conveyed here. In Ex(1.1), the activists support the gay marriage. In Ex(1.2), people are against the gay marriage. In Ex(1.3), the newly weds are positive toward the same-sex marriage. These sentiments toward the gay marriage or same-sex marriage also have two things in common: first, they are implicit because we cannot find a sentiment expression directly linking the source to the target. Second, the targets are **entities**, i.e., gay marriage in Ex(1.1) and Ex(1.2) and same sex marriage in Ex(1.3). In the examples above, a sentiment analysis system purely relying on recognizing sentiment expressions may detect the sentiments toward the events but it is not able to detect the writer's sentiment toward the entities, which are the themes of the events. What's more, the explicit sentiments and the implicit sentiments even from the same source are not always the same. For instance, in Ex(1.3), the sentiment toward the event ban is negative, while the sentiment toward the

[1] http://www.cleveland.com/metro/index.ssf/2015/06/activists_gather_on_steps_of_c_1.html
[2] http://www.breitbart.com/london/2015/06/21/over-300000-rally-in-rome-against-gay-marriage/
[3] http://elitedaily.com/news/world/wedding-cover-ears-protest-gay-marriage-ban-photos/1096481/

entity same sex marriage is positive.

Both explicit and implicit opinions are important to fully understanding the text. This work especially focuses on a new task, the **entity/event-level sentiment analysis** task, where the source and the target of an opinion frame are either entities or events, as we have seen in Ex(1.1)-Ex(1.3). In this task, a positive entity/event-level sentiment is named **PositivePair**(s,t), representing there are positive sentiments expressed from the source s toward the target t. Similarly a negative entity/event-level sentiment is named **NegativePair**(s,t). Therefore, this work develops annotated corpora and investigate computational models to detect and infer both explicit and implicit PositivePairs and NegativePairs in the text.

1.0.1 COMPLETED AND PROPOSED WORK

Wiebe and Deng [Wiebe and Deng, 2014] have developed a conceptually rule-based framework to infer sentiments and beliefs. The work provides a set of inference rules defining how to infer sentiments. But the rule-based system takes as input *all* the manual annotation excluding the implicit sentiments. To automatically extract all the PositivePairs and NegativePairs, we propose three parts of work. First of all, we propose to develop an entity/event-level sentiment corpus where both explicit and implicit sentiments are annotated, serving as the resource for training and evaluation for this work. Then, we begin investigating computational models to infer sentiments using a subset of the rules in [Wiebe and Deng, 2014], which focus on sentiments expressed toward a particular type of event: +/-effect event. Finally, we propose to utilize all the rules inferring sentiments in [Wiebe and Deng, 2014], which are not limited to +/-effect events. With more complexity introduced by more rules, joint prediction models will be investigated to resolve various ambiguities of an entity/event-level sentiment analysis task.

1.0.2 An Entity/Event-Level Sentiment Corpus (Partially Completed)

First of all, we propose to develop an entity/event-level sentiment corpus, MPQA 3.0. The corpus will be annotated with both explicit and implicit sentiments, which are expressed from entities as source toward entities or events as target. The corpus is used as the gold standard

	explicit sentiment	implicit sentiment
Ex(1.1)	positive toward (**legalization** of gay marriage)	positive toward gay marriage
Ex(1.2)	negative toward (**legalization** of gay marriage)	negative toward gay marriage
Ex(1.3)	negative toward (**ban** of same-sex marriage)	positive toward same-sex marriage

Table 1: Explicit and Implicit Sentiments Expressed in Ex(1.1), Ex(1.2) and Ex(1.3).

corpus for this thesis work. In the future, it will serve as a resource in the field of sentiment analysis and opinion mining. What's more, the corpus will facilitate research work in the other fields of natural language processing, such as opinion question answering. A system trained via the corpus annotated with entities and events as target can answer questions such as "Who is negative/positive toward X?" [Stoyanov et al., 2005]. Two expert annotators have annotated several documents in MPQA 3.0 and the agreement study has positive results [Deng and Wiebe, 2015b]. To reduce the labor and cost of expert annotations, we propose to crowd-source annotations from non-expert annotators and propose annotation schema for non-expert annotators.

1.0.3 +/-Effect Events Focused Sentiment Detection and Inference (Completed)

Second, we have developed systems to infer the implicit sentiments expressed toward a particular type of events, +/-effect events, and the entities participating in the +/-effect events [Deng et al., 2013, Deng and Wiebe, 2014, Deng et al., 2014]. As defined in [Deng et al., 2013], a +/-effect event either benefits the theme of the event, or harms the theme of the event. We choose +/-effect events as a start because many implicit sentiments are expressed via such events. For instance, the examples above all contain +/-effect events. Let's summarize the sentiments and the events in Ex(1.1), Ex(1.2) and Ex(1.3) in Table 1.

As we can see, the syntactic structures in Ex(1.1), Ex(1.2) and Ex(1.3) are the same. In Ex(1.1) and Ex(1.2), the event is *legalization* which benefits the theme, *gay marriage*. While in Ex(1.3), the event is *ban* which is harmful to the theme, *same-sex marriage*. We

define events such as legalization as **+effect event**, which has positive effect on the theme [Deng et al., 2013]. And we define events such as ban as **-effect event**, which has negative effect on the theme [Deng et al., 2013]. In Ex(1.1) and Ex(1.3), the +/-effect events are the same. But different explicit sentiments result in different implicit sentiments. In Ex(1.2) and Ex(1.3), the explicit sentiments are the same. But different +/-effect events result in different implicit sentiments.

To study how explicit sentiments interact with +/-effect events to express implicit sentiments, first we have developed a +/-effect event corpus [Deng et al., 2013]. The +/-effect events, the agents who conduct the events, the themes whom the events affect, and the influencers (e.g., negations) of the events are annotated. The writer's attitudes toward the agents and the themes are annotated as well, including both explicit sentiments and implicit sentiments. Based on the corpus, we have developed two computational models encoding several inference rules to infer implicit sentiments based on explicit sentiments and the +/-effect event information. The inference rules are adopted from [Wiebe and Deng, 2014]. The first model demonstrates the inference ability of the rules via a graph-based model built on manual annotations in the +/-effect event corpus [Deng and Wiebe, 2014]. The inference rules are used to assign scores to the edges of the graph. The second model largely reduces the need for manual annotations, and improves detecting sentiments expressed toward the agents and themes of +/-effect events via a global optimizing framework, Integer Linear Programming (ILP) [Deng et al., 2014]. The rules are encoded as constraints of the ILP framework.

1.0.4 Entity/Event-Level Sentiment Detection and Inference (Partially Completed)

Finally, we propose to build a system detecting and inferring implicit sentiments expressed toward all types of events and entities, which are not limited to +/-effect events. The source can be the writer or any entity, and the target can be any entity or event. The rules used to infer entity/event-level sentiment are adopted from [Wiebe and Deng, 2014]. Taking into account all of the rules, we propose to develop joint models where the rules are encoded as

constraints. Different from the computational models focusing on +/-effect events above, which need manual annotations to some extent, the joint model extracts all the information automatically. Given the noisy input by automatic modules instead of gold standard, it deals with more complexity. The joint model will be evaluated using the MPQA 3.0 corpus.

Currently we have conducted a pilot study [Deng and Wiebe, 2015a] of extracting entity/event-level sentiment annotations on a subset of MPQA 3.0. The pilot study implements the sentiment inference in a statistical relational learning framework, where the inference rules in first order logic are used as constraints in the framework. The experiments have shown that the joint model can improve accuracy of entity/event-level sentiment analysis over baselines. However, the pilot study makes several simplifications of the real problem. We propose to improve the framework w.r.t. each simplification in the pilot study: first, as stated in [Wiebe et al., 2005] the sources of opinions annotated in MPQA are nested (See Section 3 for details). The pilot study does not extract nested source but the most immediate source. Second, we propose to employ state-of-the-art named entity recognition and event recognition techniques to facilitate detecting the targets, instead of considering the head of any noun phrase or verb phrase (NP/VP) in the pilot study. Third, as pointed out in [Wiebe and Deng, 2014], the inference rules are implicatures and they could be defeated, we propose to detect when the inference rules should be blocked so that the inference is not carried out in those cases. Finally, a better entity/event-level sentiment analysis system will be investigated integrating each improved component above.

1.1 MAIN CONTRIBUTIONS

This thesis mainly contributes to the field of sentiment analysis and opinion mining.

- Develop an entity/event-level sentiment corpus annotated with both explicit and implicit sentiments expressed among entities and events. The entity/event-level sentiment analysis is a more fine-grained task than previous work, which are mostly span based.
- Develop an annotated corpus and two computational models focusing on the sentiments about a particular type of events: +/-effect event. The corpus is annotated with +/-effect

event information and sentiments toward the agents and themes of the +/-effect events. One model investigates the inference ability of the rules, and the other model improves detecting the sentiments expressed toward the agents and themes of the +/-effect events using a small amount of the gold standard data.

- Develop joint prediction models to improve detecting entity/event-level sentiments in the text. Compared to previous work, the entity/event-level sentiment analysis task is a more fine-grained and more difficult task. The prediction model jointly resolves the various ambiguities in the task and infers implicit sentiments based on the inference rules.

1.2 THESIS STATEMENT

Defining a new sentiment analysis task (entity/event-level sentiment analysis task), this work develops annotated corpora as resources of the task and investigates joint prediction models integrating explicit sentiments, entity or event information and inference rules together to automatically recognize both explicit and implicit sentiments expressed among entities and events in the text.

1.3 OUTLINE

The rest of this proposal is organized as follows: Chapter 3 introduces the annotation schema for the entity/event-level sentiment corpus, MPQA 3.0. Different schema are proposed for expert annotations and non-expert annotations. Chapter 4 gives the definition of +/-effect events, introduces the +/-effect event corpus, presents the sentiment inference rules applied to +/-effect events and talk about the developed computational models. Chapter 5 introduces the inference rules applied to all types of entities and events, and proposes joint prediction models to infer implicit sentiments expressed by an entity toward another entity or event. Finally, Chapter 6 gives the timeline of this proposed work and Chapter 7 summarizes this proposed work.

2.0 BACKGROUND

Previous work differs from the entity/event-level sentiment detection and inference we address in terms of targets, sources and sentiment inference.

In terms of targets, in a span-based sentiment analysis system, the target is a span instead of the exact head of the phrase referring to the target. The target in a span-based system is evaluated by measuring the overlapping proportion of an extracted span against the gold standard phrase [Yang and Cardie, 2013], while the eTarget in an entity/event-level system is evaluated against the exact word (i.e., head of NP/VP) in the gold standard. It is a stricter evaluation. Recognizing targets in news genre is also different from recognizing targets in review data. Annotated corpora of reviews (e.g., [Hu and Liu, 2004, Titov and McDonald, 2008]), widely used in natural language processing, often include target annotations. Such targets are often aspects or features of products or services, and as such are somewhat limited. For example, as stated in SemEval-2014: "We annotate only aspect terms naming particular aspects." Recently, to create the Sentiment Treebank [Socher et al., 2013], researchers crowd-sourced annotations of movie review data and then overlaid the annotations onto syntax trees. Thus, the targets are not limited to aspects of products/services. However, annotators were asked to annotate small and then increasingly larger segments of the sentence. Thus, the annotations are mixed in the degree to which context was considered when making the judgements.

In terms of source, in the annotated corpora above and the work implemented on the corpora, the only sentiments considered are those of the writer, excluding sentiments attributed to other entities. While in MPQA 3.0, the source can by any entity and the source are nested.

In terms of sentiment inference, most work in sentiment analysis focuses on classi-

fying explicit sentiments and extracting explicit opinion expressions, holders and targets [Wiebe et al., 2005, Johansson and Moschitti, 2013a, Yang and Cardie, 2013]. There is some work investigating features that directly indicate implicit sentiments [Zhang and Liu, 2011, Feng et al., 2013]. However, identifying terms that imply opinions is a different task than sentiment propagation between entities. Dasigi et al., [Dasigi et al., 2012] search for implicit attitudes shared between authors, while we address inferences within a single text. Several papers apply compositional semantics to determine polarity (e.g., [Moilanen and Pulman, 2007, Choi and Cardie, 2008, Moilanen et al., 2010]). The goal of such work is to determine one overall polarity of an expression or sentence. In contrast, our framework commits to a holder having sentiments toward various events and entities in the sentence, possibly of different polarities. In short, we focus on how we can bridge between explicit and implicit sentiments via inference. The idea of +/-effect events in sentiment analysis is not entirely new. For example, two papers mentioned above [Zhang and Liu, 2011, Choi and Cardie, 2008] include linguistic patterns for the tasks that they address that include +/-effect events, but they don't define general implicature rules relating sentiments and +/-effect events, agents, and objects as we do. Recently, in linguistics, Anand and Reschke [Anand and Reschke, 2010, Reschke and Anand, 2011] identify classes of +/-effect event terms, and carry out studies involving artificially constructed +/-effect events and corpus examples matching fixed linguistic templates. Our work focuses on +/-effect event triples in naturally-occurring data and uses generalized implicature rules. Goyal et al. [Goyal et al., 2012] generate a lexicon of *patient polarity verbs*, which correspond to +/-effect events whose spans are verbs. Riloff et al. [Riloff et al., 2013] investigate sarcasm where the writer holds a positive sentiment toward a negative situation. However, neither of these works performs sentiment inference. Previously, Wiebe and Deng [Wiebe and Deng, 2014] also propose a set of sentiment inference rules and develop a rule-based system to infer sentiments. However, the rule-based system requires *all* information regarding explicit sentiments and +/-effect event information to be provided as oracle information by manual annotations.

