INFORMATION EXTRACTION OF +/-EFFECT EVENTS TO SUPPORT OPINION INFERENCE

by

Yoonjung Choi

B.E., Korea Advanced Institute of Science and Technology, 2007

M.S., Korea Advanced Institute of Science and Technology, 2010

Submitted to the Graduate Faculty of

the Kenneth Dietrich School of Arts and Sciences

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Yoonjung Choi

It was defended on

August 29th 2016

and approved by

Janyce Wiebe, PhD, Professor, Department of Computer Science

Diane Litman, PhD, Professor, Department of Computer Science

Milos Hauskrecht, PhD, Associate Professor, Department of Computer Science

Rebecca Hwa, PhD, Associate Professor, Department of Computer Science

Tessa Warren, PhD, Associate Professor, Department of Linguistics

Dissertation Advisors: Janyce Wiebe, PhD, Professor, Department of Computer Science,

Diane Litman, PhD, Professor, Department of Computer Science

INFORMATION EXTRACTION OF +/-EFFECT EVENTS TO SUPPORT OPINION INFERENCE

Yoonjung Choi, PhD

University of Pittsburgh, 2016

Recently, work in NLP was initiated on a type of opinion inference that arises when opinions are expressed toward events which have positive or negative effects on entities, called *+/-effect events*. The ultimate goal is to develop a fully automatic system capable of recognizing inferred attitudes. To achieve its results, the inference system requires all instances of +/-effect events. Therefore, this dissertation focuses on +/-effect events to support opinion inference. To extract +/-effect events, we first need the list of +/-effect events. Due to significant sense ambiguity, our goal is to develop a sense-level rather than word-level lexicon. To handle sense-level information, WordNet is adopted. We adopt a graph-based method which is seeded by entries culled from FrameNet and then expanded by exploiting semantic relations in WordNet. We show that WordNet relations are useful for the polarity propagation in the graph model. In addition, to maximize the effectiveness of different types of information, we combine a graph-based method using WordNet relations and a standard classifier using gloss information. Further, we provide evidence that the model is an effective way to guide manual annotation to find +/-effect senses that are not in the seed set. To exploit the sense-level lexicons, we have to carry out word sense disambiguation. We present a knowledge-based +/-effect coarse-grained word sense disambiguation method based on selectional preferences via topic models. For more information, we first group senses, and then utilize topic models to model selectional preferences. Our experiments show that selectional preferences are helpful in our work. To support opinion inferences, we need to identify not only +/-effect events but also their affected entities automatically. Thus, we address both

+/-effect event detection and affected entity identification. Since +/-effect events and their affected entities are closely related, instead of a pipeline system, we present a joint model to extract +/-effect events and their affected entities simultaneously. We demonstrate that our joint model is promising to extract +/-effect events and their affected entities jointly.

Keywords: Sentiment Analysis, Implicit Opinion, Opinion Inference, Lexical Acquisition, Word Sense Disambiguation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

1.0 INTRODUCTION

Opinions are commonly expressed in many kinds of written and spoken text such as blogs, reviews, new articles, discussions, and tweets. **Sentiment Analysis** is the computational study to identify opinions, evaluations, attitues, affects, and emotions expressed in such texts [Liu, 2010]. There are many names and tasks with somewhat different objectives and models such as opinion mining, sentiment mining, subjectivity analysis, affect analysis, emotion detection, and so on.

Here is the example of reviews presented by [Liu, 2010]:

(a) I bought an iPhone a few days ago. (b) It was such a nice phone. (c) The touch screen was really cool. (d) The voice quality was clear too. (e) Although the battery life was not long, that is ok for me. (f) However, my mother was mad with me as I did not tell her before I bought the phone. (g) She also thought the phone was too expensive, and wanted me to return it to the shop.

In this example, the sentence (a) has no sentiment while others have sentiment information. We can say that the sentence (a) is the objective sentence because it presents some factual information; others are subjective sentences because they express some personal feelings, views, emotions, or beliefs. In addition, each sentence except the sentence (a) has different sentiment information. In the sentence (b), the writer has a positive sentiment toward an *iPhone*. Also, in the sentence (c) and (d), the writer has a positive sentiment toward attributes (i.e., *the touch screen* and *the voice quality*) of the *iPhone*. On the other hand, in the sentence (g), the writer's mother has a negative sentiment toward the *iPhone*. As such, we can see lots of sentiment information in a text.

Recently, there have been a surge in research in sentiment analysis. It has been exploited in many application areas such as review mining, election analysis, and information extraction. Especially, with growing interest in social media such as Facebook, Twitter, and blogs, which contain various opinionated user contents, sentiment analysis has become increasingly important because it can be applied for a variety of applications such as opinion summarization, opinion spam detection, advertisement, and customized recommendation.

Sentiment analysis consists of three subtasks. The basic subtask is classifying the opinion orientation (i.e., polarity) of a given text at the document/sentence/phrase/aspect-level. That is, it determines whether the expressed opinion in the given text is positive, negative, or neutral. For instance, the sentence (b), (c), and (d) in the previous example should be classified into positive sentences while the sentence (f) and (g) should be classified into negative sentences. In the sentence (e), even though the polarity of the sentence is neutral, in the phrase-level classification, *the battery life* should be classified into a negative phrase.

Other subtasks are the opinion holder detection which extracts the person or organization that expresses the opinion and the opinion target extraction which identifies objects or their aspects on which opinions are expressed. For example, in the sentence (b), the opinion holder is a writer and the opinion target is *iPhone*; in the sentence (c), (d), and (e), the opinion holder is a writer and the opinion target is *the touch screen*, *the voice quality*, and *the battery life* which are attributes of *iPhone*. On the other hand, in the sentence (f), the opinion holder is a writer's mother (i.e., *my mother*) and the opinion target is a writer (i.e., *me*); in the sentence (g), the opinion holder is a writer's mother (i.e., *She*) and the opinion target is *the phone*, which indicates *iPhone*.

Thus, in other words, sentiment analysis aims to determine the opinion orientation of an opinion holder with respect to an opinion target. There are various studies for sentiment analysis with different research topics (i.e., document/sentence/phrase/aspect-level polarity classification [Pang et al., 2002, Pang and Lee, 2005, Riloff et al., 2005, Wilson et al., 2004, Mei et al., 2007], opinion holder and target identification [Kim and Hovy, 2006], and sentiment lexicon construction [Kim and Hovy, 2004, Baccianella et al., 2010]) and various domains such as review texts [Turney, 2002], new articles [Wilson et al., 2005], blog data [Godbole et al., 2007], and tweets [Barbosa and Feng, 2010].

There are two types of opinions: explicit opinion and implicit opinion. The explicit opinion means that an opinion toward an opinion target is explicitly expressed by an opinion holder in a given text. The example (1) is one of example of explicit opinions.

(1) *The voice quality of this phone is fantastic.*

In this example, the opinion toward the target, *the voice quality of this phone*, is explicitly expressed with a word, *fantastic*, which is the key clue to determine an opinion orientation. These words or expressions which are used to express peoples subjective feelings and sentiments/opinions are called as **sentiment lexicon**. (It is also known as polarity words, opinion words, or opinion-bearing words.) Here are examples of positive and negative terms. Not just individual words but also phrases and idioms can be the sentiment lexicon such as *cost an arm and a leg*. That is, the explicit opinion is expressed with clues such as sentiment lexicon.

Positive terms: *wonderful*, *elegant*, *amazing*
Negative terms: *horrible*, *bad*

On the other hand, the implicit opinion means that an opinion toward an opinion target is implicitly expressed by an opinion holder in a given text. In the example (2), although it doesn't express an opinion explicitly, we can know that a writer has a negative opinion toward the entity, *the headset*.

(2) *The headset broke in two days.*

Still, research in sentiment analysis has plateaued at a somewhat superficial level, providing methods that exhibit a fairly shallow understanding of subjective language as a whole. In particular, past research in NLP has mainly addressed explicit opinion expressions [Pang et al., 2002, Turney, 2002, Hu and Liu, 2004, Kim and Hovy, 2004, Wilson et al., 2005, Mei et al., 2007, Davidov et al., 2010, Barbosa and Feng, 2010], ignoring implicit opinions expressed via implicatures, i.e., default inferences.

Recently, to determine implicit opinions, Wiebe and Deng [Wiebe and Deng, 2014] address a type of opinion inference that arises when opinions are expressed toward events which have positive or negative effects on entities. They call such events benefactive and malefactive, or, for ease of writing, goodFor and badFor events. While the term goodFor/badFor is used in their paper, we have decided that +/-effect is a better term. Thus, in this research, we call such events **+/-effect events** instead of goodFor/badFor.

[Deng and Wiebe, 2014] show how sentiments toward one entity may be propagated to other entities via opinion inference rules. They give the following example:

(3) *The bill would curb skyrocketing health care costs.*

The writer expresses an explicit **negative** sentiment (by *skyrocketing*) toward the entity, *health care costs*. The existing sentiment analysis system can determine it. However, the existing explicit sentiment analysis system cannot determine the sentiment toward *the bill*. With opinion inference rules, not only the sentiment toward *health care costs* but also the sentiment toward *the bill* can be inferred. The event, *curb*, has a **negative effect** (i.e., -effect) on *skyrocketing health care costs*, since they are reduced. We can reason that the writer is **positive** toward the **event** because it has a negative effect on *costs*, toward which the writer is negative. From there, we can reason that the writer is **positive** toward *the bill*, since it conducts the positive event.

Now, consider the another example:

(4) *Oh no! The voters passed the bill.*

Here, the writer expresses an explicit **negative** sentiment toward the **passing event** because of *Oh no!*. Although we cannot know the sentiment toward *the bill* with the existing sentiment analysis system, we can infer it with opinion inference rules. The passing event is a **positive effect** (i.e., +effect) on *the bill* since it brings into existence. Since the writer is **negative** toward an **event** that benefits *the bill*, we can infer that the writer is **negative** toward *the bill* itself.

The ultimate goal is to develop a fully automatic system capable of recognizing such inferred attitudes. The system will require a set of implicature rules and an inference mechanism. [Deng and Wiebe, 2014] present a graph-based model in which inference is achieved via propagation. They show that such inferences may be exploited to significantly improve explicit sentiment analysis systems.

To achieve its results, the inference system requires all instances of +/-effect events. However, the system developed by [Deng and Wiebe, 2014] takes manual annotations as input; that is, it is not fully automatic system. The ultimate system needs to recognize a span of +/-effect events and their polarities (i.e., +effect, -effect, or Null) automatically. For that, we first need the list of +/-effect events. Although there are similar lexicons such as SentiWordNet [Esuli and Sebastiani, 2006] and connotation lexicons [Feng et al., 2011, Kang et al., 2014], sentiment, connotation, and +/-effects are not the same.

Moreover, the information about which entities are affected is important since the sentiment toward an entity can be different. In the example (3), as we mentioned, the given event, *curb*, is -effect on the theme (i.e., the affected entity is the theme), and the writer's sentiment toward the theme is negative. Thus, we know that the writer has a positive sentiment toward the event, and the sentiment toward the agent is positive.

Consider the following example:

(5) *Yay! John's team lost the first game.*

We know that the writer expresses an explicit **positive** sentiment toward the **event** because of *Yay!*. The event, *lost*, has a **negative effect** (i.e., -effect) on the entity, *John's team*, since it fails to win. That is, the affected entity is the agent, not the theme. We can infer that the writer has **negative** sentiment toward *John's team* because the event, that the writer is positive, has a **negative** effect on *John's team*. Compared to the sentence (3), even though both are -effect events and the writer has a **positive** sentiment toward these events, the sentiment toward the agent (i.e., *the bill* in the example (3) and *John's team* in the example (5)) is different according to what the affected entity is. Such as these examples, it is important to know which entities are affected by the event in opinion inferences.

As we mentioned, for the opinion inference system to be fully automatic, +/-effect event extraction also must be fully automated. At this time, we have to consider which entities are affected by +/-effect events since the sentiment toward an entity can be different. Thus, the goal of this research is to **develop resources and methods for information extraction of a general class of events, +/-effect events, which are critical for detecting implicit sentiment and which are also important for other tasks such as narrative understanding**.

1.1 RESEARCH SUMMARY

As we mentioned, to recognize a span of +/-effect events and their polarities (i.e., +effect, -effect, or Null) automatically, we first need the list of +/-effect events. Since +/-effect lexicon is the new types of lexicons, there is not available resource for +/-effect events. Thus, we first have to create a +/-effect lexicon.

One task of this dissertation is to build +/-effect lexicons. Since a word can have one or more meanings, the +/-effect polarity of a word may not be consistent. We discover that there is significant sense ambiguity, meaning that words often have mixtures of senses among the classes *+effect*, *-effect*, and *Null*.

In the +/-effect[1] corpus [Deng et al., 2013], +/-effect events and their agents and themes are annotated at the word-level. In this corpus, 1,411 +/-effect instances are annotated; 196 different +effect words and 286 different -effect words appear in these instances. Among them, 10 words appear in both +effect and -effect instances, accounting for 9.07% of all annotated instances. They show that +/-effect events (and the inferences that motivate this work) appear frequently in sentences with explicit sentiment. Further, **all** instances of +/-effect words that are **not** identified as +/-effect events are false hits from the perspective of a recognition system.

[1] Called the goodFor/badFor in this corpus.

The following is an example of a word with senses of different classes:

carry:

S: (v) carry (win in an election) "The senator carried his home state"

⇒ **+Effect toward the agent**, *the senator*

S: (v) carry (keep up with financial support) "The Federal Government carried the province for many years"

⇒ **+Effect toward the theme**, *the province*

S: (v) carry (capture after a fight) "The troops carried the town after a brief fight"

⇒ **-Effect toward the theme**, *the town*

In the first sense, *carry* has **positive** polarity toward the agent, *the senator*, and in the second case, it has **positive** polarity toward the theme, *the province*. Even though the polarity is the same, the affected entity is different. That is, in the first sense, the affected entity is the agent while the affected entity is the theme in the second sense. In the third sense, *carry* has **negative** polarity toward the theme, *the town*, since it is captured by the troops. Moreover, although a word may not have both +effect and -effect senses, it may have mixtures of (+effect or -effect) and Null. Consider *pass*.

pass:

S: (v) legislate, pass (make laws, bills, etc. or bring into effect by legislation)

⇒ **+Effect toward the theme**

S: (v) travel by, pass by, surpass, go past, go by, pass (move past)

⇒ **Null**

The meaning of *pass* in the example (4) is the first sense, in fact, +effect toward its theme. But consider the following example:

(6) *Oh no! They passed the bridge.*

In this case, the meaning of *pass* is the second sense. This type of passing event does not (in itself) positively or negatively affect the thing passed. This use of *pass* does not warrant the inference that the writer is negative toward the bridge.

A purely word-based approach is blind to these cases. Thus, to handle these ambiguities, we firstly develop a sense-level +/-effect lexicon. There are several resources with sense information such as WordNet and FrameNet. WordNet [Miller et al., 1990] is a computational lexicon of English based on psycholinguistic principles. Nouns, verbs, adjectives, and adverbs are organized by semantic relations between senses (synsets). There are several types of semantic relations such as hyponym, hypernym, troponym, and so on. Also, each sense has gloss information which consists of a definition and optional examples. FrameNet [Baker et al., 1998] is a lexical database of English based on a theory of meaning called Frame Semantics. In general, WordNet can cover more senses since it is a large database that groups words together based on their meanings. Moreover, senses in WordNet are interlinked by semantic relations which may be useful information to acquire +/-effect events. Thus, for +/-effect lexicon acquisition, we adopt WordNet which is a widely-used lexical resource. We first explore how +/-effect events are organized in WordNet via semantic relations and expand the seed set based on those semantic relations using a bootstrapping method.

One of our goals is to investigate whether the +/-effect property tends to be shared among semantically-related senses, and another is to use a method that applies to all word senses, not just to the senses of words in a given word-level lexicon. Thus, we build a graph-based model in which each node is a WordNet synset, and edges represent semantic WordNet relations between synsets. In addition, we hypothesize that glosses also contain useful information. Thus, we develop a supervised gloss classifier and define a hybrid model which gives the best overall performance. Moreover, we provide evidence that the graph-based model is an effective way to guide manual annotation to find new +/-effect senses.

Based on the constructed +/-effect lexicon, we can extract +/-effect events from a given text. If the constructed lexicon is a word-level lexicon, events can be determined directly; however, the constructed lexicon is a sense-level lexicon. Thus, to extract +/-effect events with a sense-level lexicon, we have to carry out Word Sense Disambiguation (WSD) to find specific senses.

In this dissertation, we develop a WSD system which is customized for +/-effect events. We address the following WSD task: given +/-effect labels of *senses*, determine whether an instance of a word in the corpus is being used with a +effect, -effect, or Null sense. Consider a word W, where senses $\{S_1, S_3, S_7\}$ are -effect; $\{S_2\}$ is +effect; and $\{S_4, S_5, S_6\}$ are Null. For our purposes, we do not need to perform fine-grained WSD to pinpoint the exact sense; to recognize that an instance of W is -effect, for example, the system only needs to recognize that W is being used with *one* of senses $\{S_1, S_3, S_7\}$. Thus, we can perform coarse-grained WSD, which is often more tractable than fine-grained WSD.

Though supervised WSD is generally the most accurate method, we do not pursue a supervised approach, because the amount of available sense-tagged data is limited. Instead, we conduct a knowledge-based WSD method which exploits WordNet relations and glosses. We use sense-tagged data (i.e., *SensEval*) only as gold-standard data for evaluation.

Our WSD method is based on *selectional preferences*, which are preferences of verbs to co-occur with certain types of arguments [Resnik, 1996, Rooth et al., 1999, Van de Cruys, 2014]. We hypothesize that preferences would be fruitful for our task, because +/-effect is a semantic property that involves affected entities. Consider the following WordNet information for *climb*:

climb:

S_1: (v) climb, climb up, mount, go up (go upward with gradual or continuous progress) "Did you ever climb up the hill behind your house?"

⇒ **Null**

S_2: (v) wax, mount, climb, rise (go up or advance) "Sales were climbing after prices were lowered"

⇒ **+Effect toward the theme**

S_3: (v) climb (slope upward) "The path climbed all the way to the top of the hill"

⇒ **Null**

S_4: (v) rise, go up, climb (increase in value or to a higher point) "prices climbed steeply"; "the value of our house rose sharply last year"

⇒ **+Effect toward the theme**

Senses S_1 & S_3 are both Null. We expect them to co-occur with *hill* and similar words such as *ridge* and *mountain*. And, we expect such words to be more likely to co-occur with S_1 & S_3 than with S_2 & S_4. Senses S_2 & S_4 are both +effect, since the affected entities are increased. We expect them to co-occur with *sales*, *prices*, and words similar to them. And, we expect such words to be more likely to co-occur with S_2 & S_4 than with S_1 & S_3. This example illustrates the motivation for using selectional preferences for +/-effect WSD.

We model sense-level selectional preferences using Topic Models, specifically Latent Dirichlet Allocation (LDA) [Blei et al., 2003]. We utilize LDA for modeling relations between sense groups and their arguments, and then carry out coarse-grained +/-effect WSD by comparing the topic distributions of a word instance and candidate sense groups and choosing the sense group which has the highest similarity value.

To support inference, not only +/-effect event information but also the information about which entity is affected is important since the sentiment toward an entity can be different. As we mentioned, in the example (3) and (5), even though both are -effect events and the writer has a positive sentiment toward these events, the sentiment toward the agent is different according to what the affected entity of the given event is. In the example (3), because the affected entity of the given event, *curb*, is a theme, the writer's sentiment toward the agent is positive by the inference. On the other hand, in the example (5), the writer has negative sentiment toward the agent because the given event, *lost*, is -effect event on the agent. Such as these examples, it is important to know which entity is affected by a given event in opinion inferences.

In this dissertation, for opinion inferences, we also address the affected entity identification. The +/-effect event detection and the affected entity identification might be regarded as independent tasks, so they can be placed in a pipeline system such as firstly detecting +/-effect events and then identifying their affected entities. [Deng et al., 2014] includes such approach. They simply check the presence of +/-effect words in a word-level lexicon (not a sense-level lexicon) for the +/-effect event detection, and they adopt the semantic role labeler and generate simple rules to identify affected entities.

However, we hypothesize that there are dependencies between +/-effect events and their affected entities. As [Choi and Wiebe, 2014, Choi et al., 2014] mentioned, since words can have a mixture of +effect, -effect and Null, it is important to grasp the meaning of the given word. So, contexts, especially affected entities, are important information to detect +/-effect events. For example, in the sentence (5), because the affected entity is *John's team*, we can know the meaning of *lost* is *to fail to win* which is a -effect event. On the other hand, to identify the affected entity, +/-effect event information is also important. For instance, in the sentence (3), the affected entity is *health care costs*, which is the theme of the event, *curb*. However, in the sentence (5), since the event is *lost*, the affected entity is *John's team*, which is the agent of the event, not *the first game* (which is the theme of the event). Thus, the +/-effect events and the affected entity can help each other.

Therefore, we propose a joint model to extract both +/-effect events and their affected entities. There are several works to successfully adopt a joint model in NLP tasks such as joint text and aspects ratings for sentiment summarization [Titov and McDonald, 2008], joint parsing and named entity recognition [Finkel and Manning, 2009], joint word sense disambiguation and semantic role labeling [Che and Liu, 2010], and joint event and entity extraction [Li et al., 2013, Li and Ji, 2014]. [Deng and Wiebe, 2015] also presents the joint prediction model using probabilistic soft logic models to recognize both explicit and implicit sentiments toward entities and events in the text. For implicit sentiments, they extract +/-effect events and their agents and themes. However, as we mentioned, depending on +/-effect events and contexts, an affected entity can be different (e.g., while an affected entity is a theme in the sentence (3), an agent is an affected entity in the sentence (5)). Thus, the important information is which entity is affected by the given event. We focus on the affected entity, not an agent and a theme. In addition, we suggest lexical or syntactic relations between +/-effect events and their affected entities, which they don't consider.

We adopt the structured perceptron suggested by [Collins, 2002] for a joint model. Structured perceptron is a machine learning algorithm for structured prediction problem. Since our input (i.e., a sentence) has structures and our output (i.e., +/-effect events and their affected entities) also has structures such as sequences and trees, we hypothesize that the approach for the structured prediction is appropriate for our task.

1.2 CONTRIBUTIONS OF THIS WORK

The research in this dissertation contributes to the opinion inference system which is to extract implicit opinions. The main contribution is the study of +/-effect events, which is critical for detecting implicit sentiment and which are also important for other tasks such as narrative understanding.

Ours is the first NLP research into developing a lexicon for events that have positive or negative effects on entities (i.e., +/-effect). We first present that +/-effect events have substantial sense ambiguity; that is, some words have mixtures of +effect, -effect, and Null. Due to significant sense ambiguity, we need a sense-level approach to acquire +/-effect lexicon knowledge, leading us to employ lexical resources with fine-grained sense rather than word representations. In this research, we adopt WordNet which is widely-used lexical resource since WordNet can cover more words and senses than other resources and it also contains all possible senses of given words. Moreover, WordNet provides a synonym set, called synsets, and synsets are interlinked by semantic relations which are useful information to acquire +/-effect events. We first present the feasibility of using WordNet for +/-effect lexicon acquisition with a bootstrapping method. We explore how +/-effect events are organized in WordNet via semantic relations and expand the seed set based on those semantic relations. We present that WordNet is promising for expanding sense-level +/-effect lexicons.

Then, we investigate methods for creating a sense-level +/-effect lexicon, called Effect-WordNet. We utilize WordNet resource with two assumptions: (1) each sense (or synset) has only one +/-effect polarity and (2) +/-effect polarity tends to propagate by semantic relations such as hierarchical information. One of our goals is to develop the method that applied to many verb synsets. Also, another goal is to build a lexicon with a small number of seed data. In addition, we want to investigate whether the +/-effect property tends to be shared among semantically-related synsets. We adopt a graph-based learning method for WordNet relations and show that WordNet relations can be used for the polarity propagation with a small number of seed data. Moreover, we build a standard classifier with bag-of-word features and sentiment features for gloss information. In addition, to maximize the effectiveness of different types of information, we combine a graph-based method using

WordNet relations and a classifier using gloss information. With the hybrid method, all senses in WordNet can be labeled with a small number of seed data. We provide evidence for our assumption that different models are needed for different information to maximize effectiveness. Further, we provide evidence that the model is an effective way to guide manual annotation to find +/-effect senses that are not in the seed set.