3.0 MPQA 3.0: ENTITY/EVENT-LEVEL SENTIMENT CORPUS

In this proposal, we specifically address sentiments toward entities and events (i.e., **eTargets**) expressed in data such as blogs, newswire, and editorials. In this chapter, we introduce an ongoing work of developing an entity/event-level sentiment corpus, MPQA 3.0. The new corpus promises to be a valuable new resource for developing systems for entity/event-level sentiment analysis. Such systems, in turn, would be valuable in natural language processing applications such as Automatic Question Answering. We will first introduce the corpus and several examples from the corpus in Section 3.1. Part of the corpus has been annotated by two expert annotators [Deng and Wiebe, 2015b]. The annotation scheme and the agreement study will be presented in Section 3.2. Next, we talk about one of the proposed work of crowd-sourcing annotations of entity/event-level sentiments in Section 3.3. Finally we will give the summary.

3.1 MPQA 3.0: ENTITY/EVENT-LEVEL SENTIMENT CORPUS

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

In this work, we develop the entity/event-level sentiment corpus, MPQA 3.0, based on

the annotations in the MPQA 2.0 corpus [Wiebe et al., 2005, Wilson, 2008].[1] The MPQA opinion annotated corpus is entirely span-based, and contains no eTarget annotations. However, it provides an infrastructure for sentiment annotation that is not provided by other sentiment NLP corpora, and is much more varied in topic, genre, and publication source. In this section, we talk about how we develop MPQA 3.0 based on MPQA 2.0 in Section 3.1.1, and present several examples we have annotated in Section 3.1.2.

3.1.1 From MPQA 2.0 to MPQA 3.0

To create MPQA 3.0, entity-target and event-target (*eTarget*) annotations are added to the MPQA 2.0 annotations. The MPQA 2.0 annotations consist of *private states*, which are states of *sources* holding *attitudes* toward *targets*. There are several types of attitudes included in MPQA 2.0 [Wilson, 2008, Somasundaran et al., 2007], including sentiment, arguing, etc. This work focusses on sentiments, which are defined in [Wilson, 2008] as positive and negative evaluations, emotions, and judgements. In the future, eTargets may be added to private states with other types of attitudes.

In the MPQA 2.0 annotations, the top-level annotations are **direct subjective** (**DS**) and **objective speech event** annotations. DS annotations are for private states, and objective speech event annotations are for objective statements attributed to a source. As shown in Figure 1, one DS may contain links to multiple **attitude** annotations, meaning that all of the attitudes share the same nested source. The attitudes differ from one another in their attitude types, polarities, and/or targets. MPQA 2.0 also contains **expressive subjective element** (**ESE**) annotations, which pinpoint specific expressions used to express subjectivity [Wiebe et al., 2005]. An ESE also has a nested-source annotation. Since we focus on sentiments, we only consider ESEs whose polarity is positive or negative (excluding those marked neutral).

An important property of sources in MPQA is that they are *nested*, reflecting the fact that private states and speech events are often embedded in one another. Consider an example from MPQA.

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

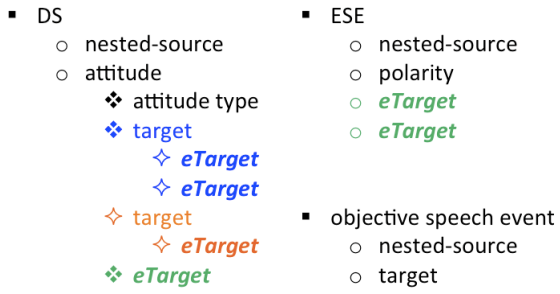

Figure 1: Structure in MPQA 3.0.

Ex(3.1) When the Imam issued the fatwa against Salman Rushdie for insulting the Prophet ...

In the example, Imam has a negative sentiment, revealed by the phrase *issued the fatwa against*. In the MPQA corpus, the negative sentiment is captured by an objective speech event annotation whose target span includes the whole sentence, and whose source is the writer. In other words, this negative sentiment is presented only within the scope of this article. The complete interpretation of this negative attitude is, according to the writer, the Imam has a negative sentiment. Thus, though the negative sentiment is attributed to Imam, more importantly it is attributed to the "Imam" in this particular writer's description. In MPQA, the complete source is represented in a nested structure (i.e., **nested-source**) and the annotated nested-source is *(writer, imam)*. We call *writer* as the **top-level source** of the nested source and call *Imam* as the **most immediate source**.

The *target-span* annotations in MPQA 2.0 are linked to the attitudes. More than one target may be linked to an attitude, but most attitudes have only one target. The MPQA 2.0 annotators identified the main/most important target(s) they perceive in the sentence. If there is no target, the target-span annotation is "none". However, there are many other eTargets to be identified. First, while ESE annotations have nested sources, they do not have any target annotations. Second, there are many more targets that may be marked than the major ones identified in MPQA 2.0. In Figure 1, the eTargets are what we add in MPQA 3.0. We identify the blue (orange) eTargets that are in the span of a blue (orange)

target in MPQA 2.0. We also identify the green eTargets that are not in the scope of any target. Since our priority is to add eTargets to sentiments, no eTargets have yet been added to objective speech events, as shown in Figure 1.

By adding eTargets to the existing annotations, the information in MPQA 2.0 is retained.

3.1.2 Examples

In this section, we list the annotations in MPQA 3.0 of Ex(3.1) sentence. The phrase in blue is an attitude span, the phrase in red is a target span, the tokens in yellow are the eTargets which are newly annotated in MPQA 3.0. Each example is followed by the MPQA annotations.

In Ex(3.2), a negative attitude is shown, *issued the fatwa against*. The source is *the Imam*. The target is the event *Rushdie insulting the Prophet*. However, the assertion that the Imam is negative toward the insult event is within the scope of this article. As we stated above, the nested source is *(writer, Imam)*.

Ex(3.2) When the Imam issued the fatwa against Salman Rushdie for insulting the Prophet ...

DS: issued the fatwa
  nested-source: w, imam
  attitude: issued the fatwa against
    attitude-type: sentiment-negative
    target: Salman Rushdie for insulting the Prophet
      eTarget : Rushdie, insulting

We find two eTargets in the target-span: "Rushdie" himself plus his act of "insulting."

In the same sentence, there is another negative attitude, *insulting*, as shown in Ex(3.3). The source is *Salman Rushdie* and the target is *the Prophet*. Note that the span covering this event is the target span of the attitude in Ex(3.2) — the private state of Ex(3.3) is nested in the private state of Ex(3.2). Thus, the complete interpretation of the negative attitude in Ex(3.3) is: according to the writer, the Imam is negative toward Rushdie insulting the Prophet. The nested source is *w, Imam, Rushdie*.

Ex(3.3) When the Imam issued the fatwa against Salman Rushdie for insulting the Prophet ...

DS: insulting
  nested-source: w, imam, rushdie
  attitude: insulting
    attitude-type: sentiment-negative
    target: the Prophet
      eTarget : Prophet

We add an eTarget for the Prophet, anchored to the head "Prophet." Interestingly, "Prophet" is an eTarget for *w,Iman,Rushdie* (i.e., Rushdie is negative toward the Prophet), but not for *w,Imam* (i.e., the Imam is not negative toward the Prophet).

3.2 EXPERT ANNOTATION

In this section, we introduce how we annotate eTargets in the MPQA 3.0 corpus.

As defined in Section 3.1, an **eTarget** is an entity or event that is the target of a sentiment (identified in MPQA 2.0 by a sentiment attitude or polar ESE span). To create MPQA 3.0, the corpus is first parsed, and *potential* eTarget annotations are automatically created from the heads of NPs and VPs. The annotators then consider each sentiment attitude and each polar ESE, and decide for each which eTargets to add.

For the formal agreement study, one document is randomly selected from each of the four topics of the OPQA subset [Stoyanov et al., 2005] of the MPQA corpus. They are not any of the documents used to develop the manual. The two annotators then independently annotate the four documents. There are 292 eTargets in the four documents in total.

To evaluate the results, the same agreement measure is used for both attitude and ESE eTargets. Given an attitude or ESE, let set A be the set of eTargets annotated by annotator X, and set B be the set of eTargets annotated by annotator Y. Following [Wilson and Wiebe, 2003, Johansson and Moschitti, 2013b], which treat each set A and B in turn as the gold-standard, we calculate the average F-measure, denoted $agr(A, B)$. The $agr(A, B)$ is 0.82 on average over the four documents, showing good agreement.

$$agr(A, B) = \frac{|A \cap B|/|B| + |A \cap B|/|A|}{2} \tag{3.1}$$

3.3 NON-EXPERT ANNOTATION

The agreement study in Section 3.2 shows that expert annotations have reached a good agreement. However, acquiring expert annotations are time consuming and costly. To reduce the expert annotation effort, we propose crowd-sourcing annotations for the MPQA 3.0 corpus. The idea of crowd-sourcing is gathering contributions from a large group of people, rather than a small group of experts or employees. In the field of natural language processing, crowd-sourcing is a useful tool to collect annotations [Akkaya et al., 2010, Socher et al., 2013]. In this section, we talk about how we propose to set up our annotation task in Section 3.3.1, and how we propose to filter out unreliable annotations and acquire useful annotations in Section 3.3.2.

3.3.1 Annotation Scheme for Non-expert Annotators

The annotators in a crowd-sourcing task should not need professional training for this annotation task. Thus, the task should be designed easy to understand. Previously, Socher et al., [Socher et al., 2013] have used Amazon Mechanical Turk to collect sentiment labels for phrases in the movie review data. Given an segment of a sentence, the non-expert annotator selects the sentiment and its degree by moving a slider, as shown in Figure 2 [Socher et al., 2013]. Their crowd-sourced annotations have reached good agreement, which shows the non-expert annotators are able to give the correct sentiment labels. In our task, we will present the whole sentence instead of the segment of the sentence in [Socher et al., 2013]. The whole sentence contains more information than the sentence segment for our task. We will also provide the span based opinion annotations already marked in MPQA 2.0 to facilitate eTarget annotations in MPQA 3.0.

For an opinion annotation in MPQA 2.0 (including the sentiment and ESE), we provide the nested source, the target span, the opinion span and the opinion polarity. For each annotation task, an eTarget candidate is also provided. Each annotation task is designed as a binary question. If the polarity of the opinion annotation in MPQA 2.0 is positive, then the question is: "Is the source A positive toward the eTarget B?", where the source A is the

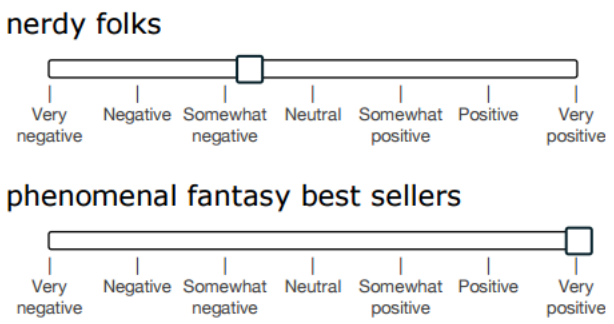

Figure 2: Sentiment Treebank Annotation Interface [Socher et al., 2013].

span of the most immediate source, and the eTarget B is one of the eTarget candidates. If the polarity of the opinion annotation in MPQA 2.0 is negative, then the question is: "Is the source A negative toward the eTarget B?". For example, for Ex(3.1) in Section 3.1.2, one of the questions is:

In the sentence:
When [the Imam] issued the fatwa against [Salman Rushdie for insulting the Prophet] ...

Given that:
[The Imam] is negative toward [**Salman Rushdie** for insulting the Prophet].

Thus:
Is the Imam negative toward Salman Rushdie?

If we regard each NP/VP head as potential eTarget in the non-expert annotations and create a question for each NP/VP head, there are too many questions to submit and the costs are high. Thus, we need to limit the size of eTarget candidates.

First, if there is only one NP/VP head in the target span, then we regard the eTarget candidate as the correct eTarget.

Second, we create a question for each NP/VP head as eTarget candidate in the target span. In MPQA 2.0, a target annotation of an opinion captures the most important target this opinion is expressed toward [Wiebe et al., 2005, Wilson, 2008]. What's more, the target annotation is the shortest span that can capture the target of the opinion [Wilson, 2008].

Thus, the information in the target span in MPQA 2.0 is very dense so we create a question for each NP/VP head in the target span.

Similarly, we create a question for each NP/VP head as eTarget candidate in the opinion span, if the source of the opinion is not the agent of the opinion span. In our expert annotation, we find there are many eTargets in the opinion spans [Deng and Wiebe, 2015b]. However, if the source is the agent of the opinion span, e.g. in Ex(3.2) Imam is the agent of the opinion expression (Imam issued the fatwa against ...), we don't consider the NP/VP heads in the opinion span since we don't assume the agent has private state toward the actions that agent conducted.