Moreover, we construct the enhanced sense-level +/-effect lexicon. The information about which entities are affected is important since the sentiment can be different in opinion inferences. Thus, we refine EffectWordNet with consideration of affected entities, called Enhanced EffectWordNet. We adopt a graph-based method such as the previous work. We represent that considering the information about which entities are affected is helpful to construct more refined sense-level +/-effect lexicon.

To extract +/-effect events with a constructed sense-level lexicon, we have to carry out Word Sense Disambiguation (WSD). Thus, we investigate +/-effect WSD approach, which identifies the +/-effect of a word sense based on its surrounding context. We develop a knowledge-based coarse-grained WSD which has large coverage without any sense-tagged training data. Our WSD method is based on selectional preferences, which are preferences of verbs to co-occur with certain types of arguments. Selectional preferences are modeled using a topic model. We show that selectional preferences are very helpful in our work since +/-effect is a semantic property that by its nature involves affected entities. Moreover, we present that a coarse-grained WSD approach is more appropriate for our work than a fine-grained WSD approach.

In addition, we conduct a pilot study to extract +/-effect events and their affected entities. We hypothesize that there are inter-dependencies between +/-effect events and their affected entities. Thus, we suggest a joint model to extract both +/-effect events and their affected entities. Since our input (i.e., a sentence) has structures and our output (i.e., +/-effect events and their affected entities) also has structures such as sequences and trees, we hypothesize that the approach for the structured prediction is appropriate for our task. Therefore, we adopt the structured perceptron and present several features for the +/-effect event detection and the affected entity identification. We show that our joint model is promising to extract +/-effect events and their affected entities jointly.

1.3 OUTLINE

In the remainder of this dissertation, Chapter 2 provides the background knowledge on NLP resources such WordNet and FrameNet which are utilized in our research and some machine learning methods which are adopted in our research. Then, we present the general information about sentiment analysis in Chapter 3. Chapter 4 introduces opinion inferences briefly and explains +/-effect events which are the main part in our research. In Chapter 5, we present the feasibility of using WordNet for +/-effect events. Chapter 6 presents a method to acquire +/-effect lexicon (called EffectWordNet) and Chapter 7 describes Enhanced EffectWordNet with consideration of affected entities. Then, Chapter 8 presents the word sense disambiguation method for sense-level +effect events. As we described, the affected entity information is also important for +/-effect events. In Chapter 9, we describes the joint extraction method to identify both +/-effect events and their affected entity. Finally, we summarize our research and discuss future work in Chapter 10.

2.0 BACKGROUND

In this chapter, we introduce two lexical resources used in this dissertation: WordNet and FrameNet. Both are widely used in research related *Natural Language Processing* (NLP). In Section 2.1, we first explain the concept of word senses and introduce WordNet resource. In Section 2.2, we describe the concept of frames and FrameNet. Finally, in Section 2.3, we explain machine learning methods utilized in this dissertation.

2.1 WORD SENSE AND WORDNET

In linguistics, a word **sense** is one of meanings of a word. Some words have only one meaning, that is, one sense. We say these are monosemous. However, words can have more than one meaning. Sometimes, these meanings of a word may be related to each other; we say these are polysemous. For instance, a noun *mouth* has two meanings such as "an organ of the body" and "the entrance of a cave" but they are related. On the other hand, a word may have entirely different meanings; called homonymous. For instance, a noun *skate* has two different meanings such as "sports equipment" and "the kind of fish".

The following is an example of a word with multiple senses:

bank:

S: (n) bank (sloping land (especially the slope beside a body of water)) "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"

S: (n) depository financial institution, bank, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) "he cashed a check at the bank"; "that bank holds the mortgage on my home"

S: (n) bank, bank building (a building in which the business of banking transacted) "the bank is on the corner of Nassau and Witherspoon"

In this example, the first sense and the second sense are homonymous since they are completely different meaning. On the other hand, the second sense and the third sense are polysemous because they are related to each other. Since the meaning of a word is important in NLP, we have to handle polysemous and homonymous cases such as the given example. For that, we first need a sense inventory such as a dictionary.

WordNet [Miller et al., 1990] is one sense inventory which is widely used in NLP. It is a computational lexicon of English based on psycholinguistic principles for English. It considers nouns, verbs, adjectives, and adverbs (and ignores others such as prepositions). Words are grouped into sets of cognitive synonyms, called **synsets**. Each synset expresses a distinct concept, that is, words in the same synset are interchangeable. Synsets provide not only words but also a short definition and one or more usage examples, called gloss information. Moreover, synsets are interlinked by means of conceptual-semantic and lexical relations. There are several relations for each lexical category (some are shared by lexical categories, but some are not):

Nouns:

- Hypernym: The generic term used to designate a whole class of specific instances. Y is a hypernym of X if X is a (kind of) Y.

- Hyponym: The specific term used to designate a member of a class. X is a hyponym of Y if X is a (kind of) Y.

- Meronym: The name of a constituent part of, the substance of, or a member of something. X is a meronym of Y if X is a part of Y.

- Holonym: The name of the whole of which the meronym names a part. Y is a holonym of X if X is a part of Y.

Verbs:

- Hypernym

- Troponym: A verb expressing a specific manner elaboration of another verb. X is a troponym of Y if to X is to Y in some manner.

- Entailment: A verb X entails Y if X cannot be done unless Y is, or has been, done.

- Groups: Verb senses that similar in meaning and have been manually grouped together.

Adjectives and **Adverbs**:

- Antonym: A pair of words between which there is an associative bond resulting from their frequent co-occurrence.

- Pertainym: Adjectives that are pertainyms are usually defined by such phrases as "of or pertaining to" and do not have antonyms. A pertainym can point to a noun or another pertainym.

Figure 1 shows the example *like* in WordNet. It presents several senses for each lexical category: two senses as a noun, five senses as a verb, and four senses as an adjective. Each sense of a word is in a different synset *S*. As we mentioned, each synset contains words, a short definition, and usage examples. For instance, in the first synset of *like* as a verb, it includes interchangeable words (i.e., *wish, care, like*), a short definition in parentheses (i.e., *prefer or wish to do something*), and one or more usage examples with quotation marks (i.e., *"Do you care to try this dish?"; "Would you like to come along to the movies?"*). Moreover, it provides several relations such as troponym in a verb, hypernym in a verb, and antonym in an adjective.

WordNet has been used for several NLP tasks such as word-sense disambiguation, machine translation, information retrieval, question answering, and information extraction because of its availability and coverage. WordNet contains more than 150,000 words organized in more than 100,000 synsets. In this research, we utilize WordNet 3.0[1].

[1]Available at http://wordnet.princeton.edu/

Noun

- S: (n) **like**, the like, the likes of (a similar kind) *"dogs, foxes, and the like"; "we don't want the likes of you around here"*
- S: (n) **like**, ilk (a kind of person) *"We'll not see his like again"; "I can't tolerate people of his ilk"*

Verb

- S: (v) wish, care, **like** (prefer or wish to do something) *"Do you care to try this dish?"; "Would you like to come along to the movies?"*
  - *direct troponym* / *full troponym*
    - S: (v) please (be the will of or have the will (to)) *"he could do many things if he pleased"*
  - *direct hypernym* / *inherited hypernym* / *sister term*
    - S: (v) desire, want (feel or have a desire for; want strongly) *"I want to go home now"; "I want my own room"*
  - *derivationally related form*
  - *sentence frame*
- S: (v) **like** (find enjoyable or agreeable) *"I like jogging"; "She likes to read Russian novels"*
- S: (v) **like** (be fond of) *"I like my nephews"*
- S: (v) **like** (feel about or towards; consider, evaluate, or regard) *"How did you like the President's speech last night?"*
- S: (v) **like** (want to have) *"I'd like a beer now!"*

Adjective

- S: (adj) **like**, similar (resembling or similar; having the same or some of the same characteristics; often used in combination) *"suits of like design"; "a limited circle of like minds"; "members of the cat family have like dispositions"; "as like as two peas in a pod"; "doglike devotion"; "a dreamlike quality"*
- S: (adj) **like**, same (equal in amount or value) *"like amounts"; "equivalent amounts"; "the same amount"; "gave one six blows and the other a like number"; "the same number"*
  - *see also*
  - ***antonym***
    - W: (adj) unlike [Opposed to: like] (not equal in amount) *"they distributed unlike (or unequal) sums to the various charities"*
- S: (adj) alike, similar, **like** (having the same or similar characteristics) *"all politicians are alike"; "they looked utterly alike"; "friends are generally alike in background and taste"*
- S: (adj) comparable, corresponding, **like** (conforming in every respect) *"boxes with corresponding dimensions"; "the like period of the preceding year"*

Figure 1: The example *like* in WordNet.

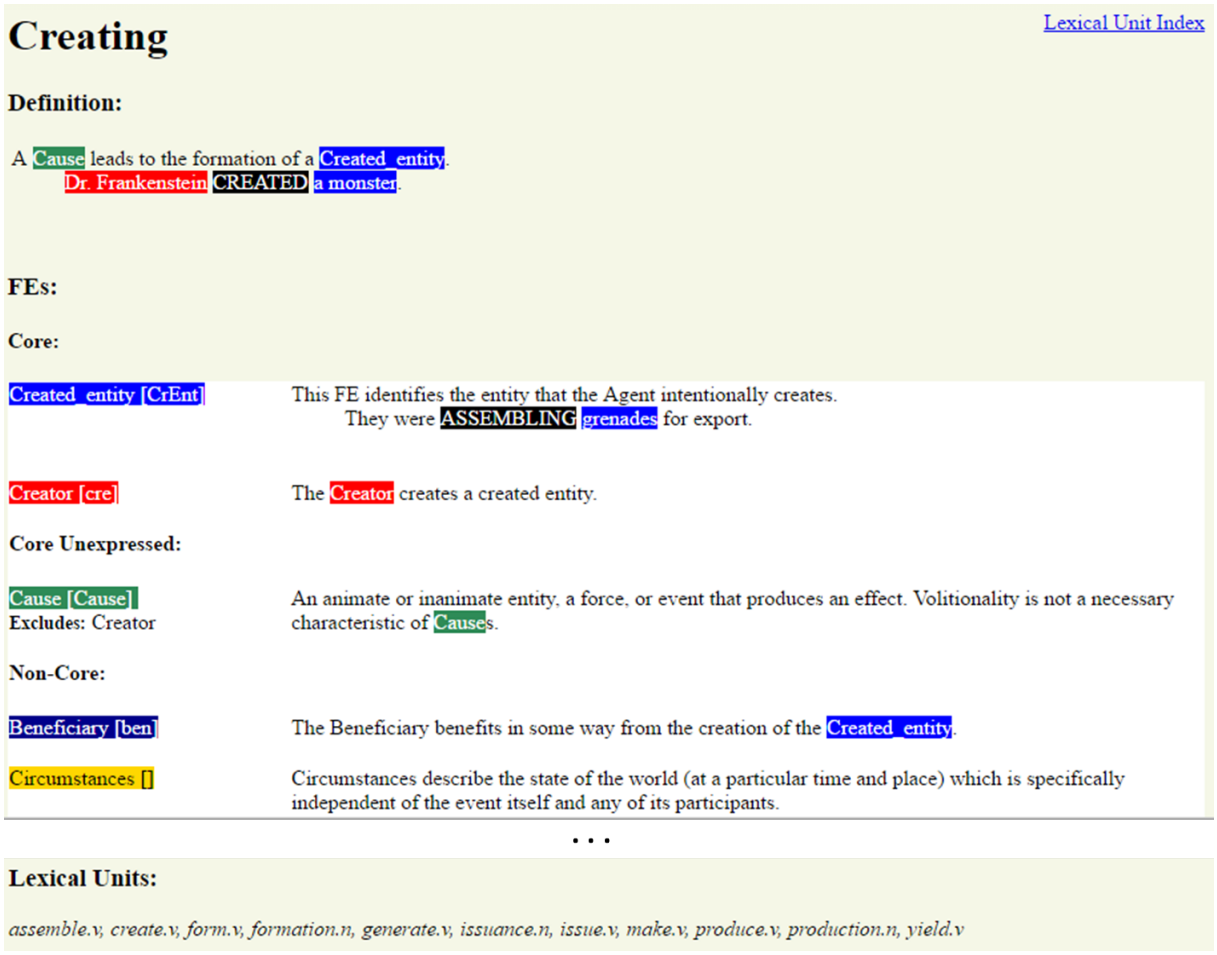
2.2 FRAMENET

FrameNet [Baker et al., 1998] is a lexical database of English containing more than 10,000 word senses with annotated examples. FrameNet is based on a theory of meaning called Frame Semantics which is developed by Charles J. Fillmore [Fillmore, 1977]. The basic idea is that the meanings of words can be understood based on a **semantic frame** such as a description of a type of event, relation, entity, and the participants in it. In FrameNet, a lexical unit is a pairing of a word with a meaning, i.e., it corresponds to a synset in WordNet.

For instance, the concept of creating involves a person or an entity to create something (i.e., Creator) and an entity that is created (i.e., Created_entity). Also, additional elements such as components to create an entity, a place where a creator creates an entity, and a purpose for which a creator creates an entity can be involved depending on a context. In FrameNet, The concept of creating is represented as a semantic frame called Creating and related elements such as Creator and Created_entity are called frame elements. For each semantic frame, they provide a definition of each frame, possible frame elements, and the list of lexical units. Figure 2 shows the semantic frame *Creating*. The definition of *Creating* frame is that a Cause leads to the formation of a Created_entity. It has two core frame elements such as Created_entity and Creator, and several additional frame elements such as Beneficiary, Circumstances, and so on. In addition, this frame contains 10 lexical units such as *assemble*, *create*, and so on.

The lexical entry of lexical unit is derived from annotations. Each lexical entry includes an associated frame and its frame elements with annotated example sentences. Figure 3 shows the example *create* of the *Creating* frame. It consists of a short definition of the lexical unit (i.e., *bring into existence*) and possible frame elements such as Created_entity and Creator. Then, there are several annotated example sentences such as *She had CREATED it from the chaos*. In each sentence, frame elements (represented by a color in the figure) are annotated; in the first sentence, *She* is the Creator, *it* is the Created_entity, and *from the chaos* is the Components.

The FrameNet database contains about 1,200 semantic frames, about 13,000 lexical units, and more than 190,000 annotated example sentences.



Creating

Lexical Unit Index

Definition:

A Cause leads to the formation of a Created_entity.
Dr. Frankenstein CREATED a monster.

FEs:

Core:

Created_entity [CrEnt]	This FE identifies the entity that the Agent intentionally creates. They were ASSEMBLING grenades for export.
Creator [cre]	The Creator creates a created entity.

Core Unexpressed:

Cause [Cause] **Excludes:** Creator	An animate or inanimate entity, a force, or event that produces an effect. Volitionality is not a necessary characteristic of Causes.

Non-Core:

Beneficiary [ben]	The Beneficiary benefits in some way from the creation of the Created_entity.
Circumstances []	Circumstances describe the state of the world (at a particular time and place) which is specifically independent of the event itself and any of its participants.

. . .

Lexical Units:

assemble.v, create.v, form.v, formation.n, generate.v, issuance.n, issue.v, make.v, produce.v, production.n, yield.v

Figure 2: The example semantic frame *Creating* in FrameNet.

create.v

Frame: Creating

Definition:

COD: bring into existence.

Frame Elements and Their Syntactic Realizations

The Frame Elements for this word sense are (with realizations):

Frame Element	Number Annotated	Realization(s)
Beneficiary	(1)	PP[for].Dep (1)
Cause	(15)	CNI.-- (2) INI.-- (1) NP.Ext (10) PP[by].Dep (2)
Co-participant	(1)	PP[with].Dep (1)
Components	(7)	PP[from].Dep (5) PP[in].Dep (1) PP[with].Dep (1)
Created_entity	(76)	NP.Obj (55) NP.Ext (15) DNI.-- (1) N.Head (2) PP[as].Dep (1) Sfin.Dep (2)
Creator	(62)	NP.Ext (39) INI.-- (1) CNI.-- (16)

Clear Sentences Turn Colors Off

[X] She had CREATED it from the chaos , she was its God .
[X] He toyed with the idea of CREATING a little angst in her life to slow her down .
[X] I 'm part of the workforce which is CREATING the direct change in my society .
[X] When he stood back , he could see that he had CREATED a superb landscape with figures .
[X] He CREATES his mood with two factors : harmony and rhythm .

Figure 3: The example lexical entry *create* of the *Creating* frame in FrameNet.

2.3 MACHINE LEARNING METHODS

In this research, we adopt three kinds of machine learning methods with different purposes. In Section 2.3.1, we first explain graph-based semi-supervised learning, which is used for sense-level lexicon acquisition. We describe topic models, which is utilized for coarse-grained word sense disambiguation, in Section 2.3.2. Then, structured prediction that is adopted for the joint extraction is briefly explained in Section 2.3.3.

2.3.1 Graph-based Semi-Supervised Learning

Semi-supervised learning falls between supervised learning, which requires labeled training data, and unsupervised learning, which do not need any labeled training data. Typically, a small number of training data is labeled while a relatively large number of training data is unlabeled. Since the training data contains unlabeled data, semi-supervised learning algorithms make one or more of the following assumptions [Subramanya and Talukdar, 2014]:

- Smoothness Assumption: If two points are close to each other, their outputs (i.e., labels) are also close.
- Cluster Assumption: If two points are in the same cluster, they are more likely to share a label.
- Manifold Assumption: The data lie approximately on a manifold of much lower dimension than the input space.

Among various semi-supervised learning algorithms, graph-based learning algorithms have received much attention recently due to their good performance and ease of implementation [Liu et al., 2012]. In graph-based semi-supervised learning, each labeled and unlabeled data is represented by a node in a graph and edges between these nodes can be built based on the similarity between the corresponding pairs. After constructing a graph, with seed data which is a small number of labeled nodes, we can predict the labels of the unlabeled nodes via

graph partition or information propagation. There are several graph-based semi-supervised learning algorithms such as graph cuts [Blum and Chawla, 2001, Blum et al., 2004], graph-based random walks [Azran, 2007], manifold regularization [Belkin et al., 2006], and graph transduction [Zhou et al., 2004, Zhu et al., 2003].

There are several reasons why graph-based semi-supervised learning algorithms are very attractive in our research. Firstly, synsets in WordNet which is the important resource in our research can be represented by a graph via semantic and lexical relations. As we mentioned, it only needs a small number of labeled data as seed data, so it doesn't require lots of human power for annotation works. In addition, as [Subramanya and Talukdar, 2014] mentioned, graph-based semi-supervised learning algorithms are effective in practice. [Subramanya and Bilmes, 2008] present that graph-based semi-supervised learning algorithms outperform other semi-supervised learning algorithms and supervised learning algorithms.

2.3.2 Topic Model

The topic model is based on the key idea that documents are mixtures of latent topics, where a topic is a probability distribution over words [Steyvers and Griffiths, 2007]. Each document may concern multiple topics in different proportions. For instance, there is a document that is 80% about sports and 20% about foods. Then, the given document would probably be four times more sport-related words than food-related words. A topic model captures this intuition.

The early topic model, Probabilistic Latent Semantic Analysis (pLSA), is presented by [Hofmann, 1999]. Each word is generated from a topic, and different words in the document may be generated from different topics; and each document is represented as a list of mixing proportion of different topics. Figure 4 presents the pLSA model. PLSA models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions such as:

$$P(d, w) = P(d) \sum_{z} P(w|z)P(z|d) \tag{2.1}$$

where d is a document, z is a topic, and w is a word. That is, for each document d, a topic z is chosen from a multinomial conditioned on d (i.e., from $P(z|d)$) and a word w is chosen from a multinomial conditioned on z (i.e., from $P(w|z)$).

Even though this model allows multiple topics in each document, pLSA doesn't make any assumptions about how the mixture weights θ are generated. Moreover, number of latent topics to learn grows linearly with the growth of the number of documents [Bao, 2012].

Thus, [Blei et al., 2003] extend pLSA model by adding Dirichlet priors to parameters for more reasonable mixtures of topics in a document. This model is called as Latent Dirichlet Allocation (LDA). Figure 5 shows the graphical LDA model where D is the number of documents, N_d is the number of words in document d, K is the number of topics, α is the parameter of the Dirichlet prior on the per-document topic distributions, β is the parameter of the Dirichlet prior on the per-topic word distribution, θ_d is the topic distribution for document d, ϕ_t is the word distribution for topic t, $z_{d,n}$ is the topic for n-th word in document d, and $w_{d,n}$ is the n-th word in document d (i.e., the observed word).

The generative process is as follows:

1. Choose $\theta_d \sim Dir(\alpha)$, where $d \in D$ and $Dir(\alpha)$ is the Dirichlet distribution for parameter α.
2. Choose $\phi_t \sim Dir(\beta)$, where $t \in K$.
3. For each of the word positions (d, n) where $d \in D$ and $n \in N_d$
   a. Draw a topic $z_{d,n} \sim \theta_d$
   b. Draw a word $w_{d,n} \sim \phi_t$

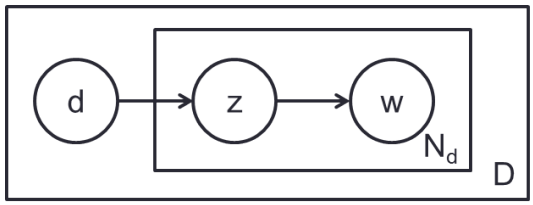


Figure 4: The graphical Probabilistic Latent Semantic Analysis (pLSA) Model.

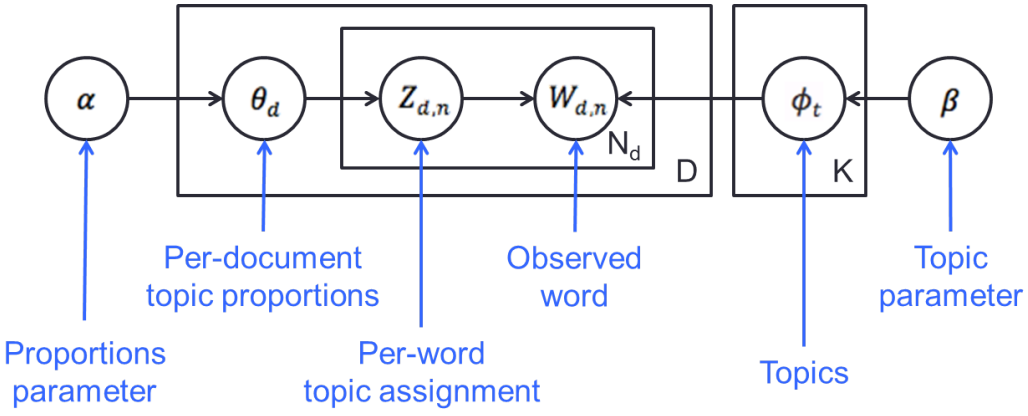


Figure 5: The graphical Latent Dirichlet Allocation (LDA) Model.

Thus, topic modeling can be used for discovery of hidden semantic structures (e.g., hidden topics) in a text. In our research, we assume that the selectional preference information is useful for our +/-effect word sense disambiguation task. The selectional preference information is hidden information such as hidden topics. Thus, we adopt a topic model to capture selectional preference.

2.3.3 Structured Prediction

Structured Prediction is machine learning techniques that involve predicting structured outputs. There are many tasks that output is represented by some structures such as sequences, trees, or graphs, especially, in NLP. For example, the Part-Of-Speech (POS) tagging task is to produce a sequence of POS tags for a given input sequence. The parsing task is another example since it builds a tree to represent some grammar for a given input sequence. In addition, there are many other NLP tasks related structured prediction such as entity detection, machine translation, and question answering.

While these tasks can be solved by independent classification of each word, this approach can not consider neighbors (i.e., contexts). The context is an important clue to resolve ambiguity. For instance, as you see in Figure 1, *like* can be a noun, a verb, and an adjective. Also, for each lexical category, there are several meanings. To disambiguate these words, the context information is important. Thus, structured prediction is required.

The basic formula of structured prediction is as follows:

$$\hat{y} = \underset{y \in Y(x)}{\arg\max}\, f(x, y) \tag{2.2}$$

where $x = (x_1, x_2, ..., x_m) \in X$ is a input sequence of length m, $y = (y_1, y_2, ..., y_m) \in Y$ is an output sequence of the same length (i.e., y_i is a label for word x_i), $Y(x)$ is the set of all possible labeled sequences for a given input x, and f is the scoring function. The prediction

$\hat{y}$ indicates the possible labeled sequence in Y that maximizes the compatibility. With linear models, a score function f can be defined by weights w such as:

$$\hat{y} = \arg\max_{y \in Y(x)} w \cdot \Phi(x, y) \tag{2.3}$$

where Φ denotes a feature vector in Euclidean space.

In our research, we adopt a structured prediction to extract both +/-effect events and their affected entity since inputs and outputs of our task are inter-related labels.

3.0 GENERAL INFORMATION ABOUT SENTIMENT ANALYSIS

With a growing interest in sentiment analysis, many researchers put some efforts for this task. Most previous works are document-level or sentence-level sentiment analysis. That is, the task is to identify whether a document/sentence expresses opinions or not and whether the opinions are positive, negative, or neutral if a document/sentence is opinionated.

The early work by [Wiebe et al., 1999] develops the probabilistic classifier to automatically discriminate the subjective and objective category. The subjective sentence refers to aspects of language used to express opinions. They utilize the Naive Bayes classifier with several features: the presence of a pronoun, an adjective, a cardinal number, a modal other than *will*, and an adverb other than *not*, whether the sentence begins a new paragraph, and the co-occurrence of words and punctuation marks. [Hatzivassiloglou and Wiebe, 2000] study the benefit of dynamic adjectives (oriented adjectives) and gradable adjectives for the sentence-level subjectivity classification. [Yu and Hatzivassiloglou, 2003] study separating opinions from facts at the document-level and the sentence-level on TREC 8, 9, and 11 collections. They also apply the Naive Bayes and multiple Naive Bayes classifier; and the presence of semantically oriented words, the average semantic orientation score of the words, and the N-grams are used for features. [Riloff and Wiebe, 2003] suggest bootstrapping methods for the subjectivity classifier. From the labeled data, they generate patterns to represent subjective expressions, and these patterns are utilized to identify more subjective sentences. Then, based on these patterns, they classify subjective sentences. In [Wiebe and Riloff, 2005], they develop the learning method for the rule-based subjectivity classifier which looks for subjective clues. [Stepinski and Mittal, 2007] also develop the new sentence classification using a Passive-Aggressive algorithm trained on unigram, bigram, and trigram features.

Although many previous sentiment analysis works are conducted in a document-level or a sentence-level, a single sentence (or a document) may contain multiple opinions. [Wilson et al., 2004, Wilson et al., 2005] suggest phrase-level sentiment analysis. They classify clauses of each sentence by the strength of opinions being expressed in individual clauses.

Recently, researchers have become increasingly interested in social media sentiment analysis. For example, one of the earlier studies is [Go et al., 2009]. They build classifiers with unigram, bigrams, and POS information features. For training data, they consider tweets ending in good emoticons as positive examples and tweets ending in bad emoticons as negative examples. They show the unigram is the most useful feature. [Barbosa and Feng, 2010] consider not only meta-features (e.g., sentiment lexicon, and POS) but also tweet syntax features (such as retweet, hashtag, and emoticon) to detect sentiments in tweets. [Paltoglou and Thelwall, 2012] propose an unsupervised lexicon-based classifier to estimate the intensity of negative and positive emotion in informal text. Linguistic Inquiry and Word Count (LIWC)[1] is used as the emotional dictionary, and the emotional score is modified by several functions such as negation detection, capitalization detection, emoticon detection, and so on. Sentiment analysis on social media is helpful to monitor political sentiment and to predict political elections. For example, [O'Connor et al., 2010] attempt to connect measures of public opinion derived from polls with detected sentiment from Twitter. They provide evidence that social media can be substituted for traditional polling with more advanced NLP techniques.

One of important information for sentiment analysis and opinion extraction is sentiment lexicons. Especially, lexicons are important in social media settings where texts are short and informal. There are several studies to construct word-level sentiment lexicon. [Kim and Hovy, 2004] and [Peng and Park, 2011] expand manually selected seed words using WordNet's synonym and antonym relations for sentiment analysis. [Strapparava and Valitutti, 2004] also utilize WordNet relations, such as *antonymy*, *similarity*, *derived-from*, *pertains-to*, *attribute*, and *also-see*, to expand AFFECT, which is a lexical database containing terms referring to emotional states.

[1] http://www.liwc.net/

Many studies show that word-level sentiment lexicon is efficient. However, the recent work [Wiebe and Mihalcea, 2006] consider relations between word sense disambiguation and subjectivity. Thus, there is a limit with word-level sentiment lexicon. To handle sense-level subjectivity classification, [Esuli and Sebastiani, 2006] construct SentiWordNet. They first expand manually selected seed synsets in WordNet using WordNet lexical relations such as *also-see* and *direct antonymy* and train a ternary classifier. This ternary classifier is applied to all WordNet synsets to measure positive, negative, and objective score. [Gyamfi et al., 2009] label the subjectivity of word senses using the hierarchical structure and domain information in WordNet. [Akkaya et al., 2009, Akkaya et al., 2011, Akkaya et al., 2014] present the subjectivity word sense disambiguation task which is to automatically determine which word instances are being used with subjective senses and which are being used with objective senses.

Such sentiment lexicons are helpful for detecting explicitly stated opinions, but are not sufficient for recognizing implicit opinions. As we mentioned in Chapter 1, inferred opinions often have opposite polarities from the explicit sentiment expressions in the sentence; explicit sentiments must be combined with +/-effect event information to detect implicit sentiments. Thus, in this research, we focus on +/-effect event information.

4.0 OPINION INFERENCE AND +/-EFFECT EVENT

In this chapter, we explain opinion inference briefly and introduce the +/-effect corpus in Section 4.1. Then, in Section 4.2, we describe +/-effect events in detail because it is the main part in our research.

4.1 OPINION INFERENCE

As we mentioned in Chapter 1, [Deng et al., 2013, Deng and Wiebe, 2014] introduce opinion inferences. Remind the following example:

(3) *The bill would curb skyrocketing health care costs.*

With an explicit sentiment analysis system, we can recognize only one explicit sentiment expression, *skyrocketing*. Thus, we can know that the writer expresses an explicit **negative** sentiment (by *skyrocketing*) toward the entity, *health care costs* while we cannot know the writer's sentiment toward *the bill* with an explicit sentiment analysis system. However, the sentiment toward *the bill* can be inferred. The event, *curb*, has a **negative effect** (i.e., -effect) on *skyrocketing health care costs*, since they are reduced. We can reason that the writer is **positive** toward the **event** because it has a negative effect on *costs*, toward which the writer is negative. From there, we can reason that the writer is **positive** toward *the bill*, since it conducts the positive event.

For that, [Deng et al., 2013, Deng and Wiebe, 2014] have built a rule-based opinion implicature system that includes default inference rules. There are ten rule schemes implemented in the system. Among them, two opinion inference rules are utilized in the given example, which are given below. In rules, $sent(S, \alpha) = \beta$ means that S's sentiment toward α is β where α is one of a +/-effect event, an object of an event, and a agent of an event and β is either positive or negative. P $\rightarrow$ Q means to infer Q from P.

RS2: sent(S, object) $\rightarrow$ sent(S, +/-effect event)

2.1 sent(S, object) = positive $\rightarrow$ sent(S, +effect) = positive

2.2 sent(S, object) = negative $\rightarrow$ sent(S, +effect) = negative

2.3 sent(S, object) = positive $\rightarrow$ sent(S, -effect) = negative

2.4 sent(S, object) = negative $\rightarrow$ sent(S, -effect) = positive

RS3: sent(S, +/-effect event) $\rightarrow$ sent(S, agent)

3.1 sent(S, +effect) = positive $\rightarrow$ sent(S,agent) = positive

3.2 sent(S, +effect) = negative $\rightarrow$ sent(S,agent) = negative

3.3 sent(S, -effect) = positive $\rightarrow$ sent(S,agent) = positive

3.4 sent(S, -effect) = negative $\rightarrow$ sent(S,agent) = negative

In summary, we can know *sent(writer, costs) = negative* with an explicit sentiment analysis system. Then, we can know that there is -effect event, *lower*. Thus, we can infer *sent(writer, -effect) = positive* via Rule 2.4, and we can infer *sent(writer, the bill) = positive* via Rule 3.3.

However, to achieve its results, their system requires an explicit sentiment and +/-effect information. For the system to be fully automatic, it needs to be able to detect an explicit sentiment and +/-effect events automatically. For an explicit sentiment analysis system, there are several systems such as OpinionFinder [Wilson et al., 2005][1]. However, there is no resource related +/-effect events. Therefore, this research focuses on +/-effect events to support opinion inference.

[1]OpinionFinder, http://mpqa.cs.pitt.edu/opinionfinder/

4.1.1 +/-Effect Corpus

[Deng et al., 2013] introduce an annotation scheme for +/-effect events and for the sentiment of the writer toward their agents and objects. Each event is representable as a triple of text spans, ⟨ *agent, +/-effect event, object* ⟩. The agent should be a noun phrase or *implicit* when the given text doesn't have the agent information explicitly. The object also should a noun phrase.

Another component is the influencer, a word whose effect is to either retain or reverse the polarity of +/-effect event. Consider the below example:

(8) *The reform prevented companies from hurting patients.*

In this example, we know there is -effect event, *hurt*. However, *prevented* reverses the polarity. That is, in *hurting patients*, it has a negative effect on *patients*, but in *prevented companies from hurting patients*, it has positive effect on *patients*. We call such event (i.e., *prevented*) a reverser.

Now, consider:

(9) *John helped Mary to save Bill.*

In this sentence, *helped* is an influencer which retains the polarity. That is, in *save Bill*, it has a positive effect on *Bill*, and in *helped Mary to save Bill*, it also has a positive effect on *Bill*. Such event (i.e., *helped*) is a retainer.

Each influencer is also representable as a triple of text spans, ⟨ *agent, influencer (retainer or reverser), object* ⟩. The agent of an influencer should be a noun phrase or *implicit* such as the agent of +/-effect events. The object of an influencer should be another influencer or a +/-effect event.

Therefore, there are two types of annotations; triple information related +/-effect events and triple information related influencers. For instance, in the example (9), there is one triple for +/-effect and one triple for influencer such as:

John helped Mary to save Bill.
$\Rightarrow$ $\langle$ *Mary, save (+effect), Bill* $\rangle$
$\Rightarrow$ $\langle$ *John, helped (retainer),* $\langle$ *Mary, save (+effect), Bill* $\rangle$ $\rangle$

Based on this annotation scheme, +/-effect corpus[2] is created. This corpus is based on the arguing corpus [Conrad et al., 2012][3], which consists of 134 documents from blogs and editorials about a controversial topic, *the Affordable Care Act*.

To validate the reliability of the annotation scheme, Lingjia Deng, who is involved in developing this annotation scheme, and I conduct the agreement study. We firstly annotate 6 documents and discuss about disagreement parts. Then, for the agreement study, we independently annotate 15 randomly selected documents.

For the agreement of text spans, we adopt two measures. The first one is that if two spans a and b overlap, it is counted as 1, otherwise 0 such as:

$$match_1(a, b) = 1 \text{ if } |a \cap b| > 0 \tag{4.1}$$

where $|a \cap b|$ provides the number of tokens that two spans have in common.

Another measure is to measure the percentage of overlapping tokens as follows:

$$match_2(a, b) = \frac{|a \cap b|}{|b|} \tag{4.2}$$

where $|b|$ is the number of tokens in the given span b.

[2] +/-Effect corpus (also call goodFor/badFor corpus), http://mpqa.cs.pitt.edu/corpora/gfbf/
[3] Arguing Corpus, http://mpqa.cs.pitt.edu/corpora/arguing/

	+/-Effect & Influencer	Agent	Object
$match_1$	0.70	0.92	1.00
$match_2$	0.69	0.87	0.97

Table 1: The agreement score about a span of +effect events & influencers, agents, and objects.

Table 1 shows the agreement score about a span of +effect events and influencers, agents, and objects. It shows high agreement scores with two measures.

To measure agreement for polarities (i.e., +effect vs. -effect, and retainer vs. reverser), we use κ [Artstein and Poesio, 2008]. κ is a statistic to measure inter-rater agreement for qualitative labels. The equation for κ is:

$$\kappa = \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e} \tag{4.3}$$

where p_0 is the relative observed agreement among annotators and p_e is the hypothetical probability of chance agreement. The change agreement p_e can be calculated with the observed data by calculating the probabilities of each annotator randomly saying each label.

If annotators are in complete agreement, κ score is 1; if there is no agreement between annotators, it is equal or less than 0. We get 0.97 κ agreement score about polarities of +/-effect events and influencers.

4.2 +/-EFFECT EVENT

+Effect events mean events that have positive or negative effect on entities. There are many varieties of +effect events (e.g., *save* and *create*) and -effect events (e.g., *lower* and *hurt*).

[Anand and Reschke, 2010] present six verb classes as evaluability functor classes: creation, destruction, gain, loss, benefit, and injury. Creation/destruction events result in states involving existence that means a participant has/lacks existence. Gain/loss events result in states involving possession that means one participant has/lacks possession of another. Benefit/Injury events result in states involving affectedness that means a participant has a positive/negative property.

Among six verb classes, the creation, gain, and benefit classes are +effect events based on the definition. As we said, in creation events, a participant has existence. It indicates these events have a positive effect on the participant. For example, in the sentence *baking a cake*, *baking* has a positive effect on *the cake* because it is created. The gain and benefit classes are also +effect events. In the sentence *increasing the tax rate*, *increasing* has a positive effect on the *the tax rate*; and in the sentence *comforting the child*, *comforting* has a positive effect on *the child*.

The antonymous classes of each (i.e., destruction, loss, and injury) are -effect events. In the sentence *destroying the building*, *destroying* has a negative effect on *the building* since it is disappeared. In the sentence *demand decreasing*, *decreasing* has a negative effect on *demand*; and in the sentence *killing Bill*, *killing* has a negative effect on *Bill*.

4.2.1 +/-Effect Events, Sentiment Terms, vs. Connotation Terms

There are several lexicons related as lexicons of +/effect events. The first one is sentiment lexicons [Wilson et al., 2005, Esuli and Sebastiani, 2006, Su and Markert, 2009]. As we mentioned in Chapter 1, the sentiment lexicon consists of words or expressions which are used to express subjective feelings and sentiments/opinions such as *wonderful*, *elegant*, *horrible*, and *bad*.

Another one is lexicons of connotation terms [Feng et al., 2011, Kang et al., 2014]. Connotation lexicon is a new type of lexicon that list words with connotative polarity. For examples, *award* and *promotion* are positive connotation; *cancer*, *war* are negative connotation. Connotation lexicons differ from sentiment lexicons. Sentiment lexicons express sentiments while connotation lexicons concern words that evoke or even simply associate with a specific polarity of sentiment.

Even though these lexicons seem similar with +/-effect events, they are different. Consider the following example:

perpetrate:

S: (v) perpetrate, commit, pull (perform an act, usually with a negative connotation) "perpetrate a crime"; "pull a bank robbery"

In this example, *perpetuate* is an **objective** term according to SentiWordNet [Esuli and Sebastiani, 2006, Baccianella et al., 2010] [4], that is, it is neutral. Then, as the definition already mentioned, it has a **negative** connotation by [Kang et al., 2014]. However, it has a **positive effect** on *a crime* since performing a crime brings it into existence. Like this, a single event may have different polarities of sentiment, connotation, and +/-effect. Therefore, we need to acquire a new type of lexicon of +/-effect events to make opinion inference.

4.2.2 Sense-level +/-Effect Ambiguity

As we mentioned, a word may have one or more meanings. To handle these, we utilize WordNet explained in Section 2.1. We assume that a synset is exactly one of +effect, -effect, or Null. Since words often have more than one sense, the polarity of a **word** may or may not be consistent, as the following WordNet examples show.

[4]SentiWordNet, http://sentiwordnet.isti.cnr.it/

First consider the words *encourage* and *assault*. Each of them has 3 senses. All senses of *encourage* have positive effects on the entity, and all senses of *assault* have negative effects on the entity. The polarity is always same regardless of sense. In such cases, for our purposes, which particular sense is being used does not need to be determined because any instance of the word will be +effect or -effect; that is, word-level approaches can work well.

- A word with only +effect senses: **encourage**

  S: (v) promote, advance, boost, further, encourage (contribute to the progress or growth of) "I am promoting the use of computers in the classroom"

  S: (v) encourage (inspire with confidence; give hope or courage to)

  S: (v) encourage (spur on) "His financial success encouraged him to look for a wife"

- A word with only -effect senses: **assault**

  S: (v) assail, assault, set on, attack (attack someone physically or emotionally) "The mugger assaulted the woman"; "Nightmares assailed him regularly"

  S: (v) rape, ravish, violate, assault, dishonor, dishonor, outrage (force (someone) to have sex against their will) "The woman was raped on her way home at night"

  S: (v) attack, round, assail, lash out, snipe, assault (attack in speech or writing) "The editors of the left-leaning paper attacked the new House Speaker"

However, word-level approaches are not applicable for all the words. Consider the words *inspire* and *neutralize*. They have 6 senses respectively. For *inspire*, while the third sense and the fourth sense have positive effects on the entity, the sixth sense doesn't have any polarity, i.e., it is a Null (we don't think of inhaling air as positive effects on the air). Also, while the second sense of *neutralize* has negative effects on the entity, the sixth sense is Null (neutralizing a solution just changes its pH). Therefore, if word-level approaches are applied using these words, some Null instances may be incorrectly classified as +effect or -effect events.

- A word with +effect and Null senses: **inspire**

  S: (v) inspire, animate, invigorate, enliven, exalt (heighten or intensify) “These paintings exalt the imagination”

  S: (v) inspire (supply the inspiration for) “The article about the artist inspired the exhibition of his recent work”

  S: (v) prompt, inspire, instigate (serve as the inciting cause of) “She prompted me to call my relatives”

  S: (v) cheer, root on, inspire, urge, barrack, urge on, exhort, pep up (spur on or encourage especially by cheers and shouts) “The crowd cheered the demonstrating strikers”

  S: (v) revolutionize, revolutionise, inspire (fill with revolutionary ideas)

  S: (v) inhale, inspire, breathe in (draw in (air)) “Inhale deeply”; “inhale the fresh mountain air”; ”The patient has trouble inspiring”; “The lung cancer patient cannot inspire air very well”

- A word with -effect and Null senses: **neutralize**

  S: (v) neutralize (make politically neutral and thus inoffensive) “The treaty neutralized the small republic”

  S: (v) neutralize, neutralise, nullify, negate (make ineffective by counterbalancing the effect of) “Her optimism neutralizes his gloom”; “This action will negate the effect of my efforts”

  S: (v) counteract, countervail, neutralize, counterbalance (oppose and mitigate the effects of by contrary actions) “This will counteract the foolish actions of my colleagues”

  S: (v) neutralize, neutralise, liquidate, waste, knock off, do in (get rid of (someone who may be a threat) by killing) “The mafia liquidated the informer”; “the double agent was neutralized”

  S: (v) neutralize, neutralise (make incapable of military action) S: (v) neutralize, neutralise (make chemically neutral) “She neutralized the solution”

The following is another example of a word with senses of different classes:

- A word with +effect and -effect senses: **purge**

  S: (v) purge (oust politically) "Deng Xiao Ping was purged several times throughout his lifetime"

  S: (v) purge (clear of a charge)

  S: (v) purify, purge, sanctify (make pure or free from sin or guilt) "he left the monastery purified"

  S: (v) purge (rid of impurities) "purge the water"; "purge your mind"

The word *purge* has 4 senses. In the first sense, the polarity is -effect since it has a negative effect on *Deng Xizo Ping*. However, the other cases have a positive effect on the entity. A purely word-based approach is blind to these cases.

In fact, words often have mixtures of +effect, -effect, and Null (i.e., neither) senses. We find that 45.6% verbs in WordNet contain two or more senses (i.e., homonymy). Among them, 63.8% words have some kind of +/-effect ambiguity. 11.3% words have mixtures of +effect, -effect, and Null senses; 3.9% words have mixtures of +effect and -effect; 25.9% and 22.7% words have +effect & Null or -effect & Null.

In the +/-effect corpus mentioned in Section 4.1.1, 1,411 +/-effect instances are annotated; 196 different +effect words and 286 different -effect words appear in these instances. Among them, 10 words appear in both +effect and -effect instances, accounting for 9.07% of all annotated instances. Since only words (not senses) are annotated in this corpus, such conflicts arise. One example is *fight*. In the corpus instance *fight for a piece of legislation*, *fight* has a positive effect on *a piece of legislation*. This is the fourth sense of *fight*. However, in the corpus instance *we need to fight this repeal*, the meaning of *fight* here is the second sense, so *fight* has a negative effect on *this repeal*.

- **fight**

  S: (v) contend, fight, struggle (be engaged in a fight; carry on a fight) "the tribesmen fought each other"; "Siblings are always fighting"; "Militant groups are contending for control of the country"

  S: (v) fight, oppose, fight back, fight down, defend (fight against or resist strongly) "The senator said he would oppose the bill"; "Don't fight it!"

  S: (v) fight, struggle (make a strenuous or labored effort) "She struggled for years to survive without welfare"; "He fought for breath"

  S: (v) crusade, fight, press, campaign, push, agitate (exert oneself continuously, vigorously, or obtrusively to gain an end or engage in a crusade for a certain cause or person; be an advocate for) "The liberal party pushed for reforms"; "She is crusading for women's rights"; "The Dean is pushing for his favorite candidate"

Therefore, approaches for determining the +/-effect event of an instance that are sense-level instead of word-level promise to have higher precision. In this research, we consider sense-level +/-effect events.

4.2.3 Lexical Category of +/-Effect Events

In examples (3) and (4), +/-effect events are verbs such as *curb* and *passed*.

(3) *The bill would curb skyrocketing health care costs.*

(4) *Oh no! The voters passed the bill.*

In most case, +/-effect events are verbs. However, sometimes we have to consider phrasal verb, not only verb word. Consider following two examples:

(10) *He sides with U.S. President Barack Obama.*

(11) *I'm siding against the current candidate.*

In both sentences (10) and (11), a verb is *side*. However, the polarity of +/-effect of *side* is different according to a preposition. In the sentence (10), because *side* is written with *with*, it has a positive effect on the entity, *U.S. President Barack Obama*. On the other hand, in the sentence (11), *side* has a negative effect on *the current candidate* since it is written with *against*. The below show the WordNet information of *side* as a verb:

- **side**

  S: (v) side (take sides for or against) "Who are you siding with?"; "I'm siding against the current candidate"

As you can see, *side* has only one sense as a verb. From the short definition, we can know that the polarity of +/-effect of the given sense is different depending on prepositions. This case is a conflict with our assumption that a sense is exactly one of +effect, -effect, or Null, mentioned in Section 4.2.2. In this research, we ignore these cases because the number of these cases is a little. Moreover, WordNet can cover some phrasal verbs such as *fight down* and *root for*. We only consider verbs and phrasal verbs in WordNet.

As [Deng et al., 2013] mentioned, +/-effect events need not be verbs and phrasal verbs. Consider the following examples:

(12) *Italys support for the Iraqi government will never waver.*

(13) *President Obama's reelection has had a devastating impact on Fox News.*

In the sentence (12), *support* is +effect on *the Iraqi government*; and in the sentence (13), *reelection* is +effect on *President Obama*. In these examples, +/effect events are nouns, not verbs. However, these cases account for a small portion. Therefore, in this research, we only focus verbs, not nouns.

4.2.4 +/-Effect Event and Affected Entity

In most case, affected entities of +/-effect events are themes. In the previous example (3), *curb* is a -effect event and its affected entity is a theme of *curb* (i.e., *skyrocketing health care costs*). In the example (4), an affected entity of a +effect event, *passed*, is a theme of *passed* (i.e., *the bill*).

However, sometimes an agent of +/-effect events can be an affected entity. Remind the following example:

(5) *Yay! Johns team lost the first game.*

In this case, the event, *lost*, has a negative effect on the agent of *lost* (i.e., *John's team*), not the theme of *lost* (i.e., *the first game*). There is another example:

(14) *The senator carried his home state.*

In this example, the meaning of *carry* is *winning in an election*. Therefore, *carried* has a positive effect on the agent of *carried* (i.e., *the senator*).

Moreover, in some cases, both the agent and the theme can be affected entities with the same or different +/-effect polarity. Consider following examples:

(15) *This car outperforms all others in its class.*

(16) *The army took the fort on the hill.*

In the sentence (15), *outperforms* has a positive effect on the agent of *outperforms* (i.e., *this car*) while it has a negative effect on the theme of *outperforms* (i.e., *all others in its class*). The event in the sentence (16), *took*, is used in the meaning *take by force*, so it also has a different +/-effect polarity on the agent and the theme. That is, *took* has +effect on *the army* since it possess *the fort on the hill*; but it has -effect on *the fort on the hill* because it is lost.