Finally, after we gather the annotations for each NP/VP head in the target span and the opinion span, we create questions for possible eTargets outside the target span and the opinion span. We hypothesize that, if there is no sentiment expressed toward the overall phrase, represented as the head of the phrase, we don't think there is any sentiment expressed **from the same source** toward more specific information in the phrase, represented as the child of the head. Based on this hypothesis, the annotation scheme for non-expert annotators is designed as two steps. In the first step, we go up the parse tree from target span semantic head. We create a question for the parent of the target span head and then present the question to the mechanical turkers. If the answer from the turkers is "yes" meaning that the **same source** has a **same polarity** sentiment toward it, then we add it as the eTarget and create question for its parent. If the answer is "no", then we stop there. We keep going up the parse tree until we meet a "no" answer or we reach the root of the whole sentence. If there are multiple target spans of the sentiment, we conduct the process for each target head. The we go to the second step. In the second step, we start from each new eTarget which are not in any target span, then create one question for each child of it. If the answer to the question w.r.t. the child is "yes", then we create one question for each child further down. If the answer is "no", we then stop.

Before annotating the remaining corpus, we will first examine the feasibility of this annotation scheme for non-expert annotators. We propose to create questions from the documents which have been annotated by the experts. Non-expert annotations of these questions will be collected. The expert annotations will be used as gold standard. We will

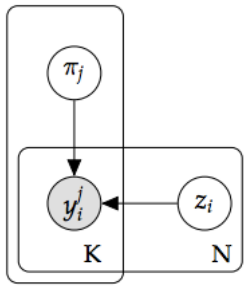

Figure 3: Dawid and Skene Model [Dawid and Skene, 1979].

examine the qualify of the non-expert annotations compared to the expert annotations.

3.3.2 Acquiring High Quality Annotations

The annotations collected from crowd-sourcing are noisy, because the turkers are not specially trained for the particular task and may not pay full attention to the annotation task. For our task, high quality annotations are very important, since the further questions we create to collect more annotations are based on the annotations of the current question. Thus, in this section, we discuss about several methods we can use to get high-quality annotations.

First of all, some crowd-sourcing platforms provide the credits of turkers. We can select the turkers with high credits, since they are more reliable than those with low credits.

Previous work have investigated how to learn the true labels from large annotations. The simplest method is taking the majority vote, especially our question is a yes or no question. However, the method of majority vote assumes that all the turkers are equally careful and qualified, and all the answers have the same weight. Such assumption doesn't hold in all the cases. Thus, Dawid and Skene [Dawid and Skene, 1979] have developed a model to learn the weight of each answer, where the answer quality π_i and the true label z_j jointly determines the observed label y_i^j. As shown in Figure 3, we can use EM algorithm to learn the answer qualities π. Then a weighted vote can be conducted to get the true labels.

Further, Welinder and Perona [Welinder and Perona, 2010] develop a model to learn the true label of each question directly. Each answer i has an unknown label z_i, and each turker

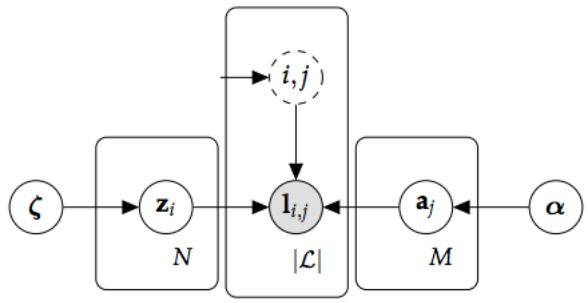

Figure 4: Welinder and Perona Model [Welinder and Perona, 2010].

j has an unknown quality a_j. Similarly, the observed label l_{ij} is determined together by z_i and a_j. We calculate the probability of the annotation process as

$$p(\mathcal{L}, a, z) = \prod_{i=1}^{N} p(z_i|\varepsilon) \prod_{j=1}^{M} p(a_j|\alpha) \prod_{l_{ij} \in \mathcal{L}} p(l_{ij}|z_i, a_j). \tag{3.2}$$

Note that Welinder and Perona's model not only gives the true labels, but also learns the reliability of annotators. Therefore, we will use the subset of the MPQA 3.0 corpus which are already annotated by expert annotators to validate the feasibility of the annotation scheme in Section 3.3.1 and **simultaneously** find the reliable annotators. The reliable annotators are then selected to annotate the remaining MPQA 3.0 corpus.

3.4 SUMMARY

This chapter presents adding entity and event target (eTarget) annotations to the MPQA corpus. A subset of MPQA has already been annotated according to the annotation scheme for expert annotators. The agreement study between the expert annotators shows that the annotation scheme is able to guide expert annotators to add the eTarget annotations. To reduce the annotation effort and cost, we propose to crowd-source the eTarget annotations for the remaining corpus. An annotation scheme for non-expert annotators is proposed, with

possible methods to collect reliable labels from the noisy annotations.

4.0 +/-EFFECT EVENT FOCUSED SENTIMENT DETECTION AND INFERENCE

Many implicit opinions are expressed via the +/-effect events, such as Ex(1.1) Ex(1.2) and Ex(1.3) in Chapter 1. Though the sentence structure in the examples are the same, different +/-effect events (i.e., legation and ban) evoke different implicit opinions toward the same sex marriage. Thus, we choose to start with inferring sentiments expressed toward +/-effect events and the entities involved in the +/-effect events. The +/-effect events have a clear definition and the inference are intuitive to understand.

In this chapter, we present a focused study of inferring sentiments expressed toward +/-effect events and the sentiments expressed toward the entities participating in the +/-effect events. We will introduce the definition of +/-effect events in Section 4.1. An corpus annotated with sentiments and +/-effect event information is given in Section 4.2. We present inference rules developed by [Wiebe and Deng, 2014] in Section 4.3. Then we embed the inference rules in the computational models to recognize implicit opinions in Section 4.4. Finally we give the summary.

4.1 DEFINITION OF +/-EFFECT EVENT

A +effect event is defined as an event which has positive effect on the theme of the event, including creation events (as in *bake a cake*, which creates the cake and subsequently has positive effect on the cake), gaining events (as in *increasing costs*, which contributes to the costs), and benefiting events (as in *comforted the child*, which is good for the child), etc [Deng et al., 2013, Anand and Reschke, 2010]. A -effect event is defined as an event which

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

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

Ex(4.1) His uncle **left** him ***a massive amount of debt***.
Ex(4.2) His uncle **left** him ***a treasure***.

There is no way to break these sentences into triples that follow our rules. ⟨His uncle, left, him⟩ doesn't work because we cannot decide whether it is a +effect event or a -effect event by looking only at the triple; whether it is +effect or -effect depends on what his uncle left him. ⟨His uncle, left him, a massive amount of debt⟩ isn't correct: the event is not bad for the debt, it is bad for *him*. Thus, the theme of the event should be *him* instead of *debt*. Finally, ⟨His uncle, left him a massive amount of debt, *Null*⟩ isn't correct, since no theme is identified.

Agent and Theme. The agent of an +/-effect event must be a noun phrase, or it may be *implicit*. In the sentence, *the constituent will be destroyed*, the agent of the destroying event is implicit. The theme must be a noun phrase.

Influencer. Another component of a +/-effect event is the **influencer**, a word either retains or reverses the effect of a +/-effect event. We define the phrase reversing the effect as **reverser**, and the phrase retaining the effect as **retainer**. In the examples below, the word in italics are influencers, and the words in boldfaced are +/effect events.

Ex(4.3) Luckily Bill *didn't* **kill** him.
Ex(4.4) The reform *prevented* companies from **hurting** patients.
Ex(4.5) John *helped* Mary to **save** Bill.

In Ex(4.3) and Ex(4.4), *didn't* and *prevented*, respectively, reverse the effect of the event from -effect to +effect (i.e., not killing Bill has positive effect on Bill; preventing companies from hurting patients has positive effect on the patients). They are reversers. In Ex(4.5), *helped* is an influencer which retains the +effect (i.e., helping Mary to save Bill is good for Bill, as saving Bill is good for Bill). It is a retainer.

An influencer has an agent and a theme. The agent must be a noun phrase or *implicit*. Ex(4.4) and Ex(4.5) illustrate the case where the agent of the influencer is different from the agent of the +/-effect event. For instance, the agent of the influencer *prevented* in Ex(4.4) is the *reform*, while the agent of the -effect event *hurting* is the *patients*. The theme must be another influencer or a +/-effect event.

Note that, semantically, an influencer can be seen as a +/-effect event of its theme. A reverser influencer makes its theme irrealis (i.e., not happen). Thus, it has negative effect on it. In Ex(4.3), for example, *prevent* is bad for the *hurting* event. A retainer influencer maintains its theme, and thus has positive effect on it. In (4), for example, *helped* maintains the *saving* event.

4.2 +/-EFFECT EVENT CORPUS

To study the implicit sentiments expressed via +/-effect events, we have annotated a +/-effect event corpus. There are four types of annotations in the corpus: +/-effect event, influencer, agent, and theme. For +/-effect event annotations, the corresponding agents, themes, and effects of the events (+effect or -effect) are identified. For influencer annotations, the agents, themes and whether they are retainers or reversers are identified. For agent and theme annotations, the writer's attitudes toward them are marked (positive, negative, or none). The annotator is required to link agents and themes to the corresponding +/-effect events or influencer annotations via explicit IDs. When an agent is not mentioned explicitly,

the annotator should indicate that it is *implicit*. For example:

Ex(4.6) GOP **Attack on** Reform Is a **Fight Against** Justice.
Ex(4.7) **Jettison** any reference to end-of-life counselling.

In Ex(4.6), the annotators are expected to identify two -effect events: ⟨GOP, Attack on, Reform⟩ and ⟨GOP Attack on Reform,Fight Against, Justice⟩. The effects of both events are -effect. The writer's attitudes toward both agents are negative, and the writer's attitudes toward both themes are positive. In Ex(4.7), the annotator is expected to identify the -effect event: ⟨implicit, Jettison, any reference to end-of-life counselling⟩. The writer conveys a negative attitude toward *end-of-life counselling*.

We have carried out an agreement study based on the annotation schema, which has positive results. Due to the space limit, we do not list the details of the agreement study. For more details, please refer to [Deng et al., 2013].

4.3 SENTIMENT INFERENCE RULES

The rules used to infer the sentiments toward +/-effect events are adopted from [Wiebe and Deng, 2014]. In [Deng and Wiebe, 2014], we present the rules as four inference rule schema. Four specific rules are presented for each rule schema. Here we only present Rule Schema 1. For a full list, please refer to Appendix A. In Chapter 1, we have summarize the sentiments in the examples in Table 1. Here we present a more detailed analysis of inferring the implicit sentiments in Table 2.

The inference proceeds as follows: In Ex(1.1), because people are positive toward an event which benefits the gay marriage, we infer that people are positive toward the gay marriage. Similarly, in Ex(1.2), because people don't like the event that benefits the gay marriage, we infer that people are negative toward the gay marriage. In Ex(1.3), because people are negative that something bad happens to the same-sex marriage, we infer people are actually positive toward the same-sex marriage.

Inspired by the observations in the examples, we propose four inference rule schemas listed below, where *sent*(α) = β means that the **writer's** sentiment toward α is β, where α

	explicit sentiment	+/-effect	implicit sentiment
Ex(1.1)	positive toward legalization	+effect	positive toward gay marriage
Ex(1.2)	negative toward legalization	+effect	negative toward gay marriage
Ex(1.3)	negative toward ban	-effect	positive toward same-sex marriage

Table 2: Explicit and Implicit Sentiments Expressed in the Examples in Chapter 1.

can be a +effect event, a -effect event, or the agent or theme of a +/-effect event, and β is either *positive* or *negative*. P $\rightarrow$ Q is to infer Q from P. For example, Rule 1.1 applies to Ex(1.1), Rule 1.2 applies to Ex(1.2), and Rule 1.4 applies to Ex(1.3).

Rule Schema 1: sentiment(+/-effect event) $\rightarrow$ sentiment(theme)

Rule 1.1: sentiment(+effect) = positive $\rightarrow$ sentiment(theme) = positive

Rule 1.2: sentiment(+effect) = negative $\rightarrow$ sentiment(theme) = negative

Rule 1.3: sentiment(-effect) = positive $\rightarrow$ sentiment(theme) = negative

Rule 1.4: sentiment(-effect) = negative $\rightarrow$ sentiment(theme) = positive

We have observed an interesting phenomenon of the rules [Deng and Wiebe, 2014]. All cases covered by Rule Schemas 1 and 3 are shown in Table 3. From Table 3, we have observed that, regardless of the writer's sentiment toward the event, **if the event is +effect, then the writer's sentiments toward the agent and theme are the same, while if the event is -effect, the writer's sentiments toward the agent and theme are opposite**. Later we will use such relations as constraints in the computational models.

4.4 COMPUTATIONAL MODELS

Instead of building a rule-based system for sentiment analysis, we have developed two computational models where the rules are used as constraints. Using the manual annotations in the gold standard corpus in Section 4.2, the first model examines whether the rules can correctly infer implicit sentiments [Deng and Wiebe, 2014]. Largely reducing the need for manual an-

			Rule Schema 3	Rule Schema 1
sentiment(event)	effect	→	sentiment(agent)	sentiment(theme)
positive	+effect	→	positive	positive
negative	+effect	→	negative	negative
positive	-effect	→	positive	negative
negative	-effect	→	negative	positive

Table 3: Rule Schema 1 & Rule Schema 3

notations, the second model jointly infers the implicit sentiments and disambiguates each component of +/-effect events [Deng et al., 2014]. The corpus used in both models are the +/-effect event corpus in Section 4.2.