In addition, affected entities may not be both the agent and the theme of +/-effect events. In the below sentence, *imparts* has a positive effect on *the students* which is neither the agent nor the theme of *imparts*.

(17) *The teacher imparts a new skill to the students.*

On rare occasion, the polarity of +/-effect events of the given synset can be different depending on the type of affected entities. Consider the following synset:

S: (v) tie down, tie up, bind, truss (secure with or as if with ropes) "tie down the prisoners"; "tie up the old newspapers and bring them to the recycling shed"

In the first example, since the affected entity *the prisoners* is a person, *tie down* has a negative effect on the affected entity. However, in the second example, the affected entity *the old newspapers* is an object, so this synset should be Null since the given event doesn't have neither positive nor negative effect on the affected entity. This case is a conflict with our assumption that a sense is exactly one of +effect, -effect, or Null, mentioned in Section 4.2.2. Therefore, in this research, we ignore these cases.

5.0 +/-EFFECT EVENTS AND WORDNET

In this chapter, we present the feasibility of using WordNet for +/-effect lexicon acquisition with the simple method.

As we mentioned in Section 4.2.2, we need a sense-level approach to acquire +/-effect lexicon knowledge, leading us to employ lexical resources with fine-grained sense rather than word representations. There are several resources with sense information such as WordNet (described in Section 2.1) and FrameNet (described in Section 2.2). As we mentioned in Section 2.1, WordNet can cover more senses. The FrameNet database contains about 1,200 semantic frames and about 13,000 lexical units; however, WordNet contains more than 150,000 words organized in more than 100,000 synsets. Also, while FrameNet cannot cover all possible senses of given words since it considers only lexical units corresponding to the given semantic frames, WordNet contains all possible senses of given words. (That is, while FrameNet cannot cover all meanings of words, WordNet can provide all meanings of given words - 150,000 words.) Moreover, WordNet provides a synonym set, called synsets, that are interchangeable in some context. The synset information is helpful because we can reduce the redundancy. In other words, since they are interchangeable in some context, they should have the same polarity of +/-effect event; we can avoid duplication. In addition, synsets in WordNet are interlinked by semantic relations which may be useful information to acquire +/-effect events. Thus, we adopt WordNet which is a widely-used lexical resource for +/-effect lexicon acquisition.

Our goal in this chapter is that starting from the seed set we explore how +/-effect events are organized in WordNet via semantic relations and expand the seed set based on those semantic relations. For that, we adopt an automatic bootstrapping method which disambiguates +/-effect polarity at the sense-level utilizing WordNet.

For the bootstrapping method, we first need seed data. To get the seed lexicon, we utilize FrameNet because we believe that using FrameNet to find +/-effect words is easier than finding +/-effect words without any information since words may be filtered by semantic frames. First, an annotator who didn't have access to our +/-effect corpus selects promising semantic frames as +/-effect in FrameNet, and we pick out all lexical units from selected semantic frames. From them, we extract +effect verb words and -effect verb words. For the pure seed set, we ignore conflicting words between the +effect verb set and the -effect verb set. Since we need a sense-level lexicon as a seed lexicon, not a word-level lexicon, we finally extract all senses of these +/-effect words and -effect words from WordNet and randomly select 200 +effect synsets and 200 -effect synsets as the seed lexicon. Section 5.1 explains the seed lexicon in detail. Then, we describe our evaluation metrics in Section 5.2.

As we mentioned, to expand the given seed set based on WordNet semantic relations, we adopt the bootstrapping method. Our detail method is explained in Section 5.3.

The expanded lexicon is evaluated in two ways. First, the lexicon is evaluated against a corpus that has been annotated with +/-effect information at the word level. Section 5.4 presents this corpus evaluation. Second, samples from the expanded lexicon are manually annotated at the sense level, which gives some idea of the prevalence of +/-effect lexical ambiguity and provides a basis for sense-level evaluation. Section 5.5 presents the evaluation based on sense annotation. Also, we conduct the agreement study in this section.

Finally, related work is described in Section 5.6 and summary is given in Section 5.7.

This work is presented in 5th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis (WASSA) which is ACL workshop [Choi et al., 2014].

5.1 SEED LEXICON

To preserve the +/-effect corpus (described in Section 4.1.1) for evaluation, we create a seed set that is independent from the corpus. An annotator who didn't have access to the +/-effect corpus manually selects +/-effect events from FrameNet.

As we mentioned in Section 2.2, FrameNet is based on a theory of meaning called Frame Semantics. In FrameNet, a lexical unit is a pairing of a word with a meaning, that is, it corresponds to a sense in WordNet. Each lexical unit of a polysemous/homonymous word belongs to a different semantic frame, which is a description of a type of event, relation, or entity and, where appropriate, its participants. For instance, in the **Creating** frame, the definition is that a **Cause** leads to the formation of a **Created_entity**. It has a positive effect on the theme, **Created_entity**. This frame contains about 10 lexical units such as *assemble*, *create*, *yield*, and so on. FrameNet consists of about 1,000 semantic frames and about 10,000 lexical units.

FrameNet is a useful resource to select +/-effect verb words since each semantic frame covers multiple lexical units. We believe that using FrameNet to find +/-effect words is easier than finding +/-effect words without any information since words may be filtered by semantic frames. To select +/-effect words, an annotator first identifies promising semantic frames as +/-effect events and extracts all lexical units from them. Then, the annotator goes through them and picks out the lexical units which s/he judges to be +effect or -effect. In total, 736 +effect lexical units and 601 -effect lexical units are selected from 463 semantic frames.

As we mentioned in Section 4.2.4, events may have positive or negative effects on themes of a given event, agents of a given event, or other entities. Thus, we consider a sense to be +effect (-effect) if it has +effect (-effect) on an entity, which may be the agent, the theme, or some other entity. In this work, we ignore the case that both the agent and the theme are affected entities with the same or different +/-effect polarity.

For a seed set and an evaluation set in this work, we need annotated sense-level +/-effect data. If we can convert selected lexical units from FrameNet into WordNet automatically, it will be easy to create sense-level +/-effect data. However, mappings between FrameNet and WordNet are not perfect. Thus, we opt to manually annotate the senses of the words in the word-level lexicon. We first extract all words from 736 +effect lexical units and 601 -effect lexical units; this extracts 606 +effect words and 537 -effect words (the number of words is smaller than the number of lexical units because one word can have more than one lexical unit). Among them, 14 words (e.g., *crush, order,* etc.) are in both the +effect word

set and the -effect word set. That is, these words have both +effect and -effect meanings. Recall that this annotator is focusing on semantic frames, not on words - s/he does not look at all the senses of all the words. For the pure seed set, we ignore these 14 words; thus, we consider only 592 +effect words and 523 -effect words.

Decomposing each word into its senses in WordNet, there are 1,525 +effect senses and 1,154 -effect senses. 83 words extracted from FrameNet overlap with +/-effect instances in the +/-effect corpus. For independence, those words were discarded. Among the senses of the remaining words, we randomly choose 200 +effect senses and 200 -effect senses as the seed lexicon.

5.2 EVALUATION METRICS

As we mentioned, we evaluate our expanded lexicon in two ways; the evaluation based on corpus and the evaluation based on sense annotation.

In corpus evaluation, we use the +/-effect annotations in the +/-effect corpus as a gold standard. The annotations in the corpus are at the word level. To use the annotations as a sense-level gold standard, all the senses of a word marked +effect (or -effect) in the corpus are considered to be +effect (or -effect). While this is not ideal, this allows us to evaluate the lexicon against the only corpus evidence available.

To evaluate our system with this data, we calculate the accuracy that is how many +effect (or -effect) synsets (i.e., senses) are correctly detected by our system. The accuracy is calculated as follows:

$$Accuracy = \frac{\text{Number of correctly detected synsets based on the gold standard}}{\text{Number of all synsets which are in the gold standard and are detected by the system}} \quad (5.1)$$

For that, we first define *+effectOverlap* and *-effectOverlap* because we can only consider synsets which are in the gold standard. While *+effectOverlap* means the overlap between the synsets in the expanded +effect (or -effect) lexicon and the gold-standard **+effect** set, *-effectOverlap* is the overlap between the synsets in the expanded +effect (or -effect) lexicon and the gold-standard **-effect** set.

That is, the accuracy for +effect is calculated based on *+effectOverlap* and *-effectOverlap* within the expanded **+effect** lexicon such as:

$$Accuracy_{+effect} = \frac{\text{The number of +effectOverlap}}{\text{The number of +effectOverlap + the number of -effectOverlap}} \quad (5.2)$$

In this equation, *+effectOverlap* indicates the overlap between the synsets in the expanded **+effect** lexicon and the gold-standard **+effect** set, and *-effectOverlap* is the overlap between the synsets in the expanded **+effect** lexicon and the gold-standard **-effect** set.

Similarly, the accuracy for -effect is calculated based on *+effectOverlap* and *-effectOverlap* within the expanded **-effect** lexicon such as:

$$Accuracy_{-effect} = \frac{\text{The number of -effectOverlap}}{\text{The number of +effectOverlap + the number of -effectOverlap}} \quad (5.3)$$

In this case, *+effectOverlap* means the overlap between the synsets in the expanded **-effect** lexicon and the gold-standard **+effect** set, and *-effectOverlap* is the overlap between the synsets in the expanded **-effect** lexicon and the gold-standard **-effect** set.

For sense annotation evaluation, we first select 60 words and two annotators annotate +/-effect polarity of all synsets of these words. We consider this data as the gold standard. Based on this annotation, we also calculate the accuracy such as:

$$Accuracy = \frac{\text{Number of correctly detected synsets}}{\text{Number of all synsets which are in the gold standard and are detected by the system}} \tag{5.4}$$

Moreover, since we conduct the annotation study, we need to evaluate the annotation work. To measure agreement between the annotators, we calculate two measures: percent agreement and κ. Percent agreement is calculate such as:

$$PercentAgreement = \frac{\text{Number of synsets annotated as the same polarity by annotators}}{\text{Number of all synsets annotated by annotators}} \tag{5.5}$$

As we mentioned in Section 4.1.1, κ is a statistic to measure inter-rater agreement for qualitative labels. The equation for κ is:

$$\kappa = \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e} \tag{5.6}$$

where p_0 is the relative observed agreement among annotators and p_e is the hypothetical probability of chance agreement. The change agreement p_e can be calculated with the observed data by calculating the probabilities of each annotator randomly saying each label. If annotators are in complete agreement, κ score is 1; if there is no agreement between annotators, it is equal or less than 0.

5.3 BOOTSTRAPPING METHOD

In WordNet, verb synsets are arranged into hierarchies, that is, verb synsets towards the bottom of the trees express increasingly specific manners. Thus, we can follow *hypernym* relations to more general synsets and *troponym* relations to more specific verb synsets. Since the troponym relation refers to a specific elaboration of a verb synsets, we hypothesize that troponyms of a synset tends to have its same polarity (i.e., +effect, -effect, or Null). We only consider the direct troponyms in a single iteration. Although the hypernym is a more general term, we hypothesize that direct hypernyms tend to have the the same or neutral polarity, but not the opposite polarity. Also, the *verb groups* are promising; even though the coverage is incomplete, we expect the verb groups to be the most helpful.

WordNet Similarity[1], is a facility that provides a variety of semantic similarity and relatedness measures based on information found in the WordNet lexical database. We choose Jiang&Conrath [Jiang and Conrath, 1997] (*jcn*) method which has been found to be effective for such tasks by NLP researchers. When two concepts aren't related at all, it returns 0. The more they are related, the higher the value is retuned. We regard synsets with similarity values greater than 1.0 to be similar synsets. That is, we consider there is a relation between synsets which have a higher similarity value.

Beginning with its seed set, each lexicon (+effect and -effect) is expanded iteratively. On each iteration, for each synset in the current lexicon, all of its direct troponyms, direct hypernyms, and members of the same verb group are extracted and added to the lexicon for the next iteration. Similarity, for each synset, all words with above-threshold *jcn* values are added. For new senses that are extracted for both the +effect and -effect lexicons, we ignore such senses, since there is conflicting evidence (recall that we assume a synset has only one polarity, even if a word may have synsets of different polarities).

[1]WN Similarity, http://wn-similarity.sourceforge.net/

5.4 CORPUS EVALUATION

In this section, we use the +/-effect annotations in the +/-effect corpus as a gold standard. The annotations in the corpus are at the word level. To use the annotations as a sense-level gold standard, all the senses of a word marked +effect (-effect) in the corpus are considered to be +effect (-effect). While this is not ideal, this allows us to evaluate the lexicon against the only corpus evidence available.

The 196 words that appear in +effect instances in the corpus have a total of 897 synsets, and the 286 words that appear in -effect instances have a total of 1,154 synsets. Among them, 125 synsets are conflicted: a sense of a word marked +effect in the corpus could be a member of the same synset as a sense of a word marked -effect in the corpus. For a more reliable gold-standard set, we ignore these conflicted synsets. Thus, the gold-standard set contains 772 +effect synsets and 1,029 -effect synsets.

Table 2 shows the results after five iterations of lexicon expansion. In total, the +effect lexicon contains 4,157 synsets and the -effect lexicon contains 5,071 synsets. The top half gives the results for the +effect lexicon and the bottom half gives the results for the -effect lexicon. As we mentioned in Section 5.2, *+effectOverlap* means the overlap between the senses in the lexicon in that row and the gold-standard **+effect** set, while *-effectOverlap* is the overlap between the senses in the lexicon in that row and the gold-standard **-effect** set. That is, of the 772 synsets in the +effect gold standard, 449 (58%) are in the +effect expanded lexicon while 105 (14%) are in the -effect expanded lexicon. With this information, we calculate the accuracy described in Section 5.2.

Overall, accuracy is higher for the -effect than the +effect lexicon. The results in the table are broken down by semantic relations. Note that the individual counts do not sum to the totals because senses of different words may actually be the same synset in WordNet. The results for the -effect lexicon are consistently high over all semantic relations. The results for the +effect lexicon are more mixed, but all relations are valuable. This evaluation shows that WordNet is promising for expanding such sense-level lexicons. Even though the seed set is completely independent from the corpus, the expanded lexicons coverage of the corpus is not small.

+Effect				
	#senses	*#+effecOverlap*	*#-effecOverlap*	Accuracy
Total	4,157	449	176	0.72
WordNet Similarity	1,073	134	75	0.64
Verb Groups	242	69	24	0.74
Troponym	4,084	226	184	0.55
Hypernym	223	75	33	0.69
-Effect				
	#senses	*#+effecOverlap*	*#-effecOverlap*	Accuracy
Total	5,071	105	562	0.84
WordNet Similarity	1,008	34	190	0.85
Verb Groups	255	11	86	0.89
Troponym	4,258	66	375	0.85
Hypernym	286	16	77	0.83

Table 2: Results after the simple lexicon expansion

Overall, the verb group is the most informative relation, as we suspected. It shows the highest accuracy in both +/-effect.

WordNet Similarity is advantageous because WordNet Similarity detects similar synsets automatically and provides coverage beyond the semantic relations coded in WordNet.

Although the +effect lexicon accuracy for the troponym relation is not high, it has the advantage is that it yields the most number of synsets. Its lower accuracy doesn't support our original hypothesis. We first hypothesized that verbs lower down in the hierarchy would tend to have the same polarity since they express specific manners characterizing an event. However, this hypothesis is wrong sometimes. Even though most troponyms have the same polarity, there are many exceptions. For example, *protect#v#1*, which means the first sense of the verb *protect*, has 18 direct troponyms such as *cover for#v#1*, *overprotect#v#2*, and so on. *protect#v#1* is a +effect event because the meaning is *"shielding from danger"* and most troponyms are also +effect events. However, *overprotect#v#2*, which is one of troponyms of *protect#v#1*, is a -effect event, not a +effect event.

For the hypernym relation, the number of detected synsets is not large because many were already detected in previous iterations (in general, there are fewer nodes on each level as hypernym links are traversed).

5.5 SENSE ANNOTATION EVALUATION

For a more direct evaluation, two annotators (one is Lingjia Deng who created the annotation scheme for +effect corpus and another is me) independently annotate a sample of synsets. We randomly select 60 words among the following classes: 10 pure +effect words (i.e., all senses of the words are classified by the expansion method, and all senses are put into the +effect lexicon), 10 pure -effect words, 20 mixed words (i.e., all senses of the words are classified by the expansion method, and some senses are put into the +effect lexicon while others are put into the -effect lexicon), and 20 incomplete words (i.e., some senses of the words are not classified by the expansion method).

The total number of synsets is 151; 64 synsets are classified as +effect, 56 synsets are classified as -effect, and 31 synsets are not classified. We include more mixed than pure words to make the results of the study more informative. Further, we want to include non-classified synsets as decoys for the annotators. The annotators only see the synset entries from WordNet. They doesn't know whether the system classifies a synset as +effect or -effect or whether it doesn't classify it at all.

Table 3 evaluates the lexicons against the manual annotations, and in comparison to the majority class baseline. The top half of the table shows results when treating the first annotator's annotations as the gold standard, and the bottom half shows the results when treating the second annotator's as the gold standard. Among 151 synsets, the first annotator (Annotator1) annotated 56 synsets (37%) as +effect, 51 synsets (34%) as -effect, and 44 synsets (29%) as Null. The second annotator (Annotator2) annotated 66 synsets (44%) as +effect, 55 synsets (36%) as -effect, and 30 (20%) synsets as Null. The incorrect cases are divided into two sets: *incorrect opposite* consists of synsets that are classified as the opposite polarity by the expansion method (e.g., the sense is classified into +effect, but annotator annotates it as -effect), and *incorrect Null* consists of synsets that the expansion method classifies as +effect or -effect, but the annotator marked it as Null. We report the accuracy described in Section 5.2 and the percentage of cases for each incorrect case. The accuracies substantially improve over baseline for both annotators and for both classes.

	accuracy	% incorrect opposite	% incorrect Null	baseline
Annotator1	0.53	0.16	0.32	0.37
Annotator2	0.57	0.24	0.19	0.44

Table 3: Results against sense-annotated data

	+effect accuracy	-effect accuracy	baseline
Annotator1	0.74	0.83	0.37
Annotator2	0.68	0.74	0.44

Table 4: Accuracy broken down for +/-effect

In Table 4, we break down the results into +/-effect classes. The *+effect accuracy* measures the percentage of correct +effect senses out of all senses annotated as +effect according to the annotations (same as *-effect accuracy*). As we can see, the accuracy is higher for the -effect than the +effect. The conclusion is consistent with what we have discovered in Section 5.4.

By the first annotator, 8 words are detected as mixed words, that is, they contain both +effect and -effect senses. By the second annotator, 9 words are mixed words (this set includes the 8 mixed words of the first annotator). Among the randomly selected 60 words, the proportion of mixed words range from 13.3% to 15%, according to the two annotators. This shows that +/-effect lexical ambiguity does exist.

To measure agreement between the annotators, we calculate two measures: percent agreement and κ, as we described in Section 5.2. κ measures the amount of agreement over what is expected by chance, so it is a stricter measure. Percent agreement is 0.84 and κ is 0.75. It is positive, providing evidence that the annotation task is feasible and that the concept of +/-effect gives us a natural coarse-grained grouping of senses.

5.6 RELATED WORK

As we mentioned in Section 2.1, WordNet is one sense inventory which is widely used in NLP. There are several works to successfully adopt WordNet to construct subjectivity, sentiment, and connotation lexicons which are similar (but different) lexicons with +/-effect lexicon.

[Esuli and Sebastiani, 2006] construct SentiWordNet for sentiment lexicons. They assume that terms with the same polarity tend to have similar glosses. So, they first expand a manually selected seed set of senses using WordNet lexical relations such as *also-see* and *direct antonymy* and train two classifiers, one for positive and another for negative. As features, a vector representation of glosses is adopted. These classifiers are applied to all WordNet senses to measure positive, negative, and objective scores. In extending their work [Esuli and Sebastiani, 2007], the PageRank algorithm is applied to rank senses in terms of how strongly they are positive or negative. In the graph, each sense is one node, and two nodes are connected when they contain the same words in their WordNet glosses. Moreover, a random-walk step is adopted to refine the scores in their recent work [Baccianella et al., 2010].

For subjectivity lexicons, [Gyamfi et al., 2009] construct a classifier to label the subjectivity of word senses. The hierarchical structure and domain information in WordNet are exploited to define features in terms of similarity (using the LCS metric in [Resnik, 1995]) of target senses and a seed set of senses. Also, the similarity of glosses in WordNet is considered. Moreover, [Su and Markert, 2009] adopt a semi-supervised mincut method to recognize the subjectivity of word senses. To construct a graph, each node corresponds to one WordNet sense and is connected to two classification nodes (one for subjectivity and another for objectivity) via a weighted edge that is assigned by a classifier. For this classifier, WordNet glosses, relations, and monosemous features are considered. Also, several WordNet relations (e.g., *antonymy*, *similar-to*, *direct hypernym*, etc.) are used to connect two nodes.

[Kang et al., 2014] present a unified model that assigns connotation polarities to both words and senses encoded in WordNet. They formulate the induction process as collective inference over pairwise-Markov Random Fields and apply loopy belief propagation for inference. Their approach relies on selectional preferences of *connotative predicates*; the polarity of a connotation predicate suggests the polarity of its arguments. We have not discovered an analogous type of predicate for the problem we address.

As we mentioned in Section 4.2, +/-effect events are different as sentiments and connotations. Our work is the first NLP work for the +effect lexicon.

5.7 SUMMARY

In this chapter, we present the feasibility of using WordNet for sense-level +/-effect lexicon acquisition with the bootstrapping method.

As we mentioned in Section 4.2.2, we need a sense-level approach to acquire +/-effect lexicon knowledge, leading us to employ lexical resources with fine-grained sense rather than word representations. In our work, we adopt WordNet which is widely-used lexical resource since WordNet can cover more words and senses than other resources and it also contains all possible senses of given words. Moreover, WordNet provides a synonym set, called synsets, and synsets are interlinked by semantic relations which are useful information to acquire +/-effect events.

Our goal in this chapter is that starting from the seed set we explore how +/-effect events are organized in WordNet via semantic relations and expand the seed set based on those semantic relations.

For our goal, we first need seed data. As we mentioned in Section 5.1, to get the seed lexicon, we utilize FrameNet because we believe that using FrameNet to find +/-effect words is easier than finding +/-effect words without any information since words may be filtered by semantic frames. As the seed lexicon, we select 200 +effect synsets and 200 -effect synsets.

With this seed data, to explore how +/-effect events are organized in WordNet via semantic relations, we adopt an automatic bootstrapping method which disambiguates +/-effect polarity at the sense-level utilizing WordNet as described in Section 5.3. That is, we expand the seed set based on WordNet semantic relations. In this chapter, we consider hierarchical relations (i.e., hypernym and troponym) and verb groups. Moreover, we utilize WordNet similarity to get more relations between synsets.

The expanded lexicon is evaluated in two ways. In Section 5.4, we first present the corpus evaluation. That is, the lexicon is evaluated against the +/-effect corpus that has been annotated with +/-effect information at the word level. Since we need a sense-level gold standard, all the synsets of +/-effect words in the corpus are considered to be +/-effect synsets. While this is not ideal, this allows us to evaluate the lexicon against the only corpus evidence available.

For a more direct evaluation, we also conduct the evaluation based on sense annotation in Section 5.5. Samples from the expanded lexicon are manually annotated at the sense level, which gives some idea of the prevalence of +/-effect lexical ambiguity and provides a basis for sense-level evaluation.

Our evaluations show that WordNet is promising for expanding sense-level +/-effect lexicons. Even though the seed set is completely independent from the corpus, the expanded lexicon's coverage of the corpus is not small. The accuracy of the expanded lexicon is substantially higher. Also, the results of the agreement study are positive, providing evidence that the annotation task is feasible and that the concept of +/-effect gives us a natural coarse-grained grouping of senses.

6.0 EFFECTWORDNET: SENSE-LEVEL +/-EFFECT LEXICON

In this chapter, we address methods for creating a lexicon of +/-effect events, to support opinion inference rules. Due to significant sense ambiguity as we mentioned in Section 4.2.2, we develop a sense-level lexicon rather than a word-level lexicon. As we mentioned in Section 4.2.3, we focus only verbs as +/-effect events in this work. We call such sense-level +/-effect lexicon **EffectWordNet**.