4.4.1 Validating the Rules: A Graph-based Propagation Model

We develop a graph-based model consisting of the +/-effect events and the agents and the themes. The graph classifies the sentiments toward entities (i.e., positive or negative) by propagation, defined by the inference rules in Section 4.3.

4.4.1.1 Definition of the Entity Graph. We define a graph $EG = \{N, E\}$. Each node in the node set N represents an annotated noun phrase agent or theme span. Each edge in the edge set E links two nodes if they co-occur in a +/-effect event. Consider the Ex(4.8) below.

Ex(4.8) Why would President Obama support health care reform? Because the reform could lower skyrocketing health care costs, and prohibit private insurance companies from overcharging patients.

E_1: ⟨reform, lower, **skyrocketing** costs⟩

E_2: ⟨reform, prohibit, E_3 ⟩

E_3: ⟨companies, overcharge, patients⟩

E_4: ⟨Obama, support, reform⟩

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

Given a text, we get the spans of +/-effect events and their agents and themes plus the effect of the events (i.e., +effect/-effect) from the manual annotations, and then build the graph upon them. However, the manual annotations of the writer's sentiments toward the agents and themes are used as the gold standard for evaluation.

4.4.1.2 Sentiment Inference via LBP The goal is to classify the writer's sentiments toward each node (i.e., each entity) on the graph. With graph EG containing cycles and no apparent structure, we utilize an approximate collective classification algorithm, *loopy belief propagation (LBP)* [Pearl, 1982, Yedidia et al., 2005], to classify nodes through belief message passing. The algorithm is shown in Table 4.

In LBP, each node has a score, $\Phi_i(y)$, and each edge has a score, $\Psi_{ij}(y_i, y_j)$. In our case, $\Phi_i(y)$ represents the writer's explicit sentiment toward n_i. $\Psi_{ij}(y_i, y_j)$ is the score on edge e_{ij}, representing the likelihood that node n_i has polarity y_i and n_j has polarity y_j. Since the inference rules define the relations of sentiment polarities between agents and themes, **we encode inference rules as definitions of the edge score** $\Psi_{ij}(y_i, y_j)$. Because in Table 3 the sentiments toward the agent and the theme of a +effect event are the same. Therefore, we define $\Psi_{ij}(pos, pos)$ and $\Psi_{ij}(neg, neg)$ to be 1 if the two nodes i and j are linked by a +effect edge; otherwise, it is 0. Similarly, the sentiments toward the agent and the theme are opposite. Thus we define $\Psi_{ij}(neg, pos)$ and $\Psi_{ij}(pos, neg)$ to be 1 if the two nodes are linked by a -effect edge; otherwise, it is 0.

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

repeat

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

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

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

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

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

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

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

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

  neutral, in case of a tie

Table 4: Loopy Belief Propagation

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

$$\begin{aligned} m_{i\to j}(pos) = \Psi_{ij}(pos, pos) * \Phi_i(pos) * \prod_{n_k \in \text{Neighbor}(n_i)/n_j} m_{k\to i}(pos) \\ + \Psi_{ij}(neg, pos) * \Phi_i(neg) * \prod_{n_k \in \text{Neighbor}(n_i)/n_j} m_{k\to i}(neg) \end{aligned} \tag{4.1}$$

$$\begin{aligned} m_{i\to j}(neg) = \Psi_{ij}(neg, neg) * \Phi_i(neg) * \prod_{n_k \in \text{Neighbor}(n_i)/n_j} m_{k\to i}(neg) \\ + \Psi_{ij}(pos, neg) * \Phi_i(pos) * \prod_{n_k \in \text{Neighbor}(n_i)/n_j} m_{k\to i}(pos) \end{aligned} \tag{4.2}$$

For example, the first part of Equation (4.1) means that the positive message n_i conveys to n_j (i.e., $m_{i\to j}(pos)$) comes from n_i being positive itself ($\Phi_i(pos)$), the likelihood of edge e_{ij} with its nodes n_i being positive and n_j being positive ($\Psi_{ij}(pos, pos)$), and the positive message n_i's neighbors (besides n_j) convey to it ($\prod_{k\in Neighbor(n_i)/n_j} m_{k\to i}(pos)$).

After convergence, the polarity of each node is determined by its explicit sentiment and the messages its neighbors convey to it, as shown at the end of the algorithm in Table 4. By this method, we take into account both sentiments and the interactions between entities via +/-effect events in order to discover implicit attitudes.

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

4.4.1.3 Performance of Graph Model.

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

In each graph (connected component), we assign only one of the nodes in the graph with its gold standard polarity. Then we run LBP on the graph and record whether the other nodes in the graph are classified correctly or not. The experiment is run on the graph $|S|$ times, where $|S|$ is the number of nodes in the subgraph, so that each node is assigned its gold-standard polarity exactly once. Each node is given a propagated value $|S| - 1$ times, as each of the other nodes in its graph receives its gold-standard polarity.

To evaluate the chance of a node given a correct propagated label, we use Equations (4.3) and (4.4),

$$c(a|b) = \begin{cases} 1 & \text{a is correct} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

$$\text{correctness}(a) = \frac{\sum_{b\in S_a, b\neq a} c(a|b)}{|S_a| - 1} \tag{4.4}$$

where a and b are two nodes in the graph, and S_a is the set of nodes in a's graph. Given b being assigned its gold-standard polarity, if a is classified correctly, then $correct(a|b)$ is 1; otherwise 0. $|S_a|$ is the number of nodes in a's graph. $correctness(a)$ is the percentage of

dataset	# graph	correctness
all graphs	983	0.8874
multi-node graphs	169	0.9030

Table 5: Performance of graph model itself.

assignments to a that are correct. If it is 1, then a is correctly classified given the correct classification of any single node in its graph.

As we can see in Table 5, a node has an 89% chance of being labeled a correct sentiment polarity if there is one correct explicit subjectivity node in its subgraph. If we only consider subgraphs with more than two nodes, the correctness chance is higher. The results indicate that, if given correct sentiments, the graph model will assign the unknown nodes with correct labels about 90% of the time. Further, the results indicate that the inference rules are able to infer implicit sentiments correctly most of the times across the corpus.

4.4.2 Joint Prediction: An Integer Linear Programming Model

The first model shows that the graph model based on inference rules could correctly propagate sentiments. However, to build such a graph, it needs all the manual annotations (except the sentiment labels) to build the graph. To reduce the need for manual annotations, we pursue to automatically resolve the ambiguities of the questions above. In this section, we will talk about an optimization framework which only needs manual annotations of +/-effect event spans. The framework is able to **jointly** resolve ambiguities of (1) the effect of events, (2) whether the event is reversed ,(3) the agents and themes of the events, and (4) the writer's attitudes toward agents and themes. In the framework, we first run local systems to extract candidates for each ambiguity of (1)-(4), then an Integer Linear Programming (ILP) framework jointly resolves all the ambiguities by selecting an optimal subset of all the candidates. Later the experiments show that the joint prediction achieves better performance than local systems.

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

In this section, we first introduce the ILP framework in Section 4.4.2.1, and we talk briefly about the local systems developed to assign local scores in Section 4.4.2.2. Finally we will present the experiment result in Section 4.4.2.3..

4.4.2.1 Integer Linear Programming Framework.

First, we extract two agent candidates and two theme candidates for each +/-effect event (one each will ultimately be chosen by the ILP model).[1] We use syntax, and the output of the SENNA [Collobert et al., 2011] semantic role labeling tool.

We use Integer Linear Programming (ILP) to assign labels to variables. Variables in *Entity* will be assigned *positive* or *negative*, representing the writer's sentiments toward them. Each variable in *EffectEvent* will be assigned the label *+effect* or *-effect*. Optionally, it may also be assigned the label *reversed*. Label *+effect* or *-effect* is the effect of the event; *reversed* is assigned if the effect is reversed (e.g., for "not harmed", the labels are *-effect* and *reversed*).

The objective function of the ILP is:

$$\min_{u_{1bf}\cdots}\left(-1 * \sum_{i \in \text{EffectEvent} \cup \text{Entity}} \sum_{c \in L_i} p_{ic} u_{ic}\right) + \sum_{\langle i,k,j \rangle \in \text{Triple}} \xi_{ikj} + \sum_{\langle i,k,j \rangle \in \text{Triple}} \delta_{ikj} \tag{4.5}$$

subject to

$$u_{ic} \in \{0, 1\}, \forall i, c \qquad \xi_{ikj}, \delta_{ikj} \in \{0, 1\}, \forall \langle i, k, j \rangle \in \text{Triple} \tag{4.6}$$

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

where L_i is the set of labels given to $\forall i \in$ *EffectEvent* $\cup$ *Entity*. If $i \in$ *EffectEvent*, L_i is {+effect, -effect, reversed} ({+effect, -effect, r}, for short). If $i \in$ *Entity*, L_i is {positive, negative} ({pos, neg}, for short). u_{ic} is a binary indicator representing whether the label c is assigned to the variable i. When an indicator variable is 1, the corresponding label is selected. p_{ic} is the score given by local detectors, introduced in the following sections. Variables ξ_{ikj} and δ_{ikj} are binary slack variables that correspond to the +/-effect inference rule constraints of $\langle i, k, j \rangle$. When a given slack variable is 1, the corresponding triple violates the inference rule constraints. Minimizing the objective function could achieve two goals at the same time. The first part ($-1 * \sum_i \sum_c p_{ic} u_{ic}$) tries to select a set of labels that maximize the scores given by the local detectors. The second part ($\sum_{ikj} \xi_{ikj} + \sum_{ikj} \delta_{ikj}$) aims at minimizing the cases where +/-effect inference rule constraints are violated. Here we do not force each triple to obey the inference constraints, but to minimize the violating cases.

According to the inference rules in Table 3 in Section 4.3, the writer has the same sentiment toward entities in a +effect event. Thus, for each triple unit $\langle i, k, j \rangle$, such constraint is applied via the following:

$$|\sum_{i,\langle i,k,j\rangle} u_{i,pos} - \sum_{j,\langle i,k,j\rangle} u_{j,pos}| + |u_{k,\text{+effect}} - u_{k,r}| <= 1 + \xi_{ikj}, \forall k \in \text{EffectEvent} \tag{4.7}$$

$$|\sum_{i,\langle i,k,j\rangle} u_{i,neg} - \sum_{j,\langle i,k,j\rangle} u_{j,neg}| + |u_{k,\text{+effect}} - u_{k,r}| <= 1 + \xi_{ikj}, \forall k \in \text{EffectEvent} \tag{4.8}$$

We use $|u_{k,\text{+effect}} - u_{k,r}|$ to represent whether this triple is +effect. In Equation (4.7), if this value is 1, then the triple should follow the +effect constraints. In that case, $\xi_{ikj} = 0$ means that the triple doesn't violate the +effect constraints, and $|\sum_i u_{i,pos} - \sum_j u_{j,pos}|$ must be 0. Further, in this case, $\sum_i u_{i,pos}$ and $\sum_j u_{j,pos}$ are constrained to be of the same value (both 1 or 0) – that is, entities i and j must be both positive or both not positive. However, if $\xi_{ikj} = 1$, Equation (4.7) does not constrain the values of the variables at all. If $|u_{k,\text{+effect}} - u_{k,r}|$ is 0, representing that the triple is not a +effect event, then Equation (4.7) does not constrain the values of the variables. Similar comments apply to Equation (4.8).

In contrast, the writer has opposite sentiments toward entities in a -effect event.

$$|\sum_{i,\langle i,k,j\rangle} u_{i,pos} + \sum_{j,\langle i,k,j\rangle} u_{j,pos} - 1| + |u_{k,\text{-effect}} - u_{k,r}| <= 1 + \delta_{ikj}, \forall k \in \text{EffectEvent} \tag{4.9}$$

$$|\sum_{i,\langle i,k,j\rangle} u_{i,neg} + \sum_{j,\langle i,k,j\rangle} u_{j,neg} - 1| + |u_{k,\text{-effect}} - u_{k,r}| <= 1 + \delta_{ikj}, \forall k \in \text{EffectEvent} \tag{4.10}$$

We use $|u_{k,\text{-effect}} - u_{k,r}|$ to represent whether this triple is a -effect event. In Equation (4.9), if a triple is a -effect event and the constraints are not violated, then $|\sum_i u_{i,pos} + \sum_j u_{j,pos} - 1|$ must be 0. Further, in this case, $\sum_i u_{i,pos}$ and $\sum_j u_{j,pos}$ are constrained to be of the opposite value – that is, if entity i is positive then entity j must not be positive. Similar comments apply to Equation (4.10).