Our assumption in this chapter is that each sense (or synset in WordNet) has only one +/-effect polarity. Moreover, we hypothesize that +/-effect polarity tends to propagate by semantic-related relations such as hierarchical information.

One of our goals is to develop the method that applied to many verb senses, not just to senses of given words such as [Akkaya et al., 2009, Akkaya et al., 2011] for subjective/objective classification. WordNet consists of about 13,000 verb synsets, which can cover about 11,000 verbs. (As we mentioned in Section 2.1, since each sense of a word is in a different synset and a synset indicates a synonym set, about 11,000 verbs can be represented as about 13,000 verb synsets. For example, one of verb synsets is *wish, care, like (prefer or wish to do something)*. Even though it is a sense of each word (i.e., *wish*, *care*, and *like*), it is considered as one synset.) Moreover, synsets are interlinked by means of semantic relations. In addition, in Chapter 5, we presented the feasibility of using WordNet for +/-effect lexicon acquisition. Thus, we utilize WordNet in this work. With WordNet, we can cover most verbs and a small number of verb phrases.

Our another goals is to build sense-level +/-effect lexicon with a small number of seed data. For that, we first need annotated sense-level +/-effect events as a seed lexicon. The simple method to create a seed lexicon is to select synsets randomly from WordNet and annotate them. However, it is an inefficient way since it is hard to get reliable +/-effect

events. Because many cases are Null, we are not sure whether randomly selected synsets are reliable +/-effect events. Also, we want to create seed data that is independent from the corpus to preserve the corpus for evaluation. Therefore, we utilize a word-level seed lexicon built in Section 5.1. In this lexicon, an annotator who didn't have access to the corpus manually selected +/-effect events from FrameNet. It consists of 736 +effect lexical units and 601 -effect lexical units which are selected from 463 semantic frames in FrameNet. From this lexicon, we can gather 606 +effect verb words and 537 -effect verb words. However, we need a sense-level lexicon as a seed lexicon, not a word-level lexicon. Thus, we first extract all senses of these +/-effect words and annotate them. Section 6.1 explains our sense-level annotated data in detail. Then, before explaining our method, we describe our evaluation metrics in Section 6.2.

Next, we describe the method to construct EffectWordNet. In this chapter, we construct EffectWordNet, which is a sense-level +/-effect lexicon without the information about which entities are affected. As we mentioned in Section 2.1, WordNet provides two kinds of information: WordNet relations (e.g., hypernym, troponym, etc.) and gloss information (i.e., a short definition and usage examples). WordNet relations represent semantic relationship between synsets while gloss information provides information for each synset. We first present a graph-based semi-supervised learning method to utilize WordNet relations in Section 6.3. With a graph-based model, we investigate whether the +/-effect property tends to be shared among semantically-related synsets. Then, we develop a classifier for a gloss information in Section 6.4. To maximize the effectiveness of different types of information, we combine a graph-based method using WordNet relations and a standard classifier using gloss information in Section 6.5.

Further, we provide evidence that the model is an effective way to guide manual annotation to find +/-effect events that are not in the seed lexicon in Section 6.6.

Finally, related work is described in Section 6.7, and summary is given in Section 6.8.

This work is presented in Empirical Methods in Natural Language Processing (EMNLP) [Choi and Wiebe, 2014].

6.1 DATA

In this section, we describe data which are used in this chapter. We extracted word-level +/-effect events from FrameNet in Section 5.1. Since we need a sense-level lexicon in this work, we create a sense-level +/-effect lexicon based on this word-level lexicon.

6.1.1 Word-level +/-Effect Lexicon

In Section 5.1, we utilized FrameNet to select +/-effect events because we believed that using FrameNet to find +/-effect events is easier than finding +/-effect events without any information. By semantic frames, words may be filtered. The annotator selected 463 semantic frames for +/-effect events, and 736 +effect lexical units and 601 -effect lexical units were extracted from these semantic frames.

We first extract all words from 736 +effect lexical units and 601 -effect lexical units. In total, we gather 606 +effect words and 537 -effect words. Since one word can have more than one lexical unit, the number of words is smaller than the number of lexical units. Among them, 14 words (e.g., *crush, order,* etc.) are in both the +effect words and the -effect words. That is, these words have both +effect and -effect meanings. Recall that this annotator was focusing on frames, not on words - he did not look at all the senses of all the words. In Section 5.1, we ignored these 14 words for a purer lexicon. However, in this work, since we handle sense-level +/-effect events, not word-level +/-effect events, we do not ignore them.

6.1.2 Sense-level +/-Effect Seed Lexicon

As we mentioned, one of our goals is to build a sense-level +/-effect lexicon with a small number of seed data. Therefore, we first need a small number of sense-level +/-effect data as seed data. Moreover, we need sense-level +/-effect data for evaluations.

As we mentioned in the previous section, we created a word-level lexicon that consists of 606 +effect words and 537 -effect words, which were extracted from FrameNet. If we can convert them into WordNet automatically, it will be easy to create sense-level +/-effect data. However, mappings between FrameNet and WordNet are not perfect.

Thus, we opt to manually annotate the senses of the words in the word-level lexicon. We go through all senses of all the words in this word-level lexicon and manually annotate each sense as to whether it is +effect, -effect, or Null. Note that we conducted the agreement study for the sense-level +/-effect annotation and got 0.75 as κ and 0.84 as percent agreement which are positive results in Section 5.5.

In total, there are 258 +effect synsets, 487 -effect synsets, and 880 Null synsets. Since +/-effect words are extracted from 463 semantic frames in FrameNet, many senses are in the same synsets. Thus, the number of +/-effect synsets is smaller than the number of +/-effect words.

For the experiments in this work, we divide this annotated data into two equal-sized sets. One is a fixed test set that is used to evaluate both the graph model and the gloss classifier. The other set is used as a seed set by the graph model and as a training set by the gloss classifier. Table 5 shows the distribution of the data. Since the dataset is not big, we do not conduct the cross-validation.

Our task is to identify unlabeled senses that are likely to be +/-effect senses, so we want to focus on +effect and -effect classes rather the Null class. Since the Null class is the majority class based on this annotated data, we need to resize the Null class to avoid it becoming the majority class. To avoid too large a bias toward the Null class, we randomly chose half (i.e., the Null set contains 440 synsets). Half of each set is used as seed data in the graph model and training data in the classifier, and the other half is used for evaluation. All experiments except the last table in Section 6.6 give results on the same fixed test set.

	+effect	**-effect**	**Null**
# Annotated data	258	487	880
# Seed/TrainSet	129	243	220
# TestSet	129	244	220

Table 5: Distribution of annotated sense-level +/-effect seed data.

6.1.3 Data for Guided Annotation

In Section 6.6, the initial seed set is the same as Seed/TraingSet in Table 5. In each iteration, new data (i.e., verb synsets) that are not in Seed/TrainSet and TestSet are extracted by the graph-based model. Then, we manually annotate them and add them to the seed set. Table 6 shows the number of top 5% newly extracted +/-effect data for each iteration. In this work, we perform four iterations.

	1st	**2nd**	**3rd**	**4th**
+effect	128	122	116	117
-effect	155	146	153	145
total	283	268	269	262

Table 6: Frequency of the top 5% for each iteration.

6.2 EVALUATION METRICS

To evaluate our system, we calculate the accuracy that is the degree of closeness of detected value to an actual or correct value. It is calculated as follows:

$$Accuracy = \frac{\text{Number of correctly detected synsets}}{\text{Number of all synsets in test data}} \tag{6.1}$$

However, with the accuracy, we cannot evaluate the performance for each label. For example, if there is a predominant class, the base rate is close to the accuracy of predicting the predominant class. In this case, even though the performances for other labels that are

not predominant labels are not good, the accuracy can be high. In our task, not only the accuracy but also the performance for each label is important. Thus, to evaluate for each label, we calculate precision, recall, and f-measure for all three labels.

The precision presents how many of detected instances are correct in each label. It is also called as positive predictive value. The precision for a given label is calculated as:

$$Precision_{label} = \frac{\text{Number of correctly detected synsets as a given label}}{\text{Number of all synsets detected as a given label}} \tag{6.2}$$

On the other hand, the recall indicates how many of relevant instances for each label is detected by the system. The recall is measured as follows:

$$Recall_{label} = \frac{\text{Number of correctly detected synsets as a given label}}{\text{Number of all synsets of a given label in test data}} \tag{6.3}$$

These two measures can be used together in the f-measure to provide a single measurement such as:

$$\text{F-measure}_{label} = 2 \cdot \frac{Precision_{label} \cdot Recall_{label}}{Precision_{label} + Recall_{label}} \tag{6.4}$$

We use these metrics for all experiments except the last table in Section 6.6.

6.3 GRAPH-BASED SEMI-SUPERVISED LEARNING FOR WORDNET RELATIONS

WordNet, described in Section 2.1, is organized by semantic relations such as *hypernymy*, *troponymy*, *verb grouping*, and so on. These semantic relations can be used to build a network. Since the most frequently encoded relation is the super-subordinate relation, most verb synsets are arranged into hierarchies; verb synsets towards the bottom of the graph express increasingly specific manner. Thus, by following this hierarchical information, we hypothesize that +/-effect polarity tends to propagate. Thus, to carry out the label propagation, we adopt a graph-based semi-supervised learning method described in Section 2.3.1.

6.3.1 Graph Formulation

We formulate a graph for semi-supervised learning as follows. Let $G = \{X, E, W\}$ be the undirected graph in which X is the set of nodes, E is the set of edges (i.e., E_{ij} is the edge between the node i and j), and W represents the edge weights (i.e., the weight of edge E_{ij} is W_{ij}). The weight matrix is a non-negative matrix.

Each data point in $X = \{x_1, \ldots, x_n\}$ is one synset. The labeled data of X is represented as $X_L = \{x_1, \ldots, x_l\}$ and the unlabeled data is represented as $X_U = \{x_{l+1}, \ldots, x_n\}$. The labeled data X_L is associated with labels $Y_L = \{y_1, \ldots, y_l\}$, where $y_i \in \{1, ..., c\}$ (c is the number of classes). As is typical in such settings, $l \ll n$: n is 13,767, i.e., the number of verb synsets in WordNet. Seed/TrainSet in Table 5 is the labeled data.

To connect two nodes, WordNet relations are utilized. We first connect nodes by the hierarchical relations. Since *hypernym* relations represent more general synsets and *troponym* relations represent more specific verb synsets, we hypothesize that hypernyms or troponyms of a verb synset tends to have its same polarity. *Verb groups* relations that represent verb synsets having a similar meaning are also promising. Even though verb group coverage is not large, its relations are reliable since they are manually grouped. The *entailment* relation is defined as the verb Y is entailed by X if you must be doing Y by doing X. Since pairs connected by this relation are co-extensive, we can assume that both are the same type

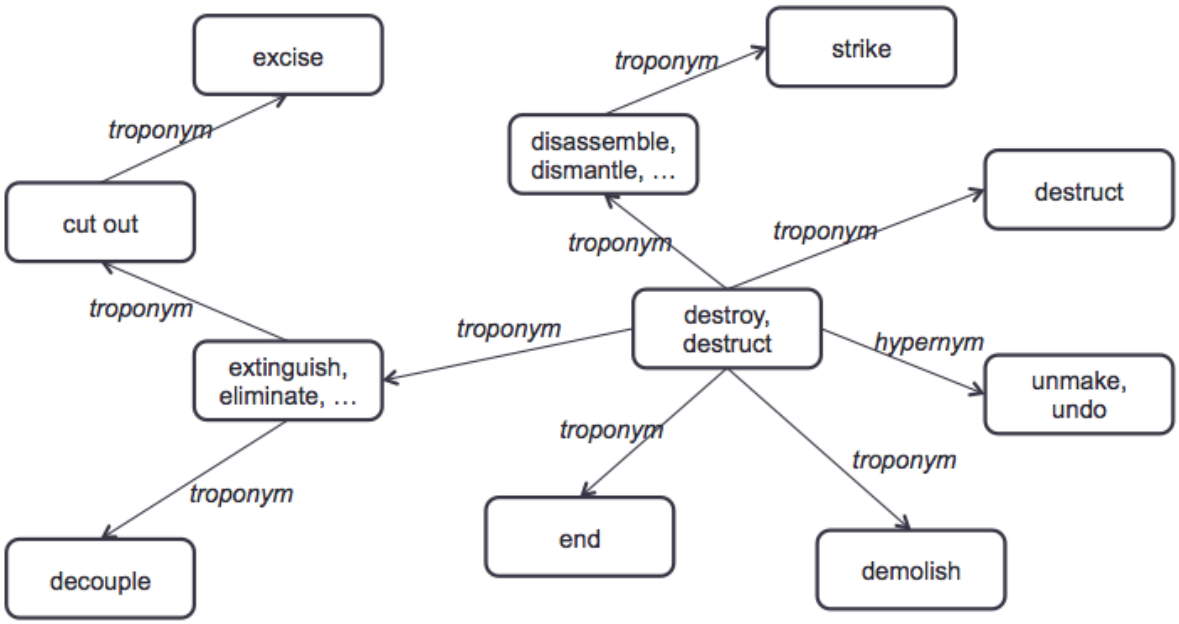


Figure 6: Part of constructed graph.

of event. The *synonym* relation is not used because it is already defined in synsets (i.e., each node in the graph is a synset), and the *antonym* relation is also not applied since WordNet doesn't provide any antonym relations for verbs. The weight value of all edges is 1.0. (Actually, we tried to set different weights for each relation, but there is no big difference. Thus, we finally give 1.0 as the weight value for all edges.) Figure 6 shows a part of the constructed graph.

We can apply the graph model in two ways. One way is that all three classes (+effect, -effect, and Null) are represented in one graph. That is, if a node is +effect, it has +1 value; if a node is -effect, it has -1 value; and if a node is Null, it has 0 value. We call such graph model **UniGraph4Rel**.

Another way is that two separate graphs are first constructed and then combined. One graph is for classifying +effect and Other (i.e., -effect or Null). This graph is called *+eGraph*. That is, if a node is +effect, it has +1 value; and if a node is -effect or Null, it has -1 value. The other graph, called *-eGraph*, is for classifying -effect and Other (i.e., +effect or Null).

That is, if a node is -effect, it has +1 value; and if a node is +effect or Null, it has -1 value. Since we are interested in +/-effect events, not Null, we build two separate graphs for +/-effect.

We have two motivations for experimenting with the two separate graphs: (1) SVM, the supervised learning method used for gloss classification (we describe this in the next section), tends to have better performance on binary classification tasks, and (2) the two graphs of the combined model can "negotiate" with each other via constraints.

There are two methods to combine two separate graphs into one model. One is **BiGraphSim4Rel** that the label is simply determined by two separate graphs as follows.

- Nodes that are labeled as +effect by +eGraph and Other by -eGraph are regarded as +effect, and nodes that are labeled as -effect by -eGraph and Other by +eGraph are regarded as -effect.
- If nodes are labeled as +effect by +eGraph and -effect by -eGraph, they are deemed to be Null.
- Nodes that are labeled Other by both graphs are also considered as Null.

Another method is to add constraints when determining the class. This is one of our motivations to build two separate graphs. With constraints, we expect to improve the results since two separate graphs can negotiate with each other. This approach is called **BiGraphConst4Rel**. As we explained, the label of instance x_i is determined by F_i in the graph. When the label of x_i is decided to be j, we can say that its confidence value is F_{ij}. There are two constraints as follows.

- If a sense is labeled as +effect (-effect), but the confidence value is less than a threshold, we count it as Null.
- If a sense is labeled as both +effect and -effect by BiGraph4Rel, we choose the label with the higher confidence value only if the higher one is larger than a threshold and the lower one is less than a threshold.

The thresholds are determined on Seed/TrainSet by running several times with different thresholds, and choosing the one that gives the best performance on Seed/TrainSet. In this work, the chosen value is 0.025 for +effect and 0.03 for -effect.

6.3.2 Label Propagation

Given a constructed graph, the label inference (or prediction) task is to propagate the seed labels to the unlabeled nodes. One of the classic graph-based semi-supervised learning label propagation methods is the local and global consistency (LGC) method suggested by [Zhou et al., 2004]. The LGC method is a graph transduction algorithm which is sufficiently smooth with respect to the intrinsic structure revealed by known labeled and unlabeled data. The cost function typically involves a tradeoff between the smoothness of the predicted labels over the entire graph and the accuracy of the predicted labels in fitting the given labeled nodes X_L. LGC fits in a univariate regularization framework, where the output matrix is treated as the only variable in optimization, and the optimal solutions can be easily obtained by solving a linear system. Thus, we adopt the LGC method in this work. Although there are some robust graph-based semi-supervised learning methods for handling noisy labels, we do not need to handle noisy labels because our input is the annotated data.

Let F be a $n \times c$ matrix to save the output values of label propagation. Thus, after the label propagation, we can label each instance x_i such as:

$$y_i = \text{argmax}_{j \leq c} F_{ij} \tag{6.5}$$

The initial discrete label matrix Y, which is also $n \times c$, is defined as:

$$Y_{ij} = \begin{cases} 1 & \text{if } x_i \text{ is labeled as } y_i = j \text{ in } Y_L \\ 0 & \text{otherwise} \end{cases} \tag{6.6}$$

The vertex degree matrix $D = \text{diag}([D_{11}, ..., Dnn])$ is defined by

$$D_{ii} = \sum_{j=1}^{n} W_{ij} \tag{6.7}$$

LGC defines the cost function Q which integrates two penalty components, global smoothness and local fitting (μ is the regularization parameter):

$$Q = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \| \frac{F_i}{\sqrt{D_{ii}}} - \frac{F_j}{\sqrt{D_{jj}}} \|^2 + \mu \sum_{i=1}^{n} \|F_i - Y_i\|^2 \tag{6.8}$$

The first part of the cost function is the *smoothness constraint*: a good classifying function should not change too much between nearby points. That is, if x_i and x_j are connected with an edge, the difference between them should be small. The second is the *fitting constraint*: a good classifying function should not change too much from the initial label assignment. The final label prediction matrix F can be obtained by minimizing the cost function Q.

6.3.3 Experimental Results

Note that, we conduct our experiments on the fixed test set (TestSet in Table 5).

Since there is no task to create +/-effect lexicon previously, we adopt the majority class classifier as a baseline system. That is, all synsets are classified into -effect events because -effect is the majority class in our test set based on Table 5.

Table 7 shows precision, recall, and f-measure for all three classes and accuracy. The top row shows the accuracy of the baseline (i.e., the majority class classifier). It shows the results of UniGraph4Rel, BiGraphSim4Rel, and BiGraphConst4Rel when they are built using the *hypernym*, *troponym*, and *verb group* relations. We will present why we choose these three relations with ablation results later.

		UniGraph4Rel	BiGraphSim4Rel	BiGraphConst4Rel
Baseline-Accuracy		0.411		
Accuracy		0.630	0.623	**0.658**
+effect	Precision	0.621	0.610	0.642
	Recall	0.655	0.647	0.680
	F-measure	0.637	0.628	**0.660**
-effect	Precision	0.644	0.662	0.779
	Recall	0.720	0.677	0.612
	F-measure	0.680	0.670	**0.686**
Null	Precision	0.615	0.583	0.583
	Recall	0.516	0.550	0.695
	F-measure	0.561	0.561	**0.634**

Table 7: Results of UniGraph4Rel, BiGraphSim4Rel, and BiGraphConst4Rel.

Our suggested methods (i.e., UniGraph4Rel, BiGraphSim4Rel, and BiGraphConst4Rel) outperform the baseline based on the accuracy measure. Since the baseline is the majority baseline and the majority class is -effect in our data, the baseline has 0.411 as the precision, 1.000 as the recall, and 0.583 as the f-measure for the -effect label. However, it has 0.0 as the recall for other labels (and we cannot calculate the precision and the f-measure since there are no senses detected as +effect or Null). In comparison, even though the recall for the -effect label in our systems is lower then the baseline, our systems show higher performance on the others. Moreover, in the -effect label, although the recall in the baseline is higher, our systems show better performance with the precision. Thus, when considering the f-measure that reflects both the precision and the recall, our systems outperform the baseline system.

Interestingly, UniGraph4Rel shows better performance than BiGraphSim4Rel (i.e., constructing two separate graphs and combine them simply) on +effect and -effect labels although the difference is relatively small. However, when adding constraints to combine two separate graphs (i.e., BiGraphConst4Rel), it outperforms not only BiGraphSim4Rel but also UniGraph. Especially, in BiGraphConst4Rel, the recall for the Null class is considerably increased, showing that constraints not only help overall, but also are particularly important for detecting Null cases.

Table 8 gives ablation results, showing the contribution of each WordNet relation in BiGraphConst4Rel. With only hierarchical information (i.e., *hypernym* and *troponym* relations), it already shows good performance for all classes. However, they cannot cover some synsets. Among the 13,767 verb synsets in WordNet, 1,707 (12.4%) cannot be labeled because there are not sufficient hierarchical links to propagate polarity information. When adding the *verb group* relation, it shows improvement in both +effect and -effect. Especially, the recall for +effect and -effect is significantly increased. In addition, the coverage of the 13,767 verb synsets increases to 95.1%. For *entailment*, whereas adding it shows a slight improvement in +effect (and increases coverage by 1.1 percentage points), the performance is decreased a little bit in the -effect and Null classes. Since the average f-measure for all classes is the highest with *hypernym*, *troponym*, and *verb group* relations (not *entailment*), we only consider these three relations when constructing the graph.

		Hypernym + Troponym	**+ Verb group**	**+ Entailment**
+effect	Precision	0.653	0.642	0.651
	Recall	0.660	0.680	0.683
	F-measure	0.656	0.660	**0.667**
-effect	Precision	0.784	0.779	0.786
	Recall	0.547	0.612	0.604
	F-measure	0.644	**0.686**	0.683
Null	Precision	0.557	0.583	0.564
	Recall	0.735	0.695	0.691
	F-measure	**0.634**	**0.634**	0.621
Coverage		87.6%	95.1%	96.2%

Table 8: Effect of each relation in BiGraphConst4Rel.

6.4 SUPERVISED LEARNING APPLIED TO WORDNET GLOSSES

In WordNet, each synset contains a gloss consisting of a definition and optional example sentences. Since a gloss consists of several words and there are no direct links between glosses, we believe that a word vector representation is appropriate to utilize gloss information as in [Esuli and Sebastiani, 2006]. For that, we adopt an SVM classifier.

6.4.1 Features

Two different feature types are used.

- **Word Features**: The bag-of-words model is applied. We do not ignore stop words for several reasons. Since most definitions and examples are not long, each gloss contains a small number of words. Also, among them, the total vocabulary of WordNet glosses is not large. Moreover, some prepositions such as *against* are sometimes useful to determine the polarity of +/-effect.
- **Sentiment Features**: Some glosses of +effect (-effect) synsets contain positive (negative) words. For instance, the definition of {*hurt#4, injure#4*} is "cause damage or affect negatively." It contains a negative word, *negatively*. Since a given event may positively (negatively) affect entities, some definitions or examples already contain positive (negative) words to express this. Thus, as features, we check how many positive (negative) words a given gloss contains. To detect sentiment words, the subjectivity lexicon provided by [Wilson et al., 2005][1] is utilized.

6.4.2 Gloss Classifier

We have three classes, +effect, -effect, and Null. Since SVM shows better performance on binary classification tasks, we generate two binary classifiers, one (*+eClassifier*) to determine whether a given synset is +effect or Other (i.e., -effect or Null), and another (*-eClassifier*)

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

to classify whether a given synset is -effect or Other (i.e., +effect or Null). Then, they are combined as follows.

- Synsets that are labeled as +effect by +eClassifier and Other by -eClassifier are regarded as +effect, and synsets that are labeled as -effect by -eClassifier and Other by +eClassifier are regarded as -effect.
- If synsets are labeled as +effect by +eClassifier and -effect by -eClassifier, they are deemed to be Null.
- Synsets that are labeled Other by both classifiers are also considered as Null.

We call such method **Classifier4Gloss** since it is a classifier considering only gloss information as features.

6.4.3 Experimental Results

Seed/TrainSet in Table 5 is used to train two classifiers, and TestSet is utilized for the evaluation. That is, the training set for +eClassifier consists of 129 +effect instances and 463 Other instances (i.e., -effect and Null), and the training set for -eClassifier contains 243 -effect instances and 349 Other instances (i.e., +effect and Null). As a baseline, we adopt a majority class classifier such as the previous one.

Table 9 shows the results of Classifier4Gloss with the ablation study. Recall that the baseline has 0.411 as the precision, 1.000 as the recall, and 0.583 as the f-measure for the -effect label. However, it has 0.0 as the recall for other labels. The second column in Table 9 is the result of Classifier4Gloss. As you can see, Classifier4Gloss shows better performance than the baseline system except recall and f-measure of the -effect label.

Interestingly, performance is better for the -effect than for the +effect class, perhaps because the -effect class has more instances. Moreover, when sentiment features are added, all metric values increase, providing evidence that sentiment features are helpful to determine +/-effect classes.

		Word Features	**Word Features + Sentiment Features**
Baseline accuracy		0.411	
Accuracy		0.509	**0.539**
+effect	Precision	0.541	0.588
	Recall	0.354	0.393
	F-measure	0.428	**0.472**
-effect	Precision	0.616	0.672
	Recall	0.500	0.511
	F-measure	0.552	**0.580**
Null	Precision	0.432	0.451
	Recall	0.612	0.657
	F-measure	0.507	**0.535**

Table 9: Results of Classifier4Gloss with the ablation study.

6.5 HYBRID METHOD

To use more combined knowledge, BiGraphConst4Rel and Classifier4Gloss can be combined. That is, the classifier is utilized for WordNet gloss information and the graph model is adopted for WordNet relations. This method is called **Hybrid4AllFea**. With this method, we can see not only the effect of propagation by WordNet relations but also the usefulness of gloss information and sentiment features. Also, while BiGraphConst4Rel cannot cover all verb synsets in WordNet because a few numbers of synsets do not have any relation information, Hybrid4AllFea can cover all verb synsets because the classifier can handle all synsets.

The outputs of BiGraphConst4Rel and Classifier4Gloss are combined as follows. The label of Classifier4Gloss is one of +effect, -effect, Null, or Both (when a given synset is classified as both +effect by +eClassifier and -effect by -eClassifier). Possible labels of BiGraphConst4Rel are +effect, -effect, Null, Both, or None (when a given synset is not labeled by BiGraphConst4Rel). There are five rules:

- If both labels are +effect (-effect), it is +effect (-effect).
- If one of them is Both and the other is +effect (-effect), it is +effect (-effect).
- If the label of BiGraphConst4Rel is None, believe the label of Classifier4Gloss
- If both labels are Both, it is Null
- Otherwise, it is Null

6.5.1 Experimental Results

Note that Seed/TrainSet in Table 5 is used for seed data in BiGraphConst4Rel and training data in Classifier4Gloss, and TestSet is utilized for the evaluation.

The results for Hybrid4AllFea are given in Table 10; the results for BiGraphConst4Rel and Classifier4Gloss are in the first and second columns for comparison. For the +effect and -effect labels, Hybrid4AllFea shows better performance than BiGraphConst4Rel and

		BiGraphConst4Rel	Classifier4Gloss	Hybrid4AllFea
+effect	Precision	0.642	0.588	0.610
	Recall	0.680	0.393	0.735
	F-measure	0.660	0.472	**0.667**
-effect	Precision	0.779	0.672	0.717
	Recall	0.612	0.511	0.669
	F-measure	0.686	0.580	**0.692**
Null	Precision	0.583	0.451	0.556
	Recall	0.695	0.657	0.520
	F-measure	**0.634**	0.535	0.538

Table 10: Results of BiGraphConst4Rel, Classifier4Gloss and Hybrid4AllFea.

Classifier4Gloss. In Hybrid4AllFea, since more +/-effect synsets are detected than by BiGraphConst4Rel, while the precision is decreased, the recall is increased by more. However, by the same token, the overall performance for the Null class is decreased. Actually, that is expected since the Null class is determined by the Other class in BiGraphConst4Rel and Classifier4Gloss. Through this experiment, we can see that the hybrid method is better for classifying +/-effect synsets, but not for Null.

6.5.2 Model Comparison

To provide evidence for our assumption that different models are needed for different information to maximize effectiveness, we compare Hybrid4AllFea with the supervised learning and the graph-based learning methods, each utilizing both WordNet relations and gloss information.

Supervised Learning (Classifier4AllFea): Classifier4Gloss is trained with word features and sentiment features for WordNet gloss information. To exploit WordNet relations (especially, the hierarchical information) in the supervised learning method, we use least common subsumer (LCS) values as in [Gyamfi et al., 2009], which were utilized for the supervised learning method of subjective/objective synsets. The values are calculated as follows. For a target sense t and a seed set S, the maximum LCS value between a target sense and a member of the seed set is found as:

$$Score(t, S) = max_{s \in S} LCS(t, s) \tag{6.9}$$

With this LCS feature and the features utilized in Classifier4Gloss, we run SVM on the same training and test data. That is, the difference between Classifier4Gloss and Classifier4AllFea is features; while Classifier4Gloss considers features for only gloss information (i.e., word features and sentiment features), Classifier4AllFea considers features for both gloss information and WordNet relations (i.e., word features, sentiment features, and LCS features) For LCS values, the similarity using the information content proposed by [Resnik, 1995] is measured. WordNet Similarity[2] package provides pre-computed pairwise similarity values for that.

Table 11 shows results of Classifier4AllFea in the last column. The results for Classifier4Gloss and Hybrid4AllFea are in the first and second columns for comparison. Compared to Classifier4Gloss, while the +effect and Null classes show a slight improvement, the performance is degraded for the -effect class. It means that the added feature (i.e., LCS feature for WordNet relation information) in the classifier is rather harmful to the -effect class. Even though the hierarchical feature is very helpful to expand +/-effect in the graph model as we presented in Section 6.3, it is not helpful in the classifier method since the classifier cannot capture a propagation according to the hierarchy.

[2]WordNet Similarity, http://wn-similarity.sourceforge.net/

		Classifier4Gloss	Hybrid4AllFea	Classifier4AllFea
+effect	Precision	0.588	0.610	0.584
	Recall	0.393	0.735	0.400
	F-measure	0.472	**0.667**	0.475
-effect	Precision	0.672	0.717	0.778
	Recall	0.511	0.669	0.316
	F-measure	0.580	**0.692**	0.449
Null	Precision	0.451	0.556	0.440
	Recall	0.657	0.520	0.813
	F-measure	0.535	0.538	**0.571**

Table 11: Comparison to Classifier4Gloss, Hybrid4AllFea, and Classifier4AllFea.

Moreover, Hybrid4AllFea outperforms Classifier4AllFea for the +effect and -effect labels. Although Classifier4AllFea shows better performance in the Null class, it is a slight improvement. Both Hybrid4AllFea and Classifier4AllFea utilize WordNet relations and gloss information. The different thing is that the graph model is utilized for WordNet relations in Hybrid4AllFea while the classifier is used for relation information in Classifier4AllFea. As you can see, the results are totally different according to which method is utilized for WordNet relation information. Through this experiment, we can know that the graph-based model is appropriate for WordNet relation information.

Graph-based Learning (BiGraph4AllFea): In Section 6.3, the graph is constructed by using WordNet relations. To apply WordNet gloss information in the graph model, we calculate a cosine similarity between glosses. If the similarity value is higher than a threshold, two nodes are connected with this similarity value. The threshold is determined by training and testing on Seed/TrainSet (the chosen value is 0.3).

		BiGraphConst4Rel	Hybrid4AllFea	BiGraph4AllFea
+effect	Precision	0.642	0.610	0.701
	Recall	0.680	0.735	0.364
	F-measure	0.660	**0.667**	0.480
-effect	Precision	0.779	0.717	0.651
	Recall	0.612	0.669	0.562
	F-measure	0.686	**0.692**	0.603
Null	Precision	0.583	0.556	0.473
	Recall	0.695	0.520	0.679
	F-measure	**0.634**	0.538	0.557

Table 12: Comparison to BiGraphConst4Rel, Hybrid4AllFea, and BiGraph4AllFea.

Table 12 shows results of BiGraph4AllFea in the last column. The results for BiGraph-Const4Rel and Hybrid4AllFea are in the first and second columns for comparison. BiGraph-Const4Rel outperforms BiGraph4AllFea (the exception is the precision of +effect). By gloss similarity, many nodes are connected to each other. However, since uncertain connections can cause incorrect propagation in the graph, this negatively affects the performance.

Compared to Hybrid4AllFea, generally Hybrid4AllFea shows better performance than BiGraph4AllFea for the +effect and -effect labels (the exception is the precision of +effect). Although BiGraph4AllFea shows better performance in the Null class, it is a slight improvement. Both methods utilize all features (i.e., WordNet relations and gloss information). The difference between them is that the classifier is adopted for gloss information in Hybrid4AllFea while the graph model is adopted for gloss information in BiGraph4AllFea. This experiment shows that the classifier is proper for gloss information in our task.

Through these experiments, we see that since each type of information has a different character, we need different models to maximize the effectiveness of each type. Thus, the hybrid method with different models can have better performance.

6.6 GUIDED ANNOTATION

Recall that Seed/TrainSet and TestSet in Table 5, the data used so far, are all the senses of the words in a word-level +/-effect lexicon. This section presents evidence that our method can guide annotation efforts to find other words that have +/-effect senses. A bonus is that the method pinpoints particular +/-effect senses of those words.

All unlabeled data are senses of words that are not included in the original lexicon. Since presumably the majority of verbs do not have any +/-effect senses, a sense randomly selected from WordNet is very likely to be Null. However, we are more interested in the +effect and -effect labels than the Null label. Thus, we don't want the random selection since we want to find more +/-effect events.

To handle this problem, we explore an iterative approach to guided annotation, using BiGraphConst4Rel and Hybrid4AllFea as the method for assigning labels. (Since BiGraph-Const4Rel and Hybrid4AllFea show good performance in our previous experiments, we adopt these two models for guided annotation.) The system is initially created as described above using Seed/TrainSet as the initial seed set. Each iteration has four steps:

1. Rank all unlabeled data (i.e., the data other than TestSet and the current seed set) based on the F_{ij} confidence values (see Section 6.3.3).
2. Choose the top 5% and manually annotate them (the same annotator as above did this).
3. Add them to the seed set.
4. Rerun the system using the expanded seed set. (We performed four iterations in this work.)

Table 13 shows the initial results (i.e., the same result of BiGraphConst4Rel in Table 7) and the results after each iteration with BiGraphConst4Rel; and Table 14 shows the initial results (i.e., the same result of Hybrid4AllFea in Table 10) and the results after each iteration with Hybrid4AllFea. We calculate precision, recall, and f-measure for each label. Recall that these are results on the fixed test set, TestSet in Table 5.

		BiGraphConst4Rel				
		Initial	**1st**	**2nd**	**3rd**	**4th**
+effect	Precision	0.642	0.636	0.642	0.636	0.681
	Recall	0.680	0.684	0.701	0.708	0.674
	F-measure	0.660	0.663	0.670	0.670	**0.678**
-effect	Precision	0.779	0.770	0.748	0.779	0.756
	Recall	0.612	0.632	0.656	0.652	0.674
	F-measure	0.686	0.694	0.699	0.710	**0.712**
Null	Precision	0.583	0.591	0.605	0.599	0.589
	Recall	0.695	0.672	0.655	0.669	0.669
	F-measure	**0.634**	0.629	0.629	0.632	0.626

Table 13: Results of an iterative approach for BiGraphConst4Rel.

		Hybrid4AllFea				
		Initial	**1st**	**2nd**	**3rd**	**4th**
+effect	Precision	0.610	0.614	0.613	0.616	0.688
	Recall	0.735	0.713	0.743	0.739	0.681
	F-measure	0.667	0.672	0.672	0.672	**0.684**
-effect	Precision	0.717	0.728	0.716	0.717	0.712
	Recall	0.669	0.681	0.697	0.706	0.764
	F-measure	0.692	0.704	0.706	0.712	**0.732**
Null	Precision	0.556	0.562	0.559	0.559	0.565
	Recall	0.520	0.523	0.497	0.494	0.527
	F-measure	0.538	0.542	0.526	0.525	**0.545**

Table 14: Results of an iterative approach for Hybrid4AllFea.

Overall for both models, the f-measure increases for both the +effect and -effect classes as more seeds are added, mainly due to improvements in recall. The evaluation on the fixed set is also useful in the annotation process because it trades off +/-effect vs. Null annotations. If the new manual annotations were biased, in that they incorrectly label Null senses as +/-effect, then the f-measure results would instead degrade on the fixed TestSet, since the system is created each time using the increased seed set.

We now consider the accuracy of the system on the newly labeled annotated data in Step 2. Note that our method is similar to Active Learning [Tong and Koller, 2001], in that both automatically identify which unlabeled instances the human should annotate next. However, in active learning, the goal is to find instances that are difficult for a supervised learning system. In our case, the goal is to find needles in the haystack of WordNet senses. In Step 3, we add the newly labeled senses to the seed set, enabling the model to find unlabeled senses close to the new seeds when the system is rerun for the next iteration.

We assess the system's accuracy on the newly labeled data by comparing the system's labels with the annotator's new labels. In this case, the evaluation matrix is different with previous experiments since the purpose is different. While we evaluate suggested systems with the same fixed test data (i.e., TestSet in Table 5) in previous experiments, we want to estimate the performance of our proposed systems with the newly labeled data by the system which is different each iteration. The accuracy for the +effect and -effect labels is calculated such as:

$$Accuracy_{+effect} = \frac{\text{\# annotated +effect}}{\text{\# top 5\% +effect data}} \tag{6.10}$$

$$Accuracy_{-effect} = \frac{\text{\# annotated -effect}}{\text{\# top 5\% -effect data}} \tag{6.11}$$

	1st	2nd	3rd	4th
+effect	65.63%	62.50%	63.79%	59.83%
-effect	73.55%	73.97%	77.78%	70.30%
+effect	128	122	116	117
-effect	155	146	153	145
total	283	268	269	262

Table 15: Accuracy and frequency of the top 5% for each iteration.

That is, the accuracy means that out of the top 5% of the +effect (-effect) data as scored by the system, what percentage are correct as judged by a human annotator. Table 15 shows the accuracy for each iteration in the top part and the number of senses labeled in the bottom part. As can be seen, the accuracies range between 60% and 78%; these values are much higher than what would be expected if labeling senses of words randomly chosen from WordNet and are **not** in the original seed lexicon.

The annotator spent, on average, approximately an hour to label 100 synsets. For finding new words with +/-effect usages, it would be much more cost-effective if a significant percentage of the data chosen for annotation are senses of words that in fact have +/-effect senses. Based on this method, we will continue to annotate +/-effect events for creating evaluation data.

6.7 RELATED WORK

Lexicons are widely used in sentiment analysis and opinion mining. Several works such as [Hatzivassiloglou and McKeown, 1997], [Turney and Littman, 2003], [Kim and Hovy, 2004], [Strapparava and Valitutti, 2004], and [Peng and Park, 2011] have tackled automatic lexicon

expansion or acquisition. However, in most such work, the lexicons are word-level rather than sense-level.

For the related (but different) tasks of developing subjectivity, sentiment and connotation lexicons, some do take a sense-level approach. [Esuli and Sebastiani, 2006] construct SentiWordNet. They assume that terms with the same polarity tend to have similar glosses. So, they first expand a manually selected seed set of senses using WordNet lexical relations such as *also-see* and *direct antonymy* and train two classifiers, one for positive and another for negative. As features, a vector representation of glosses is adopted. These classifiers are applied to all WordNet senses to measure positive, negative, and objective scores. In extending their work [Esuli and Sebastiani, 2007], the PageRank algorithm is applied to rank senses in terms of how strongly they are positive or negative. In the graph, each sense is one node, and two nodes are connected when they contain the same words in their WordNet glosses. Moreover, a random-walk step is adopted to refine the scores in their recent work [Baccianella et al., 2010]. In contrast, our approach uses WordNet relations and graph propagation in addition to gloss classification.

[Gyamfi et al., 2009] construct a classifier to label the subjectivity of word senses. The hierarchical structure and domain information in WordNet are exploited to define features in terms of similarity (using the LCS metric in [Resnik, 1995]) of target senses and a seed set of senses. Also, the similarity of glosses in WordNet is considered. Even though they investigated the hierarchical structure by LCS values, WordNet relations are not exploited directly.

[Su and Markert, 2009] adopt a semi-supervised mincut method to recognize the subjectivity of word senses. To construct a graph, each node corresponds to one WordNet sense and is connected to two classification nodes (one for subjectivity and another for objectivity) via a weighted edge that is assigned by a classifier. For this classifier, WordNet glosses, relations, and monosemous features are considered. Also, several WordNet relations (e.g., *antonymy*, *similar-to*, *direct hypernym*, etc.) are used to connect two nodes. Although they make use of both WordNet glosses and relations, and gloss information is utilized for a classifier, this classifier is generated only for weighting edges between sense nodes and classification nodes, not for classifying all senses.

[Goyal et al., 2010] generate a lexicon of patient polarity verbs (PPVs) that impart positive or negative states on their patients. They harvest PPVs from a Web corpus by co-occurrence with Kind and Evil agents and by bootstrapping over conjunctions of verbs. [Riloff et al., 2013] learn positive sentiment phrases and negative situation phrases from a corpus of tweets with hashtag "sarcasm". However, both of these methods are word-level rather than sense-level.

[Feng et al., 2011] build connotation lexicons that list words with connotative polarity and connotative predicates that exhibit selectional preference on the connotative polarity of some of their semantic argument. To learn connotation lexicon and connotative predicates, they adopted a graph-based algorithm and an induction algorithm based on Integer Linear Programming. [Kang et al., 2014] present a unified model that assigns connotation polarities to both words and senses. They formulate the induction process as collective inference over pairwise-Markov Random Fields and apply loopy belief propagation for inference. Their approach relies on selectional preferences of *connotative predicates*; the polarity of a connotation predicate suggests the polarity of its arguments. We have not discovered an analogous type of predicate for the problem we address.

Ours is the first NLP research into developing a sense-level lexicon for events that have negative or positive effects on entities.

6.8 SUMMARY

In this chapter, we investigate methods for creating a sense-level +/-effect lexicon, called EffectWordNet. Due to significant sense ambiguity as we mentioned in Section 4.2.2, we develop a sense-level lexicon rather than a word-level lexicon. Also, as we mentioned in Section 4.2.3, we focus only verbs as +/-effect events in this work.

One of our goals is to develop the method that applied to many verb synsets. Also, another goal is to build a lexicon with a small number of seed data. In addition, we want to investigate whether the +/-effect property tends to be shared among semantically-related synsets.

As we mentioned in Section 6.1, we have a small number of annotated data. We have 258 +effect annotated verb synsets, 487 -effect synsets, and 440 Null synsets. Among them, half of each set is used as seed data in the graph-based model and training data in the classifier, and the other half is used for evaluation. In this work, we present that our method is promising even though the size of data is small.

We utilize WordNet resource with two assumptions: (1) each sense (or synset) has only one +/-effect polarity and (2) +/-effect polarity tends to propagate by semantic relations such as hierarchical information.

To utilize WordNet relations, we adopt a graph-based learning method in Section 6.3. Since we have three labels (e.g., +effect, -effect, and Null), there are two ways to build graphs; one way is to build one graph to represent all three labels (called UniGraph4Rel), and another way is to build two separate graphs (i.e., one for +effect and one for -effect) and combine them (called BiGraphSim4Rel). Also, when combining them, we can add constraints (called BiGraphConst4Rel). As the baseline system, we adopt the majority classifier (in this work, the majority class is -effect). As we presented in Table 7, our systems (UniGraph4Rel, BiGraphSim4Rel, and BiGraphConst4Rel) outperforms the baseline. While the baseline shows 0.411 as the accuracy, all our systems show over 0.6 as the accuracy. Moreover, even though UniGraph4Rel shows better performance than BiGraphSim4Rel (i.e., combining two separate graphs without any constraints), BiGraphConst4Rel (i.e., combining two separate graphs with constraints) shows the best performance. Through these experiments, we know that WordNet relations can be used for the polarity propagation. Moreover, constructing two separate graphs and combining them with constraints is better than building only one graph in our work. In addition, in BiGraphConst4Rel, the recall for the Null class is considerably increased, showing that constraints not only help overall, but also are particularly important for detecting Null cases.

For WordNet gloss information, we build a classifier with bag-of-word features and sentiment features called Classifier4Gloss in Section 6.4. Since +/-effect means events that have positive or negative effect on entities, some definitions or examples already contain positive or negative words to express a given event. In Table 9, we present that Classifier4Gloss outperform the baseline system. Also, in our experiment, it shows better performance in all

labels when considering sentiment words as features. It is evidence that sentiment features are helpful to determine +/-effect classes.

To maximize the effectiveness of each type of information, we combine a graph-based method using WordNet relations and a standard classifier using gloss information in Section 6.5. We call such method Hybrid4AllFea. As we presented in Table 10, Hybrid4AllFea gives the best results in +effect and -effect labels although the performance for the Null label is dropped. Moreover, we provide evidence for our assumption that different models are needed for different information to maximize effectiveness. In Table 11, we experiment with the supervised learning method that utilizes both WordNet relations and gloss information and present that the graph-based model is appropriate for WordNet relation information. In Table 12, we experiment with the graph-based learning method with not only WordNet relations but also gloss information and shows that the classifier is proper for gloss information in our task.

Overall, BiGraphConst4Rel shows good performance for all three classes. However, as we mentioned, we are more interested in the +effect and -effect labels than the Null label. Thus, when considering only the +effect and -effect labels, Hybrid4AllFea shows better performance.

Further, in Section 6.6, we provide evidence that the model is an effective way to guide manual annotation to find +/-effect words that are not in the seed word-level lexicon. This is important, as the likelihood that a random WordNet synset (and thus word) is +effect or -effect is not large.

7.0 ENHANCED EFFECTWORDNET

As we mentioned in Section 4.2.4, the information about which entities are affected is important since the sentiment can be different in opinion inferences. For instance, let's assume that the given event is -effect on the **theme**; then, if the writer's sentiment toward the event is positive, the sentiment toward the **theme** is negative and the sentiment toward the **agent** is positive by opinion inference rules in Section 4. On the other hand, if the given event is -effect on the **agent**, the sentiment toward the **agent** is negative on the assumption that the writer's sentiment toward the event is positive. Thus, depending on what the affected entities are, the sentiment toward the agent is different.

Consider the following:

carry:

S_1: (v) carry (win in an election) "The senator carried his home state"

⇒ **+Effect toward the agent**

S_2: (v) carry (keep up with financial support) "The Federal Government carried the province for many years"

⇒ **+Effect toward the theme**

S_3: (v) carry (capture after a fight) "The troops carried the town after a brief fight"

⇒ **-Effect toward the theme**

In the first sense, *carry* has a positive effect on the **agent**, *the senator*, and in the second sense, it has a positive effect on the **theme**, *the province*. Even though the polarity of +/-effect is the same as +effect, the affected entity is different. In the third sense, *carry* has a negative effect on the **theme**, *the town*, since it is captured by the troops.

Like *carry*, a word can have a mixture of +/-effect polarities with different affected entities. However, in Chapter 6, we didn't consider the information about which entities are affected. In EffectWordNet built in Chapter 6, the first and second senses of *carry* are considered as the same label (i.e., +effect), and the third one is regarded as the different label (i.e., -effect). However, as we mentioned, the sentiment can be different according to the information about which entities are affected. Thus, the first two senses of *carry* should have different labels. Of course, the third sense also should have a different label.

Moreover, events can have positive or negative effects on both the **theme** and the **agent** with same or different polarities as we mentioned in Section 4.2.4. Consider one sense (or synset) of *take*:

S: (v) take (take by force) "Hitler took the Baltic Republics"; "The army took the fort on the hill"

In this case, *took* has a positive effect on the **agent**, *Hitler* or *the army*, but it has a negative effect on the **theme**, *the Baltic Republics* or *the fort on the hill*. It should have a different label from three senses of *carry*; or it should have two labels such as one for the agent and another for the theme.

In this chapter, to handle these problems, we construct the enhanced sense-level +/-effect lexicon that considers the affected entities for opinion inferences. That is, we refine EffectWordNet with consideration of affected entities. We call such lexicon **Enhanced EffectWordNet**. As we mentioned in Section 4.2.4, other entities which are neither the agent nor the theme can also be affected entities. However, It is very rare. Thus, we only consider the theme and the agent as the affected entity in this chapter.

In Chapter 6, we created the sense-level +/-effect lexicon by combining a graph-based method for WordNet relations and a standard classifier for gloss information. Even though the hybrid method (Hybrid4AllFea) shows the best performance on +effect and -effect labels, generally the graph-based model (BiGraphConst4Rel) shows better performance for all three labels (i.e., +effect, -effect, and Null). Thus, we adopt this graph model, but in this chapter, we build four separate graphs for considering different types of affected entities.