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

+/-Effect Score: $p_{k,\textbf{+effect}}, p_{k,\textbf{-effect}}$**.** We utilize a sense-level +/-effect lexicon by [Choi et al., 2014]. In total there are 6,622 +effect senses and 3,290 -effect senses. The +effect lexicon covers 64% of the +effect words in the corpus and the -effect lexicon covers 42% of the -effect words. We then look up the +/-effect span k in the +/-effect lexicon. $p_{k,\text{+effect}}$ is the fraction of +effect senses out of all the senses of k, and $p_{k,\text{-effect}}$ is the fraction of -effect senses out of all the senses of k.

Local Reversed Score: $p_{k,r}$**.** We build reverser lexicons from Wilson's shifter lexicon [Wilson, 2008], namely the entries labeled as *genshifter*, *negation*, and *shiftneg*. There are 219 reversers in the entire corpus; 134 (61.19%) are instances of words in one of the two lexicons. We use this lexicon to extract reverser words as well as negations. The probability of a +/-effect event being reversed decreases as the length of the path increases. We define $p_{k,r}$ so it is inversely proportional to the length of the path.

Local Sentiment Score: $p_{i,pos}, p_{i,neg}$**.** We use Opinion Extractor [Johansson and Moschitti, 2013a] , opinionFinder [Wilson et al., 2005], MPQA subjectivity lexicon [Wilson et al., 2005], General Inquirer [Stone et al., 1966] and a connotation lexicon [Feng et al., 2013], to detect writer's sentiments toward all agent and theme candidates, and all +/-effect events. We adopt Rule Schema 1 and 3 to infer from the sentiment toward event to the sentiment toward theme. Then we conduct a majority voting based on the results. The sentiment scores range from 0.5 to 1.

4.4.2.3 Experiment and Result. We use the +/-effect event corpus in Section 4.2, consisting of 134 online editorials and blogs. In total, there are 1,762 annotated triples, out of which 692 are +effect or retainers and 1,070 are -effect or reversers. From the writer's perspective, 1,495 noun phrases are annotated positive, 1,114 noun phrases are negative and the remaining 8 are neutral. Out of 134 documents in the corpus, 3 do not have any annotation. 6 are used as a development set to develop the heuristics in Sections 4.4.2.2. We use the remaining 125 for the experiments.

We compare the output of the global optimization framework with the outputs of baseline systems built from the local detectors in Section 4.4.2.2 (denoted *Local*).

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

$$\text{P} = \frac{\#(\text{auto=gold \& gold!=neutral})}{\#\text{auto!=neutral}} \qquad \text{Accuracy} = \text{R} = \frac{\#(\text{auto=gold \& gold!=neutral})}{\#\text{gold!=neutral}} \tag{4.11}$$

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

		correct span subset			whole set, strict eval			whole set, relaxed eval		
		P	R	F	P	R	F	P	R	F
1	ILP	0.6421	0.6421	0.6421	0.4401	0.4401	0.4401	0.5939	0.5939	0.5939
2	Local	0.6409	0.3332	0.4384	0.4956	0.2891	0.3652	0.5983	0.3490	0.4408
3	Majority	0.5792	0.5792	0.5792	0.3862	0.3862	0.3862	0.5462	0.5462	0.5462

Table 6: Performances of sentiment detection

We report the performance results for **(A) sentiment detection** in Table 6, on two sets. One is the subset containing the agents and themes where *auto* has the correct spans with *gold*. The other is the set of all agents and themes. As shown in Table 6, *ILP* significantly improves performance on F-measure over different baselines. Though *Local* has a competitive precision with *ILP*, it has a much lower recall. That means the local sentiment detector cannot recognize implicit sentiments toward most entities. But *ILP* is able to recognize more entities correctly.

In terms of the other tasks: For **(B) agent/theme span**, the baseline achieves 66.67% in accuracy, compared to 68.54% for *ILP*. For **(C) +/-effect**, the baseline has an accuracy of 70.68%, whereas *ILP* achieves 77.25%. This improvement is interesting because it represents cases in which the optimization framework is able to ***infer*** the correct polarity even though the +/-effect span is not recognized by the local detector (i.e., the span isn't in the +/-effect lexicon). For **(D) reverser**, the baseline is 88.07% in accuracy. *ILP* is competitive with the baseline: 89%. Note that both our local detector and *ILP* surpass the majority class (not reversed) which has an accuracy of 86.60%.

4.5 SUMMARY

Attitude inferences arise from interactions between sentiment expressions and +/-effect events. This chapter investigates utilizing +/-effect event information to improve detection of the writer's sentiments toward entities mentioned in the text.

Based on the inference rules, we implement a graph-based model to propagate sentiments. In an evaluation of the model itself, we find it has an 89% chance of propagating sentiments correctly. However, that graph-based model in Section 4.4.1 requires all of the +/-effect event information to be input from the manual annotations; the only ambiguity it resolves is sentiments toward entities. In contrast, the Integer Linear Programming (ILP) framework in Section 4.4.2 tackles four ambiguities (1)-(4) simultaneously, incorporating inference rules as constraints. The ILP framework jointly infers the effect of events, whether or not they are reversed, which candidate NPs are the agent and theme, and the writer's sentiments toward them. In addition to beating the baselines for sentiment detection, the framework significantly improves the accuracy of disambiguating the effect of events.

5.0 EVENT/ENTITY-LEVEL SENTIMENT DETECTION AND INFERENCE

The ultimate goal of this thesis is to utilize the +/-effect events information and inference rules to improve detecting entity/event-level sentiments in the documents. There are ambiguities in each step of the whole task. We decompose this task into several subtasks, as shown in Figure 5. In this section, we illustrate what are the ambiguities in each subtask, then propose models aiming at detecting sentiments and solving the ambiguities in different subtasks simultaneously.

(1) The region in the blue circle in Figure 5 represents the +/-effect events and the agents and themes to be identified. The ambiguities come from: (1.1) Which spans are +/-effect events? (1.2) Which NPs are the agents, which are the themes? (1.3) What is the polarity of the +/-effect event? (1.4) Is the polarity reversed?

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

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

Though there are many ambiguities, they are interdependent. Inference rules define dependencies among these ambiguities. Recall that in Section 4.4.2, we have carried out the study of identifying the writer's sentiments toward the agents and themes of the +/-effect events [Deng et al., 2014] . In the study, we first develop traditional classifiers to

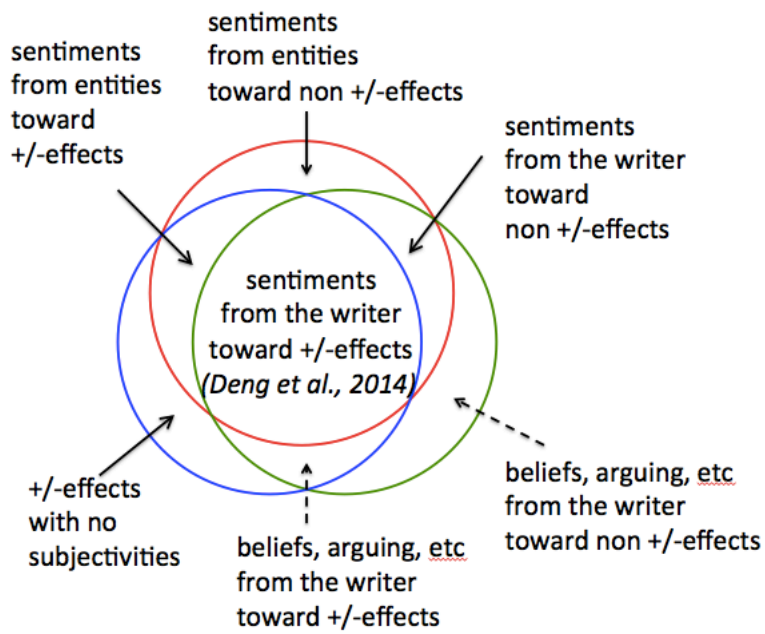

Figure 5: Overview of Subtasks.

generate the candidates of each ambiguity. Then we define each candidate as variables in an Integer Linear Programming (ILP) framework and incorporate four inference rules as constraints into the framework. The ILP framework simultaneously beat the baselines by more than 5 points in F-measure on sentiment detection and more than 7 points in accuracy on +/-effect polarity disambiguation, without any loss in performance on the other tasks: judging whether the +/-effect polarity is reversed and extracting the agents and themes.

The ILP experiment [Deng et al., 2014] corresponds to the intersection of the three regions in Figure 5. Though the ILP experiment only part of the whole task, the success of it encourages us to extend our work from the intersection to all the regions with *solid lines* pointed to: the sources of sentiments are not limited to only the writer but all entities , and the targets of sentiments are not only the agents and themes of +/-effect events, but are all entities and events, including other sentiments. With these extensions, there are many more ambiguities. The increased complexity added by the new ambiguities in the extended regions will be mitigated by more constraints defined by the inference rules we will introduce below.

We expect the full set of rules will help efficiently resolve more ambiguities, compared to the simplest version set of rules used in the ILP experiment [Deng et al., 2014]. Although the two regions with *dashed lines* pointed to are out of scope in this proposal, we can adopt the framework in this proposal to jointly analyze sentiments and beliefs in the future.

In this chapter, we first introduce how we model the entity/event-level sentiment analysis task in Section 5.1. The sentiment inference rules to infer sentiments toward general entities and events are introduced in Section 5.2. Since each inference rule is an implicature, meaning they are defeasible. In Section 5.3, we propose to investigate when the inference rules should be blocked. Based on the task description and the inference rules, we talk about the joint models we propose to jointly resolve the ambiguities listed above in Section 5.4. Finally we give the summary.

5.1 ENTITY/EVENT-LEVEL SENTIMENT ANALYSIS TASK DESCRIPTION

In this section, we introduce the definition of the entity/event-level sentiment analysis task, followed by a description of the gold standard corpus.

For each sentence s, we define a set E consisting of entities, events, and the writer of s, and sets P and N consisting of positive and negative sentiments, respectively. Each element in P is a tuple, representing a **positive pair** of two entities, (e_1, e_2) where e_1, $e_2 \in E$, and e_1 is positive toward e_2. A positive pair (e_1,e_2) aggregates all the positive sentiments from e_1 to e_2 in the sentence. N is the corresponding set for **negative pairs**.

The goal of this work is to automatically recognize a set of positive pairs (P_{auto}) and a set of negative pairs (N_{auto}). We compare the system output ($P_{\text{auto}} \cup N_{\text{auto}}$) against the gold standard ($P_{\text{gold}} \cup N_{\text{gold}}$) for each sentence.

5.2 SENTIMENT INFERENCE RULES

We adopt the inference rules developed in [Wiebe and Deng, 2014]. Let us go through an example inference for Ex(3.1) in Chapter 3, in particular, the inference that Imam is positive toward the Prophet. Recall the two explicit sentiments: Ex(3.2) shows Imam is negative toward the insulting sentiment (revealed by *issued the fatwa against*), and Ex(3.3) shows Rushdie is negative toward the Prophet (revealed by *insulting*). The inference is: since Imam is negative that there is any negative opinion expressed toward the Prophet, we infer that Imam is positive toward the Prophet. The inference rules are novel in that the target of a sentiment may be another sentiment (i.e., **sentiment toward sentiment** structure). For example, the target of Imam's negative sentiment is another negative sentiment,revealed by *insulting*. The inference rules link sentiments to sentiments and, transitively, link entities to entities (e.g., from Imam to Rushdie to the Prophet).

The rule supporting this inference is Rule 1.4 below, where "sentiment(-attitude) = negative" means that Imam is negative toward *insulting* attitude. And "sentiment(target) = positive" means that we infer Imam is positive toward the Prophet, which is the target of the *insulting* attitude. A full list in in Appendix B.

Rule 1.4: sentiment(-attitude) = negative → sentiment(target) = positive

Ultimately, each rule in this section and the rules w.r.t +/-effect events in Chapter 4 are implicatures [Wiebe and Deng, 2014]. Implicatures are defeasible, meaning that the inference may be blocked given outside evidence to the contrary [Greene and Resnik, 2009] and the inference should not be carried out. In the next section, we propose to investigate under which cases the rules should be blocked and how to automatically detect such cases.

5.3 BLOCKING THE RULES

Recall in Section 4.4.1, the graph-based model correctly propagates 89% of the sentiments expressed toward the agents and the themes of the +/-effect events, given a correct sentiment in the graph. In other words, there is 11% chance that the propagation fails. In this section,

we propose to investigate why the propagation fails and how we can recognize the cases when the propagation fails.

5.3.1 Data Collection

First, we need to collect the inference cases when the propagation fails. We can collect data from both +/-effect event corpus in Section 4.2. We can also collect data from the MPQA 3.0 corpus in Chapter 3.

For the +/-effect event corpus, we can run the graph-based model and record toward which agent or theme the sentiment label is wrongly propagated. The sentence containing that agent or theme and the +/-effect event can be used to study why the propagation fails.