First, we need seed data for the graph-based model. Even though we created sense-level +/-effect seed data in Chapter 6, this data didn't consider the information about which entities are affected. Thus, we conduct the additional annotation study to recognize what the affected entities are in Section 7.1. Then, we describe our evaluation metrics in Section 7.2. Next, we provide the framework in Section 7.3. As we mentioned, we build four separate graphs and combine them for considering different types of affected entities. The experiments and results are presented in Section 7.4. Finally, related work is described in Section 7.5 and summary is given in Section 7.6.

7.1 NEW ANNOTATION STUDY

In Chapter 6, we provided manually annotated +/-effect data. It consists of 258 +effect synsets, 487 -effect synsets, and 880 Null synsets. However, it only provided the label of +/-effect, not the information about which entities are affected. Thus, we conduct an additional annotation study to recognize what the affected entities are. Note that we conducted the agreement study for the annotation of agents and themes and got positive results in Section 4.1.1. (As we presented in Table 1, for the agent annotation, we got 0.92 and 0.87 with two different measures; and for the theme annotation, we got 1.00 and 0.97.)

Since these is no affected entity information for the Null label, we only conduct the additional annotation study for only synsets which are already annotated as +effect or -effect labels. Figure 7 presents diagrams of the distribution of which entities are affected for each label (i.e., +effect and -effect).

Based on this study, among +effect synsets, about 76.43% of events are +effect on the theme and about 20.15% of events are +effect on the agent; there is one case in which there is +effect on both the agent and the theme. About 3% of events are +effect on the other entity, not the agent nor the theme.

Also, among -effect synsets, about 88.89% of events are -effect on the theme and about 7.4% of events are -effect on the agent; about 1.85% of events are -effect on both the agent

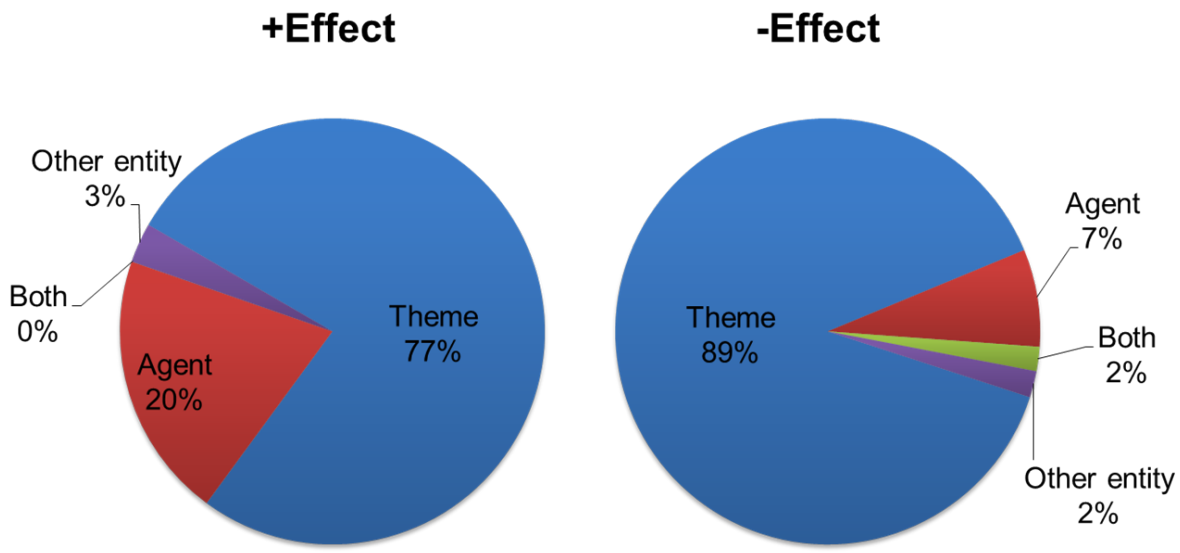


Figure 7: The distribution of which entities are affected for the +effect and -effect labels.

and the theme. About 2% of events are -effect on the other entity.

There are 16 instances which have positive or negative effects on both the agent and the theme with different polarities. Most instances are -effect on the theme and +effect on the agent such as *defeat*, *win*, and so on.

Even though affected entities can be neither the agent nor the theme, these are rare (i.e. about 3% for +effect events and about 2% for -effect events). Thus, this work focuses on +/-effect on the agent and +/-effect on the theme.

7.2 EVALUATION METRICS

As we mentioned in Chapter 6, the performance for each label is important in our task. Thus, to evaluate for each label, we calculate precision, recall, and f-measure for all three labels such as Section 6.2.

The precision presents how many of detected instances are correct in each label. It is also called as positive predictive value. The precision for a given label is calculated as:

$$Precision_{label} = \frac{\text{Number of correctly detected synsets as a given label}}{\text{Number of all synsets detected as a given label}} \tag{7.1}$$

On the other hand, the recall indicates how many of relevant instances for each label is detected by the system. The recall is measured as follows:

$$Recall_{label} = \frac{\text{Number of correctly detected synsets as a given label}}{\text{Number of all synsets of a given label in test data}} \tag{7.2}$$

These two measures can be used together in the f-measure to provide a single measurement such as:

$$\text{F-measure}_{label} = 2 \cdot \frac{Precision_{label} \cdot Recall_{label}}{Precision_{label} + Recall_{label}} \tag{7.3}$$

We use these metrics for all experiments in this chapter.

7.3 FRAMEWORK

For creating sense-level +/-effect lexicon considering the affected entity, We adopt the same graph-based model described in Chapter 6. Even though the hybrid method (Hybrid4AllFea)

shows the best performance on +effect and -effect labels, generally the graph-based model (BiGraphConst4Rel) shows better performance for all three classes (i.e., +effect, -effect, and Null). Thus, in this chapter, we focus on the graph-based model.

Our task is slightly different from the previous task in Chapter 6. While the previous task has three labels (i.e., +effect, -effect, Null), we have more labels because we have to consider the information about which entities are affected.

The simply method is to build one graph to handle all cases. In this case, we have nine labels such as +/-effect on the theme, +/-effect on the agent, +/-effect on both the agent and the theme, +effect on the theme but -effect on the agent, -effect on the theme but +effect on the agent, and Null.

However, since it has lots of labels, it is hard to consider all cases in one graph. Moreover, in Chapter 6, we already presented that building two separate graphs (i.e., one for +effect and another for -effect) and combining them with constraints is better than building one graph for three labels (i.e., +effect, -effect, and Null). Thus, in this work, we build several separate graphs and combine them.

First, we define five labels such as:

- *+effectTheme*: +effect on the theme
- *-effectTheme*: -effect on the theme
- *+effectAgent*: +effect on the agent
- *-effectAgent*: -effect on the agent
- Null

While we assumed that each synset can have only one label in the previous work, in this work, each synset can have one or two labels according to rules; for example, *take (take by force)* has two labels - *+effectAgent* and *-effectTheme*. (Of course, one synset cannot have both +effectTheme and -effectTheme or both +effectAgent and -effectAgent because it is a conflict. Also, the Null label cannot coexist with other labels; that is, if a given synset has the Null label, it should have only one label, Null, and it cannot have another label.)

To handle these five labels and to allow one or two labels for each synset, we build four separate graphs such as:

- *+eTGraph*: Classifying *+effectTheme* and Other (i.e., other four labels).
- *-eTGraph*: Classifying *-effectTheme* and Other.
- *+eAGraph*: Classifying *+effectAgent* and Other.
- *-eAGraph*: Classifying *-effectAgent* and Other.

With four separate graphs, not only can we consider all five labels (i.e., +effectTheme, -effectTheme, +effectAgent, -effectAgent, and Null), but also we can handle a case in which an event has positive or negative effects on both the agent and the theme with same or different polarities. In addition, we can provide different features for each label although we don't present it in this work.

Then, we combine these four separate graphs as follows:

- If nodes are labeled as *+effectTheme* (or *-effectTheme*) by +eTGraph (or -eTGraph) and Other by -eTGraph (or +eTGraph),
  → They are regarded as *+effectTheme* (or *-effectTheme*).
- If nodes are labeled as *+effectAgent* (or *-effectAgent*) by +eAGraph (or -eAGraph) and Other by -eAGraph (or +eAGraph),
  → They are regarded as *+effectAgent* (or *-effectAgent*).
- If nodes are labeled as *+effectTheme* (or *-effectTheme*) by +eTGraph (or -eTGraph) and *-effectTheme* (or *+effectTheme*) by -eTGraph (or +eTGraph),
  → They are deemed to be Null.
- If nodes are labeled as *+effectAgent* (or *-effectAgent*) by +eAGraph (or -eAGraph) and *-effectAgent* (or *+effectAgent*) by -eAGraph (or +eAGraph),
  → They are deemed to be Null.
- If nodes are labeled as Other by all graphs,
  → They are considered as Null.

7.4 EXPERIMENTAL RESULTS

For seed data of the graph model and data for evaluation, we use the annotated data in Section 7.1. We divide this data into two equal-sized sets: one for seed data and another for evaluation such as the previous work in Chapter 6. As we mentioned in Section 6.1, we want to focus on +effect and -effect labels rather the Null label. Since the Null class is the majority class based on annotated data, we need to resize the Null class to avoid it becoming the majority class. To avoid too large a bias toward the Null class, we randomly chose half (i.e., the Null set contains 440 synsets).

To evaluate our system, we calculate precision, recall, and f-measure for each label described in Section 7.2. Table 16 presents the results of our suggested model (Enhanced EffectWordNet). As a baseline system, we adopt the graph-based model (BiGraphConst4Rel) in Chapter 6. Table 17 shows the results of BiGraphConst4Rel. The +effectTheme and +effectAgent labels should be compared with the +effect label since +effectTheme and +effectAgent are the part of +effect; the -effectTheme and -effectAgent labels should be compared with the -effect label.

Compared to BiGraphConst4Rel, Enhanced EffectWordNet shows better performance except -effectAgent. Especially, while the precision of +effect in BiGraphConst4Rel is 0.642, the precision of the +effectTheme and +effectAgent labels is higher (i.e., 0.739 for +effectTheme and 0.667 for +effectAgent). Moreover, while the precision of the Null class in BiGraphConst4Rel is 0.583, the precision of Null class in Enhanced EffectWordNet is 0.697, which represents a significant improvement. In addition, the -effectTheme label presents higher recall value than the -effect label in BiGraphConst4Rel (recall of -effect in BiGraphConst4Rel is 0.612). Even though the -effectAgent label presents lower performance, the portion of the -effectAgent label is small as we mentioned in Figure 7 (i.e., 7.4% of -effect events). Since we show better performance in the others (+effectTheme, -effectTheme, +effectAgent, and Null) which account for a substantial portion, we can say overall performance is better than BiGraphConst4Rel. Through this experiment, we can know that considering the information about which entities are affected shows improvement.

		Enhanced EffectWordNet
+effectTheme	Precision	0.739
	Recall	0.667
	F-measure	**0.701**
-effectTheme	Precision	0.713
	Recall	0.726
	F-measure	**0.719**
+effectAgent	Precision	0.667
	Recall	0.828
	F-measure	**0.739**
-effectAgent	Precision	0.545
	Recall	0.571
	F-measure	0.558
Null	Precision	0.697
	Recall	0.690
	F-measure	**0.693**

Table 16: Results of Enhanced EffectWordNet.

		BiGraphConst4Rel
+effect	Precision	0.642
	Recall	0.680
	F-measure	0.660
-effect	Precision	0.779
	Recall	0.612
	F-measure	0.686
Null	Precision	0.583
	Recall	0.695
	F-measure	0.634

Table 17: Results of BiGraphConst4Rel in Chapter 6.

7.5 RELATED WORK

As we mentioned in Chapter 6, lexicons are widely used in sentiment analysis and opinion extraction. There are several previous works to acquire or expand sentiment lexicons such as [Kim and Hovy, 2004], [Strapparava and Valitutti, 2004], [Esuli and Sebastiani, 2006], [Gyamfi et al., 2009], [Mohammad and Turney, 2010] and [Peng and Park, 2011]. Such sentiment lexicons are helpful for detecting explicitly stated opinions, but are not sufficient for recognizing implicit opinions. Inferred opinions often have opposite polarities from the explicit sentiment expressions in the sentence; explicit sentiments must be combined with +/-effect event information to detect implicit sentiments.

There are a few previous works closest to ours. [Feng et al., 2011] build *connotation lexicons* that list words with connotative polarity and connotative predicates. [Goyal et al., 2010] generate a lexicon of *patient polarity verbs* that imparts positive or negative states on their patients. [Riloff et al., 2013] learn a lexicon of negative situation phrases from a corpus of tweets with hashtag "sarcasm".

Our work is complementary to theirs in that their acquisition methods are corpus-based, while we acquire knowledge from lexical resources. Further, all of their lexicons are word level while ours are sense level. Finally, the types of entries among the lexicons are related but not the same. Ours are specifically designed to support the automatic recognition of implicit sentiments in text that are expressed via implicature.

7.6 SUMMARY

In this chapter, we present a graph-based method for constructing a sense-level +/-effect lexicon with consideration of affected entities called Enhanced EffectWordNet. EffectWordNet built in Chapter 6 is a sense-level +/-effect lexicon without the information about which entities are affected. However, as we mentioned, the information about which entities are affected is important since the sentiment can be different in opinion inferences. Thus, we refine EffectWordNet with consideration of affected entities. In this chapter, we only consider the theme and the agent as the affected entity.

As we mentioned in Section 7.1, we had a small number of annotated data. Among 258 +effect synsets built in Chapter 6, 197 synsets (76.43%) are +effect on the theme and 52 synsets (20.15%) are +effect on the agent; there is one case in which there is +effect on both the agent and the theme. Also, among 487 -effect synsets, 433 synsets (88.89%) are -effect on the theme and 36 synsets (7.4%) are -effect on the agent; 9 synsets (1.85%) are -effect on both the agent and the theme. There are 16 instances which have positive or negative effects on both the agent and the theme with different polarities. Among them, half of each set is used as seed data in the graph model, and the other half is used for evaluation. In this work, we present that our method is promising even though the size of data is small.

We first define five labels such as +effectTheme, -effectTheme, +effectAgent, -effectAgent, and Null. Then, we assume that each synset can have one or two labels under no conflict (e.g., +effectTheme and -effectAgent). To handle these five labels and to allow one or two labels for each synset, we build four different graphs for each label except the

Null label using WordNet relations and then combine them according to rules in Section 7.3. Note that we already presented that building separate graphs and combining them with constraints is better than building one graph for three labels in Chapter 6.

In Section 7.4, we present that ENHANCED EFFECTWORDNET achieves good performance, which is generally better than BiGraphConst4Rel in Chapter 6. It represents that considering the information about which entities are affected is helpful to construct more refined sense-level +/-effect lexicon.

8.0 COARSE-GRAINED +/-EFFECT WORD SENSE DISAMBIGUATION

In Chapter 6 and Chapter 7, we developed a sense-level +/-effect lexicon due to significant sense ambiguity as we mentioned in Section 4.2.2. The sense of the word in context affects whether (or which) inference should be made. Consider the following example:

(4) *Oh no! The voters passed the bill.*

The meaning of *pass* in (4) is the following:

S_3: (v) legislate, pass (make laws, bills, etc. or bring into effect by legislation)

Under this sense, *pass* is, in fact, **+effect** for its theme. But, consider (6):

(6) *Oh no! They passed the bridge.*

In this context, the sense of *pass* is:

S_2: (v) travel by, pass by, surpass, go past, go by, pass (move past)

This type of passing event does not (in itself) positively or negatively affect the thing passed (*bridge*). That is, it is **Null**, not +effect nor -effect. This use of *pass* does not warrant the inference that the writer is negative toward the bridge. These examples illustrate that exploiting +/-effect event information for sentiment inference requires Word Sense Disambiguation (WSD).

In this chapter, we focus on +/-effect WSD, which is important for opinion inferences to extract implicit opinions. Thus, the goal of this chapter is to show that we can effectively identify the +/-effect events in a given text. Since our task is new, the architecture is different from typical WSD systems.

We address the following task: given +/-effect labels of *senses*, determine whether an instance of a word in the corpus is being used with a +effect, -effect, or Null sense. Consider a word W, where senses $\{S_1, S_3, S_7\}$ are -effect; $\{S_2\}$ is +effect; and $\{S_4, S_5, S_6\}$ are Null. For our purposes, we do not need to perform fine-grained WSD to pinpoint the exact sense; to recognize that an instance of W is -effect, for example, the system only needs to recognize that W is being used with *one* of senses $\{S_1, S_3, S_7\}$. Thus, we can perform **coarse-grained WSD**, which is often more tractable than fine-grained WSD.

Though supervised WSD is generally the most accurate method, we do not pursue a supervised approach, because the amount of available sense-tagged data is limited. Instead, we conduct a **knowledge-based WSD** method that exploits WordNet relations and glosses (described in Section 2.1). We use sense-tagged data (SensEval-3) only as gold-standard data for evaluation.

Our WSD method is based on *selectional preferences*, which are preferences of verbs to co-occur with certain types of arguments [Resnik, 1996, Rooth et al., 1999, Van de Cruys, 2014]. We hypothesize that preferences would be fruitful for our task, because +/-effect is a semantic property that involves affected entities. Consider the following WordNet information for *climb*:

climb:

S_1: (v) climb, climb up, mount, go up (go upward with gradual or continuous progress) "Did you ever climb up the hill behind your house?" **Null**

S_2: (v) wax, mount, climb, rise (go up or advance) "Sales were climbing after prices were lowered" **+effect**

S_3: (v) climb (slope upward) "The path climbed all the way to the top of the hill" **Null**

S_4: (v) rise, go up, climb (increase in value or to a higher point) "prices climbed steeply"; "the value of our house rose sharply last year" **+effect**

Senses S_1 & S_3 are both Null. We expect them to co-occur with *hill* and similar words such as *ridge* and *mountain*. And, we expect such words to be more likely to co-occur with S_1 & S_3 than with S_2 & S_4. Senses S_2 & S_4 are both +effect, since the affected entities are increased. We expect them to co-occur with *sales*, *prices*, and words similar to them. And, we expect such words to be more likely to co-occur with S_2 & S_4 than with S_1 & S_3. This example illustrates the motivation for using selectional preferences for +/-effect WSD.

We model sense-level selectional preferences using topic models, specifically Latent Dirichlet Allocation (LDA) [Blei et al., 2003]. We utilize LDA for modeling relations between sense groups and their arguments, and then carry out coarse-grained +/-effect WSD by comparing the topic distributions of a word instance and candidate sense groups and choosing the sense group that has the highest similarity value. Because selectional preferences are preferences toward arguments, the method must create a set of arguments to consider for each sense group. We exploit information in WordNet for automatically defining sets of arguments.

The system carries out WSD by matching word instances to sense groups. While the obvious way to group senses is simply by +/-effect label, the system does not need to group senses in this way. We experiment with a clustering process that allows more than one sense group with the same label for a given word. The motivation for allowing this is that there may be subsets of senses that have the same +/-effect label, but which are more similar to each other than they are to the other senses with the same +/-effect label. We also experiment with using mixtures of manually and automatically assigned sense labels in this clustering process, exploiting the results presented in Chapter 6 for automatically assigning +/-effect labels to verb synsets in WordNet.

In this chapter, we first explain the gold-standard data for evaluation in Section 8.1 and describe our evaluate metrics in Section 8.2. Then, our task is defined in Section 8.3. The detail method of creating the WSD system is described in Section 8.4, and the experiments and results are presented in Section 8.5. Finally, we discuss related work in Section 8.6 and describe our summary in Section 8.7. We are preparing to submit a journal for this work.

8.1 DATA

For evaluation, the SENSEVAL-3[1] English lexical sample task data is used. It provides training and test data for 57 words out of which 32 are verbs. Since we consider only verbs as +/-effect events in this work, we only utilize the verb data. We adopt the SENSEVAL-3 test data as our test data, which has a total of 1,978 instances for the 32 verbs.

To complete the gold standard, +/-effect labels are required. Although we provide our annotated data in Chapter 6, that data does not include the 32 verbs in the Senseval-3 data. Thus, we manually annotate the senses of all 32 verbs as +effect, -effect, or Null. The total number of synsets is 246. We follow the annotation scheme described in Section 4.1.1, which was found to lead to good inter-annotator agreement (0.84 percent agreement and 0.75 κ value reported in the previous study - Section 5.5). Our annotation rate was approximately 100 senses per hour. Note that sense labeling requires much less effort than creating sense-tagged training data, and can be viewed as a manual augmentation of WordNet, which was itself manually created. For future additional annotations, Section 6.6 give a method for guided manual annotation, where the model identifies unlabeled words that are likely to have +/-effect senses.

According to the manual annotations, among 246 synsets, 49 synsets (19.9%) are +effect, 36 synsets (14.6%) are -effect, and the rest (65.6%) are Null. Among 32 verbs, two verbs have +effect, -effect, and Null synsets, and 20 verbs have Null and one of +/-effect synsets. Thus, we see that 68.75% (22/32) of the verbs chosen for inclusion in SENSEVAL-3 require sense disambiguation to determine +/-effect labels for word instances.

Based on the sense labels, labels are assigned to the SENSEVAL-3 data to create the gold standard used as test data in all the experiments reported in this work. The test data consists of 467 +effect, 108 -effect, and 825 Null instances.

[1] SENSEVAL-3, http://www.senseval.org/

8.2 EVALUATION METRICS

In this chapter, we calculate the accuracy, precision, recall, and f-measure to evaluate our system such as Chapter 6. The accuracy is the degree of closeness of detected value to an actual or correct value. It is calculated as follows:

$$Accuracy = \frac{\text{Number of correctly detected synsets}}{\text{Number of all synsets in test data}} \tag{8.1}$$

As we mentioned, with the accuracy, we cannot evaluate the performance for each label. For example, if there is a predominant class, the base rate is close to the accuracy of predicting the predominant class. In this case, even though the performances for other labels that are not predominant labels are not good, the accuracy can be high. In our task, not only the accuracy but also the performance for each label is important. Thus, to evaluate for each label, we calculate precision, recall, and f-measure for all three labels.

The precision presents how many of detected instances are correct in each label. It is also called as positive predictive value. The precision for a given label is calculated as:

$$Precision_{label} = \frac{\text{Number of correctly detected synsets as a given label}}{\text{Number of all synsets detected as a given label}} \tag{8.2}$$

On the other hand, the recall indicates how many of relevant instances for each label is detected by the system. The recall is measured as follows:

$$Recall_{label} = \frac{\text{Number of correctly detected synsets as a given label}}{\text{Number of all synsets of a given label in test data}} \tag{8.3}$$

These two measures can be used together in the f-measure to provide a single measurement such as:

$$\text{F-measure}_{label} = 2 \cdot \frac{Precision_{label} \cdot Recall_{label}}{Precision_{label} + Recall_{label}} \tag{8.4}$$

8.3 TASK DEFINITION

The task addressed in this chapter is to recognize whether word instances in a corpus are used with +effect, -effect, or Null senses. Specifically, the gold standard consists of pairs $\langle w, l \rangle$, where w is an instance of word W in the corpus, and l is w's label, meaning that w is a use of W with a sense whose label is l. In this work, the gold standard is created by combining sense-tagged (Senseval) data and +/-effect sense labels as follows: $\langle w, l \rangle$ in our gold standard means that w has sense label W_s, and W_s has +/-effect label l.

For example, the label for the instance of *pass* in (4) is +effect, because the sense is S_3, and S_3 has the label +effect.

8.4 +/-EFFECT WORD SENSE DISAMBIGUATION SYSTEM

This section describes our method for building a selectional-preference +/-effect coarse-grained WSD system, given a resource such as WordNet and +/-effect labels on word senses.

In the first step, a coarse-grained sense inventory is constructed, by grouping senses (Section 8.4.1). The ultimate WSD system will assign each word instance in the corpus to one of the sense groups. For final evaluation, a word instance w that the WSD system has assigned to any sense group with label l is mapped to the pair $\langle w, l \rangle$, for comparison with the gold standard. The obvious grouping is simply by +/-effect labels: one group for the +effect senses, one for the -effect senses, and one for the Null senses. Alternatively, there may be multiple groups for a single label, where the senses in a group are more closely related to each other than they are to other senses with the same label. Our hypothesis for experimenting with variable grouping (i.e., allow more than one sense group with the same label for a given word) is that an effective method could be developed for creating a more fine-grained sense inventory customized to our task that would result in more accurate WSD performance.