For the entity/event-level sentiment corpus, MPQA 3.0, we can revise the graph model and run it on MPQA 3.0. Similarly, we can collect the sentences where the propagation fails. In Section 4.4.1, each node is either the agent or the theme participating in +/-effect events. Here we revise each node to be an entity participating in any event. In Section 4.4.1, the edge scores $\Psi_{ij}(\text{pos}, \text{pos})$ and $\Psi_{ij}(\text{neg}, \text{neg})$ are both 1 if the two nodes on the edge participate in a +effect event. This is because the sentiments toward the entities in a +effect event are supposed to be the same according to the rules. On the contrary, the edge scores $\Psi_{ij}(\text{pos}, \text{neg})$ and $\Psi_{ij}(\text{neg}, \text{pos})$ are both 1 if the two nodes on the edge participate in a -effect event. This is because the sentiments toward the entities in a -effect event are supposed to be opposite according to the rules. We can define the edge score for MPQA 3.0 in the similar way. Thus, if the sentiments toward the two entities in MPQA 3.0 are supposed to be the same according to the rules in both Section 4.3 and Section 5.2, an edge is created linking the two entities, and the edge scores $\Psi_{ij}(\text{pos}, \text{pos})$ and $\Psi_{ij}(\text{neg}, \text{neg})$ are defined as 1, while the edge scores $\Psi_{ij}(\text{pos}, \text{neg})$ and $\Psi_{ij}(\text{neg}, \text{pos})$ are defined as 0. If the sentiments are supposed to be opposite according to the rules, an edge is created linking the two entities, and the edge scores $\Psi_{ij}(\text{pos}, \text{pos})$ and $\Psi_{ij}(\text{neg}, \text{neg})$ are 0, while the edge scores $\Psi_{ij}(\text{pos}, \text{neg})$ and $\Psi_{ij}(\text{neg}, \text{pos})$ are 1. In this way, we can run the graph-based model on MPQA 3.0 corpus and collect the instances where the propagation fails.

5.3.2 Learning

After we collect data, we will category the instances and learn to recognize them. Wiebe and Deng [Wiebe and Deng, 2014] have observed one type of cases when the propagation fails. For example, when talking about +/-effect events, so far we only distinguish the effect of that event - whether it is +effect or -effect. However, inferences may not go through if the agent is **forced** to conduct the +/-effect event. Consider Table 7. In the first sentence, the writer is negative toward the private insurance companies. In the second sentence, according to the rules, the writer is positive toward the private insurance companies, which is wrong. The companies are not voluntary to stop eroding the pension, but they are forced to do so. Thus, if the agent is not voluntary to conduct the action, the inference should be blocked.

To facilitate observation and categorizing the cases when the inference rules should be blocked, we will choose a sentence when the propagation is wrong. We will try to extract any sentence when the propagation is correct containing the same +/-effect event. For example, the two sentences in Table 7 can be compared. We will investigate several methods of automatically finding the differences of the two sentences instead of purely observation by humans. For instance, we can compare the syntax parse tree, e.g., calculate the tree edit distance by finding what is inserted or what is deleted. Further, we can even try to align the two sentences and see which component cannot be aligned. In this way, we can categorize the reasons why the propagation is wrong in a faster way.

After we categorize the case why the propagations are wrong, we will develop classifiers to recognize each case. Traditional linguistic features will be investigated. For example, in recognize when the agent is being forced to conduct an +/-effect event, we will use word-level clues such as unigram "force" and its synonyms.

5.3.3 Evaluation

The graph model in Section 4.4.1 can be used to evaluate whether we have correctly recognized the cases when the rules should be blocked and the inference should be not carried out. Depending on what classifier we use to recognize whether to block the rules, we propose different methods to adjust the edge scores according to the classifier output.

The private insurance companies are eroding the valuable pension of the old all the time.	
⟨companies, erode, pension⟩ POSITIVE(writer, pension)	✓ POSITIVE(writer, pension) ⇒ NEGATIVE(writer, eroding) ✓ NEGATIVE(writer, eroding) ⇒ NEGATIVE(writer, companies)
Under the new passed bill, they will be forced to stop doing so. (*Interpreted as:*) The private insurance companies will stop eroding the valuable pension.	
⟨companies, stop eroding, pension⟩ POSITIVE(writer, pension)	✓ POSITIVE(writer, pension) ⇒ POSITIVE(writer, stop eroding) × POSITIVE(writer, stop eroding) ⇒ POSITIVE(writer, companies)

Table 7: An Exception to the Rules.

If it is a binary classifier, we do not generate an edge linking a and b. Thus, there is no sentiment propagation between a and b. If it is a numerical classifier, we use another method. Recall that the edge score being 1 indicates that the sentiments toward a and b should be the same, while the edge score being 0 indicates that the sentiments toward a and b should be opposite. As defined in Equations 4.1 and 4.2, if all the edge scores are 0.5, then the positive message transmitted equals the negative message transmitted and the propagation are offset Therefore, we adjust the edge score according to the classifier output. If the edge score is 1, it will be deducted the classifier output score. If the edge score is 0, it will be added the classifier output score. A large classifier output score indicates this case is probably should be blocked, and the score will force the edge scores moving closer to 0.5.

Then we run the graph model again to propagate sentiments. Ideally if we correctly recognize when the rules are blocked, the chance that a node is propagated with correct sentiment label should increase.

5.4 COMPUTATIONAL MODELS

At the beginning in this chapter, we have decomposed the whole task into three subtasks. In each subtask, there are different ambiguities. Recall that in Chapter 4, we have utilized an Integer Linear Programming framework, in which the candidates of each ambiguity are detected by local systems. Then the ILP framework simultaneously selects the best candidate w.r.t all the ambiguities and jointly resolves all the ambiguities. The experiment has shown that the joint model is better than individual systems. Thus, in this section, we follow the same procedure. First, we propose local systems to extract candidates of each kind of ambiguity in Section 5.4.1. Then, we propose joint models to resolve the ambiguities, where the inference rules we have developed are embedded as constraints of the joint model, in Section 5.4.2.

5.4.1 Local Systems

In this section, we propose to develop local systems to extract candidates of ambiguities in each subtask. Generally speaking, there are two tasks. The first is to analyze the +/-effect event information in Section 5.4.1.1. The second is to analyze the entity/event-level sentiments. As stated in Section 1, an opinion frame has three components: the source, the polarity and the target. We will talk about recognizing explicit sentiments in Section 5.4.1.2, recognizing the entity/event-level target in Section 5.4.1.3, and recognizing the sources in Section 5.4.1.4.

5.4.1.1 +/-Effect Events. We follow the individual systems developed in Chapter 4 to extract candidates of the ambiguities caused by +/-effect events, corresponding to the blue circle in Figure , including (1.1) Which spans are +/-effect events? (1.2) Which NPs are the agents, which are the themes? (1.3) What is the polarity of the +/-effect event? (1.4) Is the polarity reversed?

5.4.1.2 Explicit Sentiments. Instead of building an entity/event-level sentiment system from scratch, we propose to fully utilize off-the-shelf tools and resources for extracting opinions, the sources, and the targets [Yang and Cardie, 2013, Yang and Cardie, 2014, Johansson and Moschitti, 2013a, Riloff et al., 2013, Xu et al., 2013, Liu et al., 2014, Tang et al., 2014, Liu et al., 2013, Irsoy and Cardie, 2014, Scholz and Conrad, 2013, Zhou et al., 2013]. Some of the resources and tools extract the opinion expressions, the opinion polarities, the opinion sources, and the opinion targets. Some of the resources and tools only extract the opinion expressions and the polarities. Moreover, the sources and targets extracted by off-the-shelf tools are usually span-based. We will take the union of all the explicit opinions and the components of the opinions that state-of-art systems extract and use them as a basis for our sentiment inference.

5.4.1.3 Entity/Event-Level Target (ETarget) An eTarget is an entity or event (event includes states). Since each eTarget is the head of an noun phrase or a verb phrase. Each noun or verb could be an eTarget. However, that may result in too many eTarget candidates. We first narrow down the possible eTarget candidates. We can use semantic role labelling tools to extract the potential eTargets of opinions, since the eTargets are usually the semantic objects of an opinion expression, or the NP or VP that an opinion expression modifies. Also, we can use off-the-shelf Named Entity Recognizer (NER) to extract named entities as potential entity eTargets [Pan et al., 2015, Finkel et al., 2005, Nadeau and Sekine, 2007]. Currently there are resources extracting events and focus on the event coreference task [Li et al., 2013, Chen et al., 2009, Chen and Ji, 2009]. The pronouns should be taken into consideration as well.

To measure the probability of an eTarget candidate being the correct eTarget, we propose the features listed below.

Opinion Span Features. Several common features used to extract targets will be used, including Part-Of-Speech, path in the dependency parse graph, distance of the constituents on the parse tree, etc [Yang and Cardie, 2013, Yang and Cardie, 2014].

Target Span Features. Among the off-the-shelf systems and resources, some work extracts the target spans in addition to the opinions. We will investigate features depicting

the relations between a NP/VP head and the extracted target spans, such as whether the head overlaps with the target span. However, some off-the-shelf systems only extract the opinion spans, but do not extract any target span. For a NP/VP head, if the target span feature is false, there may be two reasons: (1) There is a target span extracted, but the target span feature is false (e.g. the head doesn't overlap with the target span). (2) There is no target span extracted by any tool at all.

Due to this fact, we propose three ways to define target span features. The simplest method (M1) is to assign zero to a false target span feature, regardless of the reason. A similar method (M2) is to assign different values (e.g. 0 or -1) to a false target span feature, according to the reason that causes the feature being false. For the third method (M3), we propose the Max-margin SVM [Chechik et al., 2008]. Unlike the case where a feature exists but its value is not observed or false, here this model focus on the case where a feature may not even exist (structurally absent) for some of the samples [Chechik et al., 2008]. In other words, the Max-margin SVM deals with features that are known to be non-existing, rather than have an unknown value. This allows us to fully utilize the different structures of outputs from different state-of-the-art resources.

5.4.1.4 Nested Source The method and the features to extract the sources of opinions are similar to the method and features used to extract eTargets. However, an important advantage of the MPQA corpus is that the source of an opinion is nested. Consider the example below.

When the Imam issued the fatwa against Salman Rushdie for insulting the Prophet

There is a negative sentiment in the sentence above, *issued the fatwa against*. The direct source of the negative sentiment is *Imam*, because it is Imam who issued the fatwa. According to the MPQA corpus, this negative sentiment is expressed only in this article. In the other articles, Imam may not always be negative toward Rushdie. This negative sentiment is claimed only in this article and only by the writer of this article. Thus, the negative sentiment is nested under the writer's private state. The source of this negative sentiment is (writer, Imam), which is also the annotation in MPQA 3.0. As introduced in

Section 3.1.1, in MPQA, the source (writer, Imam) is called nested source. According to Section 3.1.1, we call *writer* as the top level source of the nested source, and call *Imam* the immediate source.

Generally speaking, the opinions expressed in a document are nested under the writer's private state. Thus, the top level source is always the writer. There are cases where the nested source consists of more than the writer and the immediate source. Let's see another opinion from the example above. The negative opinion, *insulting*, is expressed toward *the Prophet*. The immediate source is *Rushdie*. Similarly, this negative sentiment is expressed in this document, i.e., according to the writer of this document. Thus, the top level of the nested source is writer. However, the writer doesn't directly states Rushdie being negative toward the Prophet. Instead, the writer quotes the Imam's action. Because the Imam believes Rushdie has insulted the Prophet, then the Imam thinks Rushdie is negative toward the Prophet so that the Imam issued the fatwa. Thus, Imam is part of the nested source of this negative sentiment. According to the annotation, the nested source is (writer, Imam, Rushdie).

The nested source annotation of MPQA corpus allows us to capture different levels of attributions. In order to automatically analyze the nested sources of opinions, we propose to investigate the following linguistic features. First, detecting the nested source can be modelled as attributing quotations. There is much work automatically attributing quotations [Pareti et al., 2013, de La Clergerie et al., 2011, Almeida et al., 2014]. Some representative phrases are strong indicators of nested source, such as *according to*, *he said*, etc. Second, the sources are nested because the target of an opinion is another opinion, such as the example we state above. Thus, when we detect an opinion in a target span, we can detect nested source.

5.4.2 Joint Model

We are pursing such a model that combines the probabilistic calculation of many ambiguities under the constraints of the dependencies of the data, defined by inference rules in the first order logic form. Every candidate of every ambiguity is represented as a variable in

the joint model. The goal is to find an optimal configuration of all the variables, thus the ambiguities are solved. Our previous study in Section 4.4.1 [Deng et al., 2014] and many previous work in various applications of NLP [Roth and Yih, 2004, Punyakanok et al., 2004, Punyakanok et al., 2008, Das et al., 2012, Choi et al., 2006, Yang and Cardie, 2013, Denis and Baldridge, Martins and Smith, 2009, Somasundaran and Wiebe, 2009], have used Integer Linear Programming (ILP) as a joint model to do so, by setting the dependencies as constraints in the ILP framework. However, the constraints in ILP are linear equations and inequations. In order to choose a framework that computes the first order logic directly, we propose the Markov Logic Network (MLN) [Richardson and Domingos, 2006] and its variations as an appropriate way to solve our problem.

The MLN is a framework for probabilistic logic that employ weighted formulas in first order logic to compactly encode complex undirected probabilistic graphical models (i.e., Markov networks) [Beltagy et al., 2014].

A MLN model is defined using a set of atoms to be grounded, and a set of weighted if-then rules expressed in first-order logic. For example, we define the atom eTarget(y,t) to represent an opinion y having eTarget t. If y and t are constants, then eTarget(y,t) is a ground atom (e.g., eTarget(insulting, Prophet)). Each ground atom is assigned a score by a local system. MLN takes as input all the local scores as well as the constraints defined by the rules among atoms, so that it is able to jointly resolve all the ambiguities. In the final output, for example, the score eTarget(insulting, Prophet) = 1 means that MLN considers Prophet to be an eTarget of *insulting*, while eTarget(insulting, countries) = 0 means that MLN does not consider *countries* to be an eTarget of *insulting*. The goal of the MLN is to find an optimal grounding which maximizes the values of all the satisfied first order logic formula in the knowledge base [Richardson and Domingos, 2006].

5.4.2.1 Atoms. Consistent with the task definition in Section 5.1, we define two atoms in MLN:

(1) PosPair(s,t): a positive pair from s toward t

(2) NegPair(s,t): a negative pair from s toward t

Both s and t are chosen from the set E. The values of ground atoms (1) and (2) are not

observed and are inferred by MLN.

Then, we define atoms to model an entity/event-level opinion:

(3) Pos(y): y is a positive sentiment

(4) Neg(y): y is a negative sentiment

(5) Source(y,s): the source of y is s

(6) eTarget(y,t): the eTarget of y is t

We define the following atoms to represent +/-effect events:

(7) +Effect(x): x is a +effect event

(8) -Effect(x): x is a -effect event

(9) Agent(x,a): the agent of x is a

(10) Theme(x,h): the theme of x is h

5.4.2.2 Inference Rules in First Order Logic We use the atoms defined above to express the inference rules in Section 4.3 and Section 5.2 in first order logic. Two examples of rules and representations are shown in Table 8. A full set of all the rules is listed in Appendix C.

5.4.3 Pilot Study

We have conducted a pilot study to jointly infer entity/event-level sentiments based on explicit sentiments, +/-effect event information and inference rules [Deng and Wiebe, 2015a]. The pilot study is conducted on the annotated MPQA 3.0 corpus in Section 3. In this section, we will first introduce how we build the local systems and joint model corresponding to Section 5.4.1 and Section 5.4.2. Then we will talk about the evaluation and the performances.

5.4.3.1 Local Systems. To build the local systems as described in Section 5.4.1, we use the +/-effect sense-level lexicon [Choi et al., 2014] to extract +/-effect events in Section 5.4.1.1, and use three span-based sentiment analysis system [Yang and Cardie, 2013,

The bill will lower the skyrocketing healthcare cost.
(THEME(x, y) $\wedge$ POLARITY(x, -*effect*)) $\Rightarrow$ (POSITIVE(s, x) $\Leftrightarrow$ NEGATIVE(s, y))
(THEME(lower, cost) $\wedge$ POLARITY(lower, -*effect*)) $\Rightarrow$ (POSITIVE(writer, lower) $\Leftrightarrow$ NEGATIVE(writer, cost))
Great! They like it.
(TARGET(x, y) $\wedge$ POLARITY(x, *positive*)) $\Rightarrow$ (POSITIVE(s, x) $\Leftrightarrow$ POSITIVE(s, y))
(TARGET(x, it) $\wedge$ POLARITY(x, *positive*)) $\Rightarrow$ (POSITIVE(writer, s) $\Leftrightarrow$ POSITIVE(writer, it))

Table 8: Examples and Inference Rules. In each box, line 1: sentence. Line 2: inference rule. Line 3: presenting the sentence in the rule.

Yang and Cardie, 2014, Socher et al., 2013] to extract the opinion expressions in Section 5.4.1.2. We use span-based sentiment analysis system to extract the target spans. All NP/VP heads of the target spans and the opinion spans are considered as eTarget candidates in Section 5.4.1.3. A SVM classifier is trained to assign scores to all the eTarget candidate. We use state-of-the art span-based systems to extract the most immediate source spans in Section 5.4.1.4.

5.4.3.2 Joint Model. To build the joint model as described in Section 5.4.2, we use the inference rules in Section 5.2, and the inference rules for inferring sentiments expressed toward +/-effect event information in Section 4.3. We use the atoms defined in Section 5.4.2. Instead of Markov Logic Network, we use its variation, Probabilistic Soft Logic (PSL) [Kimmig et al., 2012], to jointly infer entity/event-level sentiments. The atoms in MLN are boolean values, either 0 or 1. Instead of only being boolean value, the atoms in PSL could have numerical values. Given the predicates being numerical, PSL uses the *Lukasiewicz t-norm* and its corresponding co-norm to quantify the degree to which a grounding of the logic formula is satisfied [Kimmig et al., 2012].

$l_1 \wedge l_2 = \max\{0, I(l_1) + I(l_2) - 1\}$
$l_1 \vee l_2 = \min\{I(l_1) + I(l_2), 1\}$
$\neg l_1 = 1 - I(l_1)$

5.4.3.3 Baselines. Since each noun and verb may be an eTarget, the first baseline (All NP/VP) regards all the nouns and verbs as eTargets. The first baseline estimates the difficulty of this task.

The second baseline (SVM) uses the SVM local classification results, which are the score of ETARGET(y,t). Then it is normalized as input into PSL. Before normalization, if the score assigned by the SVM classifier is above 0, the SVM baseline considers it as a correct eTarget.

5.4.3.4 Evaluations. Two kinds of evaluations are carried out. First, we examine the performance of the PSL models on correctly recognizing eTargets of a particular opinion. This evaluation is carried out on a subset of the corpus: we only examine the opinions which are automatically extracted by the span-based systems. If an opinion expression in the gold standard is not extracted by any span-based system, it is not input into PSL, so PSL cannot possibly find its eTargets.

The second assesses performance of the PSL models on correctly extracting positive and negative pairs. Note that our sentiment analysis system has the capability, through inference, to recognize positive and negative pairs even if corresponding opinion expressions are not extracted. Thus, the second evaluation is carried out on the entire corpus.

ETargets of An Opinion. According to the gold standard in Chapter 3, each opinion has a set of eTargets. But not all eTargets are equally important. Thus, our first evaluation assesses the performance of extracting the most important eTarget. As introduced in Chapter 3, a span-based target annotation of an opinion in MPQA 2.0 captures the most important target this opinion is expressed toward. Thus, the head of the target span can be considered to be the most important eTarget of an opinion. We model this as a ranking problem to compare models. For an opinion y automatically extracted by a span-based system, both the SVM baseline and PSL assign scores to ETARGET(y,t). We rank the eTargets according to the scores. Because the ALL NP/VP baseline does not assign scores to the nouns and verbs, we do not compare with that baseline in this ranking experiment. We use the Precision@N

evaluation metric. If the top N eTargets of an opinion contain the head of target span, we consider it as a correct hit. The results are in Table 9.

	Prec@1	Prec@3	Prec@5
SVM	0.0370	0.0556	0.0820
PSL	0.5503	0.7434	0.8148

Table 9: Precision@N of Most Important ETarget.

Table 9 shows that SVM is poor at ranking the most important eTarget. This shows that SVM, which only uses local features, cannot distinguish the most important eTarget from the others. But the PSL model considers all the opinions, and can recognize a true negative even if it ranks high in the local results. The ability of PSL to rule out true negative candidates will be repeatedly shown in the later evaluations.

We not only evaluate the ability to recognize the most important eTarget of a particular opinion, we also evaluate the ability to extract all the eTargets of that opinion. The F-measure of SVM is 0.2043, while the F-measures of PSL is 0.3275, respectively. Correctly recognizing all the eTargets is difficult, but all the PSL models are better than the baseline.

Positive Pairs and Negative Pairs. Now we evaluate the performance in a stricter way. We compare automatically extracted sets of sentiment pairs: $P_{\text{auto}} = \{\textsc{PosPair}(s,t) > 0\}$ and $N_{\text{auto}} = \{\textsc{NegPair}(s,t) > 0\}$, against the gold standard sets P_{gold} and N_{gold}. Table 5.4.3.4 shows the accuracies.

	PosPair	NegPair
All NP/VP	0.1280	0.1654
SVM	0.0765	0.0670
PSL	0.4315	0.3892

Table 10: Accuracy comparing PSL models

As shown in Table 5.4.3.4, the low accuracy of baseline All NP/VP shows that entity/event-level sentiment analysis is a difficult task. Even the SVM baseline does not have good

accuracy. The baseline classifies the heads of target spans and opinion spans, which are extracted by state-of-the-art span-based sentiment analysis systems. This shows the results from span-based sentiment analysis systems do not provide enough accurate information for the more fine-grained entity/event-level sentiment analysis task. In contrast, PSL achieves much higher accuracy than the baselines. An important reason is that SVM uses a hard constraint to cut off many eTarget candidates, while the PSL models take the scores as soft constraints.

A more critical reason is due to the definition of accuracy: (TruePositive+TrueNegative)/All. A significant benefit of using PSL is correctly recognizing true negative eTarget candidates and eliminating them from the set.

Note that F-measure does not count true negatives. Precision is $\frac{\text{TruePositive}}{\text{TruePositive+FalsePositive}}$, and recall is $\frac{\text{TruePositive}}{\text{TruePositive+FalseNegative}}$; neither considers true negatives (TN). Thus, the biggest advantage of PSL models is to correctly rule out true negative eTargets. For the baselines, though the SVM baseline has higher precision, it eliminates so many eTarget candidates that the F-measure is not high.

	Precision	Recall	F-measure
	PosPair		
All NP/VP	0.1481	0.4857	0.2270
SVM	0.3791	0.0870	0.1415
PSL	0.1659	0.3523	0.2256
	NegPair		
All NP/VP	0.1824	0.6408	0.2840
SVM	0.3568	0.0761	0.1254
PSL	0.2586	0.4529	0.3292

Table 11: F-measure comparing PSL models.

5.5 SUMMARY

In this chapter, we propose to build upon state-of-the-art span-based sentiment analysis systems to perform entity/event-level sentiment analysis covering both explicit and implicit sentiments expressed among entities and events in text. The Markov Logic Network and its variations incorporating explicit sentiments, inference rules and +/-effect event information are proposed to jointly disambiguate the ambiguities sentiment analysis and +/-effect event extraction. Our pilot study shows that Probabilistic Soft Logic is able to improve over baseline accuracies in recognizing entity/event-level sentiments [Deng and Wiebe, 2015a]. Based on the pilot study, we propose to improve the joint model for entity/event-level sentiment analysis in several aspects. First, currently our pilot study uses all nouns and verbs as eTarget candidates. More sophisticated methods will be developed to extract eTarget candidates. Second, we propose to extract nested sources, while current study only analyses the most immediate source of the opinion expression. Third, we need to recognize when the inference rules should be blocked.

6.0 TIMELINE OF PROPOSED WORK

Date	Content	Deliverable results
Sep - Nov	Collecting Non-Expert Annotation in Chapter 3 in Section 3.3	MPQA 3.0 corpus
Nov - Jan	Extracting Nested Source and ETarget in Chapter 5 in Section 5.4	A System Extracting Nested Source Entity/Event-Level ETarget
Jan - Mar	Analysing Blocked Rules in Chapter 5 in Section 5.3	Improved Graph Model in Section 4.4.1 of Chapter 4 by Identifying Blocked Rules
Mar - May	An Improved Joint Model Integrating Improved Components Above	Journal Submitted About Entity/Event-Level Sentiment Inference
May - Aug	Thesis Writing	Thesis Ready for Defense
Aug - Dec	Thesis Revising	Completed Thesis

Table 12: Timeline of Proposed Work.

7.0 SUMMARY

We focus on entity/event-level sentiment detection and inference. The source of a sentiment is the writer or an entity in the text. The target of a sentiment is an entity or event. The sentiment is not necessarily expressed via a sentiment expression (explicit sentiments), but it can be inferred by inference rules (implicit sentiments). We propose to work in three aspects.

First, we propose to annotate an entity/event-level sentiment corpus, MPQA 3.0, which will be annotated with both explicit and implicit sentiments expressed among entities and events. A subset of the new corpus has been annotated by expert annotators and the agreement study result is good. To collect more annotations in a faster and cheaper way, we propose to crowd-source the annotations. An annotation scheme is proposed for non-expert annotators.

To detect both explicit and implicit sentiments, we use inference rules and incorporate the inference rules in computational models. As a start, we focus on one particular type of event: +/-effect event. We have developed a corpus annotated with +/-effect event information and the sentiments. Two computational models have been developed. One shows the inference rules can correctly propagate sentiments. The other improves analyzing the sentiments expressed toward the agents and themes of +/-effect events.

Ultimately, a joint model will be used to simultaneously resolve various ambiguities in an entity/event-level sentiment analysis task and improve inferring entity/event-level sentiment analysis. The inference rules will be used as constraints in the joint model. A pilot study has been conducted and improved accuracy of sentiment analysis. We propose to proceed over the pilot study by analyzing the nested-source, utilizing state-of-the-art entity and event extraction systems and recognizing when the rules should be blocked. A better entity/event-level sentiment analysis system will be developed.

APPENDIX A

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

Rule Schema 1: sentiment(+/-effect event) → sentiment(theme)

Rule 1.1: sentiment(+effect) = positive → sentiment(theme) = positive

Rule 1.2: sentiment(+effect) = negative → sentiment(theme) = negative

Rule 1.3: sentiment(-effect) = positive → sentiment(theme) = negative

Rule 1.4: sentiment(-effect) = negative → sentiment(theme) = positive

Rule Schema 2: sentiment(theme) → sentiment(+/-effect event)

Rule 2.1: sentiment(theme) = positive → sentiment(+effect) = positive

Rule 2.2: sentiment(theme) = negative → sentiment(+effect) = negative

Rule 2.3: sentiment(theme) = positive → sentiment(-effect) = negative

Rule 2.4: sentiment(theme) = negative → sentiment(-effect) = positive

Rule Schema 3: sentiment(+/-effect event) → sentiment(agent)

Rule 3.1: sentiment(+effect) = positive → sentiment(agent) = positive

Rule 3.2: sentiment(+effect) = negative → sentiment(agent) = negative

Rule 3.3: sentiment(-effect) = positive → sentiment(agent) = positive

Rule 3.4: sentiment(-effect) = negative → sentiment(agent) = negative

Rule Schema 4: sentiment(agent) → sentiment(+/-effect event)

Rule 4.1: sentiment(agent) = positive → sentiment(+effect) = positive

Rule 4.2: sentiment(agent) = negative $\rightarrow$ sentiment(+effect) = negative

Rule 4.3: sentiment(agent) = positive $\rightarrow$ sentiment(-effect) = positive

Rule 4.4: sentiment(agent) = negative $\rightarrow$ sentiment(-effect) = negative

APPENDIX B

SENTIMENT INFERENCE RULES

Rule Schema 1: sentiment(attitude) → sentiment(target)

Rule 1.1: sentiment(+attitude) = positive → sentiment(target) = positive

Rule 1.2: sentiment(+attitude) = negative → sentiment(target) = negative

Rule 1.3: sentiment(-attitude) = positive → sentiment(target) = negative

Rule 1.4: sentiment(-attitude) = negative → sentiment(target) = positive

Rule Schema 2: sentiment(attitude) → sentiment(source)

Rule 2.1: sentiment(+attitude) = positive → sentiment(source) = positive

Rule 2.2: sentiment(+attitude) = negative → sentiment(source) = negative

Rule 2.3: sentiment(-attitude) = positive → sentiment(source) = positive

Rule 2.4: sentiment(-attitude) = negative → sentiment(source) = negative

APPENDIX C

SENTIMENT INFERENCE RULES IN FIRST ORDER LOGIC

Part 1.	Aggregation Rules.	
1.1	$\textsc{Source}(y,s) \wedge \textsc{eTarget}(y,t) \wedge \textsc{Pos}(y)$	$\Rightarrow \textsc{PosPair}(s,t)$
1.2	$\textsc{Source}(y,s) \wedge \textsc{eTarget}(y,t) \wedge \textsc{Neg}(y)$	$\Rightarrow \textsc{NegPair}(s,t)$
Part 2.	**Inference Rules.**	
2.1	$\textsc{PosPair}(s_1,y_2) \wedge \textsc{Source}(y_2,s_2)$	$\Rightarrow \textsc{PosPair}(s_1,s_2)$
2.2	$\textsc{PosPair}(s_1,y_2) \wedge \textsc{eTarget}(y_2,t_2) \wedge \textsc{Pos}(y_2)$	$\Rightarrow \textsc{PosPair}(s_1,t_2)$
2.3	$\textsc{PosPair}(s_1,y_2) \wedge \textsc{eTarget}(y_2,t_2) \wedge \textsc{Neg}(y_2)$	$\Rightarrow \textsc{NegPair}(s_1,t_2)$
2.4	$\textsc{NegPair}(s_1,y_2) \wedge \textsc{Source}(y_2,s_2)$	$\Rightarrow \textsc{NegPair}(s_1,s_2)$
2.5	$\textsc{NegPair}(s_1,y_2) \wedge \textsc{eTarget}(y_2,t_2) \wedge \textsc{Pos}(y_2)$	$\Rightarrow \textsc{NegPair}(s_1,t_2)$
2.6	$\textsc{NegPair}(s_1,y_2) \wedge \textsc{eTarget}(y_2,t_2) \wedge \textsc{Neg}(y_2)$	$\Rightarrow \textsc{PosPair}(s_1,t_2)$
Part 3.	**Inference Rules w.r.t +/-Effect Event Information.**	
3.1	$\textsc{PosPair}(s,x) \wedge \textsc{Agent}(x,a)$	$\Rightarrow \textsc{PosPair}(s,a)$
3.2	$\textsc{PosPair}(s,x) \wedge \textsc{Theme}(x,h) \wedge +\textsc{Effect}(x)$	$\Rightarrow \textsc{PosPair}(s,h)$
3.3	$\textsc{PosPair}(s,x) \wedge \textsc{Theme}(x,h) \wedge \text{-}\textsc{Effect}(x)$	$\Rightarrow \textsc{NegPair}(s,h)$
3.4	$\textsc{NegPair}(s,x) \wedge \textsc{Agent}(x,a)$	$\Rightarrow \textsc{NegPair}(s,a)$
3.5	$\textsc{NegPair}(s,x) \wedge \textsc{Theme}(x,h) \wedge +\textsc{Effect}(x)$	$\Rightarrow \textsc{NegPair}(s,h)$
3.6	$\textsc{NegPair}(s,x) \wedge \textsc{Theme}(x,h) \wedge \text{-}\textsc{Effect}(x)$	$\Rightarrow \textsc{PosPair}(s,h)$
3.7	$\textsc{PosPair}(s,a) \wedge \textsc{Agent}(x,a)$	$\Rightarrow \textsc{PosPair}(s,x)$
3.8	$\textsc{PosPair}(s,h) \wedge \textsc{Theme}(x,h) \wedge +\textsc{Effect}(x)$	$\Rightarrow \textsc{PosPair}(s,x)$
3.9	$\textsc{PosPair}(s,h) \wedge \textsc{Theme}(x,h) \wedge \text{-}\textsc{Effect}(x)$	$\Rightarrow \textsc{NegPair}(s,x)$
3.10	$\textsc{NegPair}(s,a) \wedge \textsc{Agent}(x,a)$	$\Rightarrow \textsc{NegPair}(s,x)$
3.11	$\textsc{NegPair}(s,h) \wedge \textsc{Theme}(x,h) \wedge +\textsc{Effect}(x)$	$\Rightarrow \textsc{NegPair}(s,x)$
3.12	$\textsc{NegPair}(s,h) \wedge \textsc{Theme}(x,h) \wedge \text{-}\textsc{Effect}(x)$	$\Rightarrow \textsc{PosPair}(s,x)$

Table 13: Rules in First-Order Logic.

BIBLIOGRAPHY

[Akkaya et al., 2010] Akkaya, C., Conrad, A., Wiebe, J., and Mihalcea, R. (2010). Amazon mechanical turk for subjectivity word sense disambiguation. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 195–203. Association for Computational Linguistics.

[Almeida et al., 2014] Almeida, M. S., Almeida, M. B., and Martins, A. F. (2014). A joint model for quotation attribution and coreference resolution. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden*, pages 39–48.

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

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

[Chechik et al., 2008] Chechik, G., Heitz, G., Elidan, G., Abbeel, P., and Koller, D. (2008). Max-margin classification of data with absent features. *The Journal of Machine Learning Research*, 9:1–21.

[Chen and Ji, 2009] Chen, Z. and Ji, H. (2009). Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 54–57. Association for Computational Linguistics.

[Chen et al., 2009] Chen, Z., Ji, H., and Haralick, R. (2009). A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the Workshop on Events in Emerging Text Types*, pages 17–22. Association for Computational Linguistics.

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

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

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

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

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

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

[Dawid and Skene, 1979] Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28.

[de La Clergerie et al., 2011] de La Clergerie, É., Sagot, B., Stern, R., Denis, P., Recourcé, G., and Mignot, V. (2011). Extracting and visualizing quotations from news wires. In *Human Language Technology. Challenges for Computer Science and Linguistics*, pages 522–532. Springer.

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

[Deng and Wiebe, 2014] Deng, L. and Wiebe, J. (2014). Sentiment propagation via implicature constraints. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 377–385, Gothenburg, Sweden. Association for Computational Linguistics.

[Deng and Wiebe, 2015a] Deng, L. and Wiebe, J. (2015a). Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal. Association for Computational Linguistics.

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

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

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

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

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

[Goyal et al., 2012] Goyal, A., Riloff, E., and III, H. D. (2012). A computational model for plot units. *Computational Intelligence*, pages no–no.

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

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

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

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

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

[Kimmig et al., 2012] Kimmig, A., Bach, S., Broecheler, M., Huang, B., and Getoor, L. (2012). A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, pages 1–4.

[Li et al., 2013] Li, Q., Ji, H., and Huang, L. (2013). Joint event extraction via structured prediction with global features. In *ACL (1)*, pages 73–82.

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

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

[Martins and Smith, 2009] Martins, A. F. T. and Smith, N. a. (2009). Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing - ILP '09*, pages 1–9, Morristown, NJ, USA. Association for Computational Linguistics.

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

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

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

[Nadeau and Sekine, 2007] Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

[Pan et al., 2015] Pan, X., Cassidy, T., Hermjakob, U., Ji, H., and Knight, K. (2015). Unsupervised entity linking with abstract meaning representation.

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

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

[Pearl, 1982] Pearl, J. (1982). Reverend bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the American Association of Artificial Intelligence National Conference on AI*, pages 133–136, Pittsburgh, PA.

[Punyakanok et al., 2008] Punyakanok, V., Roth, D., and Yih, W.-t. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

[Punyakanok et al., 2004] Punyakanok, V., Roth, D., Yih, W.-t., and Zimak, D. (2004). Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1346. Association for Computational Linguistics.

[Reschke and Anand, 2011] Reschke, K. and Anand, P. (2011). Extracting contextual evaluativity. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pages 370–374, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Richardson and Domingos, 2006] Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine learning*, 62(1-2):107–136.

[Riloff et al., 2013] Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.

[Roth and Yih, 2004] Roth, D. and Yih, W.-t. (2004). A linear programming formulation for global inference in natural language tasks. In *CONLL*.

[Scholz and Conrad, 2013] Scholz, T. and Conrad, S. (2013). Opinion mining in newspaper articles by entropy-based word connections. In *EMNLP*, pages 1828–1839.

[Socher et al., 2013] Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Citeseer.

[Somasundaran and Wiebe, 2009] Somasundaran, S. and Wiebe, J. (2009). Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 226–234, Suntec, Singapore. Association for Computational Linguistics.

[Somasundaran et al., 2007] Somasundaran, S., Wilson, T., Wiebe, J., and Stoyanov, V. (2007). Qa with attitude: Exploiting opinion type analysis for improving question answering in on-line discussions and the news. In *International Conference on Weblogs and Social Media*, Boulder, CO.

[Stone et al., 1966] Stone, P., Dunphy, D., Smith, M., and Ogilvie, D. (1966). *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge.

[Stoyanov et al., 2005] Stoyanov, V., Cardie, C., and Wiebe, J. (2005). Multi-perspective question answering using the opqa corpus. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 923–930. Association for Computational Linguistics.

[Tang et al., 2014] Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565.

[Titov and McDonald, 2008] Titov, I. and McDonald, R. T. (2008). A joint model of text and aspect ratings for sentiment summarization. In *ACL*, volume 8, pages 308–316. Citeseer.

[Turney, 2002] Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

[Welinder and Perona, 2010] Welinder, P. and Perona, P. (2010). Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 25–32. IEEE.

[Wiebe and Deng, 2014] Wiebe, J. and Deng, L. (2014). An account of opinion implicatures. *arXiv*, 1404.6491[cs.CL].

[Wiebe et al., 2005] Wiebe, J., Wilson, T., and Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2/3):164–210.

[Wilson, 2008] Wilson, T. (2008). *Fine-grained subjectivity analysis*. PhD thesis, Doctoral Dissertation, University of Pittsburgh.

[Wilson et al., 2005] Wilson, T., Wiebe, J., , and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *HLP/EMNLP*, pages 347–354.

[Wilson and Wiebe, 2003] Wilson, T. and Wiebe, J. (2003). Annotating opinions in the world press. In *Proceedings of the 4th ACL SIGdial Workshop on Discourse and Dialogue (SIGdial-03)*, pages 13–22.

[Xu et al., 2013] Xu, L., Liu, K., Lai, S., Chen, Y., and Zhao, J. (2013). Mining opinion words and opinion targets in a two-stage framework. In *ACL (1)*, pages 1764–1773.

[Yang and Cardie, 2013] Yang, B. and Cardie, C. (2013). Joint Inference for Fine-grained Opinion Extraction. In *Proceedings of ACL*, pages 1640–1649.

[Yang and Cardie, 2014] Yang, B. and Cardie, C. (2014). Joint modeling of opinion expression extraction and attribute classification. *Transactions of the Association for Computational Linguistics*, 2:505–516.

[Yedidia et al., 2005] Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312.

[Yu and Hatzivassiloglou, 2003] Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136. Association for Computational Linguistics.

[Zhang and Liu, 2011] Zhang, L. and Liu, B. (2011). Identifying noun product features that imply opinions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 575–580, Portland, Oregon, USA. Association for Computational Linguistics.

[Zhou et al., 2013] Zhou, X., Wan, X., and Xiao, J. (2013). Collective opinion target extraction in chinese microblogs. In *EMNLP*, pages 1840–1850.