Once the sense inventory is created, a model of selectional preferences for the sense groups is developed. Selectional preferences are preferences toward arguments. Thus, we have to identify a set of arguments for each group (Section 8.4.2). For example, suppose that S_2 and S_4 of *climb* are one sense group. The arguments for this group include nouns extracted from their glosses (*sales, prices*, etc.) together with others found by WordNet relation expansion. The final step in creating the WSD system is to model relations between sense groups and arguments to capture selectional preferences using LDA modeling (Section 8.4.3). This step defines argument class distributions, where the classes are hidden variables.

Finally, these distributions are exploited to perform WSD, as described in Section 8.4.4.

8.4.1 Sense Grouping

Performing coarse-grained WSD has the advantage that individual senses are aggregated, providing more information about each coarse-level sense.

For each word, senses can be simply grouped by label. However, a problem is that senses with the same +/-effect label but with very different selectional preferences are forced into the same group, making them indistinguishable to the WSD system. For instance, one sense of *carry* is *win in an election* and another is *keep up with financial support*. Though both are +effect, they have very different arguments. Nevertheless, they are forced into the same group. Because such groups contain several types of arguments, they can confuse the LDA models.

Thus, we adopt sense clustering to allow multiple groups with the same label, which can benefit the LDA models because each sense group can have purer arguments. The process is as follows:

1. Features are extracted from WordNet.
2. Senses are clustered based on the features.
3. Labels are assigned to clusters.

The features represent the absence or presence of the following words: words in the synset and the gloss for synset S_i; words in the synsets and the glosses for all S_i's hypernyms (i.e., more general word synsets); words in the synsets and the glosses of S_i's troponyms (i.e., more specific word synsets); words in the synsets and the glosses of S_i's verb groups (i.e., verb synsets with similar meanings).

For sense clustering, we adopt Expectation Maximization (EM) [Dempster et al., 1977] as implemented in the Weka library[2], which is modeled as a mixture of Gaussians. It follows an iterative approach to find the parameters of the probability distribution. In each iteration, the E-step (Expectation) estimates the probabilities of each data belong to each cluster, and the M-step (Maximization) estimates the parameter of the probability distribution of each cluster. In Weka, EM assigns a probability distribution to each instance, the probability of it belonging to each cluster. Further, EM selects the number of clusters automatically by maximizing the log-likelihood. It begins with one cluster and continues to add clusters until the estimated log-likelihood is decreased such as:

[2]Weka3, http://www.cs.waikato.ac.nz/ml/weka/

- Step1: The number of clusters is set to 1
- Step2: The data is split randomly into 10 folds, and EM is performed once for each fold.
- Step3: If the log likelihood, averaged over the 10 folds, increased, the number of clusters is increased by 1 and go to step 2. Otherwise, terminate.

After clustering, labels are assigned to clusters as follows. If all or a majority of senses in a cluster have the same label, then the cluster is assigned that label. If there is not a majority, then the cluster is labeled Null.

8.4.2 Arguments for Selectional Preferences

After grouping senses, arguments for each sense group must be extracted to exploit selectional preferences. Gloss information (definitions and examples) and semantic relations in WordNet are utilized.

We first combine gloss information of all senses in each sense group SG_k. Since glosses are not long, we consider all nouns in the combined glosses as arguments of the given sense group. We call this noun set N.

We also consider arguments gleaned from senses related to those in the sense group. While such arguments are less tightly coupled to the senses they are being extracted for, we hypothesize that, on balance, having a larger number of arguments may improve overall performance.

Let $commonSynset(word1, word2)$ be $True$ if there is at least one synset that contains a sense of $word1$ and a sense of $word2$. We add all words new for which $commonSynset(n, new)$ for some $n \in N$. The synset relation is the closest relationship between senses in WordNet, so we anticipate that adding these new arguments would be a conservative way to increase recall.

Going one step further, we consider WordNet verb relations for sense S_i in each sense group SG_k because we hypothesize that the super-subordinate relations can provide richer information. All nouns in glosses of hypernyms and troponyms of S_i are extracted and added to the argument set. In addition, the argument set contains all nouns in glosses of the senses that are in the same verb group with S_i. Generally speaking, the coverage of WordNet verb groups is not large, but the relations are reliable.

8.4.3 Topic Model

To model relations between sense groups and arguments for each +/-effect event, we adopt LDA, which is a generative model that discovers similarities in data using latent variables. It was introduced to model a set of documents in terms of topics, representing the underlying semantic structure of a document collection. In this work, sense groups play the role of documents, arguments play the role of terms, and argument classes play the role of topics in traditional usage of LDA. That is, rather than modeling relations between documents and terms, we model relations between sense groups and arguments. One advantage of LDA is argument classes need not be pre-defined, since LDA discovers these classes automatically. We adopt a variant of LDA suggested by [Griffiths and Steyvers, 2002, Griffiths and Steyvers, 2003, Griffiths and Steyvers, 2004].

Figure 8 shows the graphical model of our proposed topic model. Arrows represent conditional dependencies between variables. SG is a set of sense groups, N_{sg} is a set of arguments for each sense group sg, and C is a set of argument classes, which are hidden variables being discovered by the model.

Each sense group sg has a corresponding multinomial distribution Θ_{sg} over latent argument classes c. Distribution Θ_{sg} is defined from a Dirichlet distribution with prior parameter α. Each argument class c also has a corresponding multinomial distribution Φ_c over arguments n. Distribution Φ_c is defined from a Dirichlet with prior parameter β. To generate an argument n, a hidden argument class c is first chosen by Θ_{sg}, and then an argument n is chosen from Φ_c. The formal process is as follows:

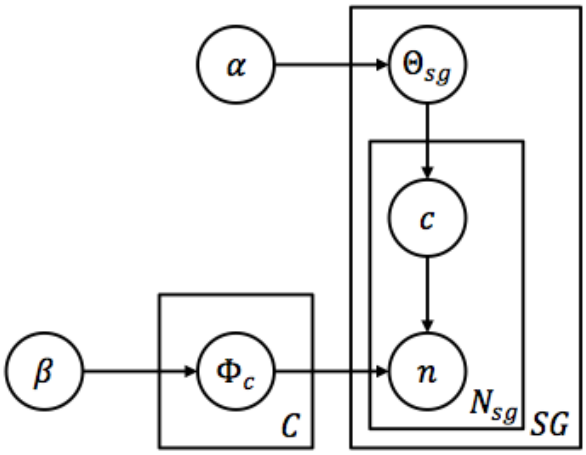


Figure 8: Plate notation representing our topic model.

1. Choose $\Theta_{sg} \sim Dir(\alpha)$, where $sg \in SG$ and $Dir(\alpha)$ is the Dirichlet distribution for parameter α.
2. Choose $\Phi_c \sim Dir(\beta)$, where $c \in C$.
3. To generate an argument,
   a. Draw a specific argument class $c \sim \Theta_{sg}$
   b. Draw an argument $n \sim \Phi_c$

In this model, the main variables are the argument distribution Φ for each argument class and the argument class distribution Θ for each sense group. They can be estimated directly, but this approach can get stuck in a local maximum of the likelihood function. Another method is to directly estimate the posterior distribution over argument class c [Steyvers and Griffiths, 2007]. For posterior inference, we use Gibbs sampling, which has been shown to be a successful inference method for LDA [Griffiths and Steyvers, 2004]. It sequentially samples variables from their distribution conditioned on the current values of all other variables. With these samples, we can approximate the posterior distribution.

For the implementation, we use the Mallet library[3] and use its default setting that assumes seven topics.

8.4.4 Word Sense Disambiguation

The topic model defines argument class distributions for each sense group. Let D_k be the argument class distribution of SG_k.

To disambiguate word instance W in the corpus, the nouns within a window size of five are extracted to serve as its arguments. We create a test instance with these nouns and obtain the argument class distribution of W by the topic model described above. Let this distribution be D_W.

We hypothesize that arguments can help determine the +/-effect polarity of senses for the given word. Each word can have several meanings, and the polarity can be different according to the meanings. We can distinguish these meanings based on their arguments. That is, our assumption is that if senses of W have similar types of arguments, they have the same +/-effect polarity. Thus, the system chooses the sense group that has the highest similarity value to D_W, since similar types of arguments can be expected to show similar argument class distributions. In particular, similarity is assessed as the cosine value between the distribution vectors D_W and D_k, for all D_k, and the k for which similarity is highest is selected. That is, if D_W has higher similarity value with D_3 than the others, SG_3 is selected. Finally, W is assigned the label of SG_k as its +/-effect label.

8.5 EXPERIMENTS

In this section, we first describe baselines for comparison in Section 8.5.1. We provide our experimental results in Section 8.5.2, and then we present the role of word sense clustering

[3]Mallet, http://mallet.cs.umass.edu/topics.php

in Section 8.5.3. Finally, we show the role of manual (vs. automatic) +/-effect sense labels in Section 8.5.4.

8.5.1 Baselines

As one baseline system, we adopt WordNet::SenseRelate::TargetWord,[4] which is an unsupervised WSD method that is freely available [Patwardhan et al., 2005]. In the table of results, this system is referred to as *BL1:SenseRelate*. Because it performs unsupervised WSD, it does not require sense-tagged training data. Since it is a WSD method, its output is a sense. Thus, after running it, we assign +/-effect labels based on the manually annotated senses described in Section 8.1. Among 1,978 instances in the test data, it does not provide any sense information for 691 instances (34.93%).

Another system is GWSD (*BL2:GWSD*), which is an unsupervised graph-based WSD system developed by [Sinha and Mihalcea, 2007][5]. Since its output is also a sense, we assign +/-effect labels based on the manually annotated senses similar to the strategy used for the previous baseline. When we run GWSD, we select the verbs as the target part of speech, Leacock & Chodorow (*lch*) as the similarity metric used to build the graph, six for the window size, and *indegree* for the graph centrality measure (indegree was found to have a performance comparable to other more sophisticated measures, and it is more efficient).

The other baseline system, called *Majority* Baseline simply chooses the majority class (Null).

8.5.2 Experimental Results

We evaluate our system for two verb sets: *All* consists of all 32 verbs and *Conf* contains the 22 verbs with +/-effect ambiguity.

[4]WordNet::SenseRelate, http://senserelate.sourceforge.net/
[5]GWSD, https://web.eecs.umich.edu/~mihalcea/downloads.html

Table 18 shows results for the *Majority* baseline, *BL1:SenseRelate*, *BL2:GWSD*, and our system. It gives accuracy, precision (P), recall (R), and f-measure (F) for all three labels.

While *BL1:SenseRelate* has the highest +effect precision and *Majority* baseline has the highest Null recall (as it assigns everything to the Null class), our system is substantially better on all other measures.

As we mentioned in Section 8.5.1, two baseline systems (except *Majority*) did not detect any sense information for many instances, so their recall is low. Nevertheless, they show high +effect and Null precision. In addition, in *BL2:GWSD*, Null precision is quite good.

However, our system outperforms them. It shows high recall scores for all three labels and the best accuracy score. Moreover, our system is better at detecting -effect events than all three baselines. In fact, the overall accuracy is 0.83 and all three f-measures are over 0.78, representing a good performance for a WSD approach that is not supervised.

Table 19 and Table 20 shows the role of argument types. As we explained in Section 8.4.2, we utilize gloss information and semantic relations in WordNet to extract arguments for selectional preferences. All cases of arguments are as follows:

- **ArgSet1**: All nouns (*Ns*) in gloss information of senses S in each sense group.
- **ArgSet2**: Case1 + synsets of *Ns*.
- **ArgSet3**: Case2 + all nouns in gloss information of hypernyms of S.
- **ArgSet4**: Case2 + all nouns in gloss information of troponyms of S.
- **ArgSet5**: Case2 + all nouns in gloss information of verb groups of S.

Table 19 presents the performance of each argument type and all of them. Based on our experiments, we get the best result with the combination of ArgSet1, ArgSet2, and ArgSet5. Table 20 shows the results of backward-ablation. We can know that each argument type is helpful to our task even though the difference is not big.

		Majority		BL1:SenseRelate		BL2:GWSD		Our Method	
		All	**Conf**	**All**	**Conf**	**All**	**Conf**	**All**	**Conf**
Accuracy		0.701	0.625	0.535	0.519	0.499	0.425	**0.880**	**0.833**
+effect	P	0.000	0.000	**0.807**	**0.814**	0.568	0.534	0.791	0.776
	R			0.449	0.469	0.368	0.344	**0.808**	**0.794**
	F			0.577	0.595	0.447	0.418	**0.799**	**0.785**
-effect	P	0.000	0.000	0.620	0.438	0.556	0.410	**0.943**	**0.921**
	R			0.220	0.130	0.425	0.313	**0.817**	**0.759**
	F			0.325	0.200	0.482	0.355	**0.875**	**0.832**
Null	P	0.701	0.625	0.804	0.773	0.834	0.736	**0.909**	**0.856**
	R	**1.000**	**1.000**	0.606	0.650	0.550	0.477	0.914	0.864
	F	0.824	0.769	0.691	0.706	0.663	0.579	**0.911**	**0.860**

Table 18: Experimental results for *All* and *Conf* set.

	+effect			-effect		
	P	R	F	P	R	F
ArgSet1	0.775	0.794	0.784	0.921	0.759	0.832
ArgSet2	0.773	0.791	0.782	0.921	0.759	0.832
ArgSet3	0.767	0.791	0.779	0.921	0.759	0.832
ArgSet4	0.726	0.804	0.763	0.921	0.759	0.832
ArgSet5	0.772	0.836	0.803	0.921	0.759	0.832
ArgAll(ArgSet1-5)	0.776	0.794	0.785	0.921	0.759	0.832
Best(ArgSet1,2,5)	0.778	0.838	0.807	0.921	0.759	0.832

	Null			Accuracy
	P	R	F	
ArgSet1	0.856	0.863	0.860	0.832
ArgSet2	0.855	0.862	0.858	0.831
ArgSet3	0.854	0.857	0.856	0.828
ArgSet4	0.855	0.822	0.838	0.811
ArgSet5	0.876	0.854	0.865	0.841
ArgAll(ArgSet1-5)	0.856	0.864	0.860	0.833
Best(ArgSet1,2,5)	0.877	0.858	0.868	0.844

Table 19: Performance of argument types on the *Conf* set.

	+effect			-effect		
	P	R	F	P	R	F
ArgAll(ArgSet1-5)	0.776	0.794	0.785	0.921	0.759	0.832
ArgAll - ArgSet1	0.766	0.796	0.781	0.921	0.759	0.832
ArgAll - ArgSet2	0.755	0.800	0.777	0.921	0.759	0.832
ArgAll - ArgSet3	0.770	0.814	0.791	0.921	0.759	0.832
ArgAll - ArgSet4	0.773	0.812	0.792	0.921	0.759	0.832
ArgAll - ArgSet5	0.768	0.810	0.788	0.921	0.759	0.832

	Null			Accuracy
	P	R	F	
ArgAll(ArgSet1-5)	0.856	0.864	0.860	0.833
ArgAll - ArgSet1	0.856	0.856	0.856	0.829
ArgAll - ArgSet2	0.857	0.847	0.852	0.825
ArgAll - ArgSet3	0.865	0.856	0.861	0.835
ArgAll - ArgSet4	0.864	0.858	0.861	0.836
ArgAll - ArgSet5	0.863	0.855	0.859	0.833

Table 20: The results of backward-ablation.

8.5.3 The Role of Word Sense Clustering

As described above, sense groups can be simply grouped by a label. That is, each word has one sense group for each label. In this case, each word can have at most 3 groups: +effect, -effect, and Null. We call such method the fixed sense grouping. Table 21 shows the result of the fixed sense grouping (Fixed) based on manually annotated senses described in Section 8.1. It also includes results for full fine-grained WSD (No Group). The same gold standard test set continues to be used for all experiments and only the *Conf* set is evaluated.

As expected, accuracy and all f-measures are the worst for fine-grained WSD, where no sense grouping is performed. Also, accuracy and all f-measures are substantially better than Fixed after automatically refining the system's sense inventory via clustering.

Following is an example illustrating how clustering can improve performance. Consider *suspend*, which has 5 -effect senses and 1 Null sense. Following are examples from SENSEVAL-3. The sense in Ex1-Ex2 is S_3, *bar temporarily*. The sense in Ex3-Ex4 is S_5, *make inoperative or stop*.

(Ex1) S_3 He was later suspended for two European games for unsporting behaviour.

(Ex2) S_3 He was suspended for two years after he tested positive for drugs when finishing second in the 1988 New York race.

(Ex3) S_5 France is to suspend nuclear tests at its South Pacific atoll site, Mururoa, this year, M Pierre Beregovoy, Prime Minister, said in his inaugural speech to parliament yesterday.

(Ex4) S_5 That was good enough to prompt Gordon Taylor, the PFA chief executive, to suspend the threat of industrial action.

S_3 and S_5 are both -effect, so fixed sense grouping forces them into the same group. But the contexts in which S_3 and S_5 are used are different, and the topic model must contend with one -effect group which includes quite varied contexts (sports related, politics related, etc.). In fact, the system incorrectly labels Ex3 as Null when the fixed sense groupings are used. With clustering, the system gets all of Ex1-Ex4 correct. A singleton cluster is correctly

		No Group	Fixed	Our Method
Accuracy		0.585	0.758	**0.833**
+effect	Precision	0.502	0.689	0.776
	Recall	0.699	0.743	0.794
	F-measure	0.584	0.715	**0.785**
-effect	Precision	0.500	0.638	0.921
	Recall	0.798	0.815	0.759
	F-measure	0.615	0.716	**0.832**
Null	Precision	0.713	0.824	0.856
	Recall	0.490	0.760	0.864
	F-measure	0.581	0.791	**0.860**

Table 21: Comparison among fine-grained WSD (No Groups), a fixed number of sense groups (Fixed), and a variable number of sense groups (Our Method) on *Conf* set.

	Precision	Recall	F-measure
+effect	0.776	0.794	0.785
-effect	0.921	0.759	0.832
Null	0.856	0.864	0.860
Overall	0.833	0.833	0.833

Table 22: Precision, Recall, and F-measure figures broken down per +/- effect.

created for the Null sense (*suspension in a fluid*). S_3 and S_5 are placed into separate groups with other senses. With these purer sense groups, the topic model is able to better model the selectional preferences and provide more accurate results.

8.5.4 The Role of Manual +/-Effect Sense Labels

Recall that the WSD system assigns the same label to all the senses in a cluster (the majority label, or Null if there isn't one). In Section 8.5.2 and Section 8.5.3, we used manually labeled sense data explained in Section 8.1. While sense labeling requires much less labor than sense tagging corpora, it is still desirable not to require full manual sense tagging. In this section, we also utilize EffectWordNet described in Chapter 6, which automatically labels all verb senses with +/-effect labels.

Figure 9 presents a learning curve with increasing percentages of (randomly selected) manual sense labels to determine cluster labels. We only show results for variable sense grouping because we carried out experiments on *Conf* set using 100% automatic labels comparing fixed versus variable sense grouping, and found that performance is much better with variable sense grouping.

On the left, 100% of the labels are automatic. Accuracy is 57.7% which is lower than the 84.4% accuracy reported in Table 18, when 100% of the manual labels are used. The f-measures are lower as well ($51 < 78.5$ for +effect; $80 < 83.2$ for -effect; and $62 < 86.0$ for

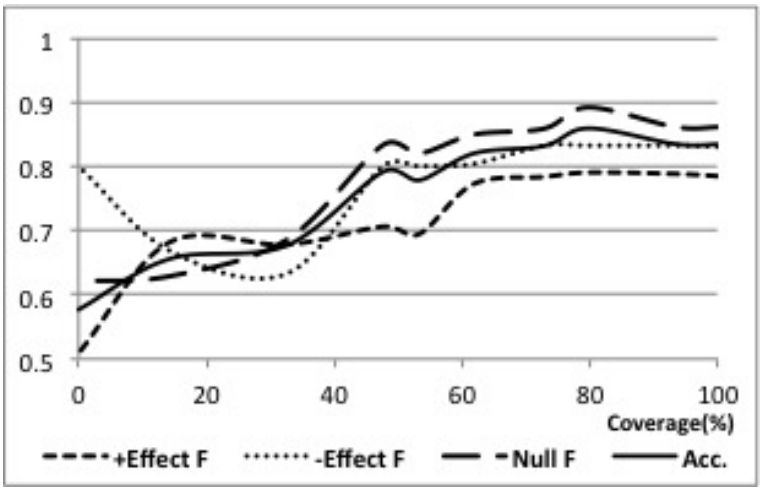


Figure 9: Learning curve on *Conf* with increasing percentages of manual sense annotations.

Null). Fortunately, with only 65% of manually annotated senses, we are close to maximum performance; with 80%, we reach the maximum performance. This suggests that, until all verbs have been manually labeled, good performance can still be obtained using some automatic labels to fill out coverage.

8.6 RELATED WORK

Several methods for WSD have been developed. We can distinguish WSD approaches into three parts: supervised WSD, unsupervised WSD, and knowledge-based WSD. The supervised WSD approaches use machine learning techniques, and in most tasks, they treat it as a classification problem [Mooney, 1996, Ng, 1997, Agirre and Martínez, 2000, Lee and Ng, 2002, Klein et al., 2002, Tsatsaronis et al., 2007]. The classifier is trained with training data in order to assign the appropriate sense to a given word. Generally, supervised WSD approaches show better performance than other methods as in [Navigli, 2009]. However, since they need labeled data as training data, it requires lots of human efforts for large coverage.

The unsupervised WSD approaches can overcome the lack of large-scale annotated data. This method is based on the idea that the same sense will have similar context. So, they automatically group words which have similar context and assign labels to each cluster. [Patwardhan et al., 2005] and [Pedersen and Kolhatkar, 2009] found the sense based on the semantic relatedness between a target word and its neighbors. [Agirre et al., 2006] built a co-occurrence graph and performed HyperLex [Véronis, 2004] and PageRank [Brin and Page, 1998] algorithm for WSD. [Klapafits and Manandhar, 2010] used a hierarchical structure in which vertices are the contexts of a word and edges represent the similarity between contexts and applied the Hierarchical Random Graphs for inferring it.

The knowledge-based approaches rely on the use of external lexical resources such as dictionaries and ontologies. These methods usually have lower performance than supervised methods, but they have a wider coverage. [Lesk, 1986] presented the gloss overlap approach. [Banerjee and Pedersen, 2003] introduced the extended gloss overlap, which expands the gloss information using WordNet relations. [Basile et al., 2014] proposed an enhanced Lesk method; the sense is selected by the distribution similarity between the gloss and context using BabelNet [Navigli and Ponzetto, 2012], which is large semantic network exploiting both WordNet and Wikipedia. Carroll and McCarthy [Carroll and McCarthy, 2000, McCarthy and Carroll, 2003] utilized selectional preference information. They showed that although the effect of selectional preferences is not huge, it can improve coverage and recall. [Mihalcea, 2004] built a graph that consists of all possible senses of words and connects two nodes when they have the same hypernym. They applied PageRank to this graph and selected the highest ranking sense. [Navigli and Lapata, 2010] also presented a study of graph-based WSD; nodes in a graph are senses, and edges are WordNet relations. [Chen et al., 2014] presented the unified model for joint word sense representation and disambiguation.

Recently, topic models are utilized for WSD. [Cai et al., 2007] used topic models for exploiting the global context. After developing topic models from a unlabeled large corpus, they combined it and other features and applied a supervised method. [Boyd-Graber et al., 2007] developed LDA with WordNet (LDAWN) where the sense of the word is a hidden variable. The multinomial topic distribution is replaced with a WordNet-Walk, which is a probabilistic process of word generation based on the hyponymy relationships. [Li et al.,

2010] presented a probabilistic model for WSD. They compared the topic distribution of a target with the sense candidates and chose the best one. We also use an LDA model, but to exploit knowledge-based selectional preferences for coarse-grained WSD.

8.7 SUMMARY

In this chapter, we investigate +/-effect WSD approach, which identifies the +/-effect of a word sense based on its surrounding context. Our goal is to show that we can effectively identify the +/-effect events in a given text, which is different from typical WSD systems.

Since our purpose is to determine whether an instance of a word in the corpus is being used with a +effect, -effect, or Null sense, we do not need to perform fine-grained WSD to pinpoint the exact sense. Thus, we perform coarse-grained WSD, which is often more tractable than fine-grained WSD. Moreover, because the amount of available sense-tagged data is limited, we conduct a knowledge-based WSD method, that exploits WordNet relations and glosses, rather than supervised WSD. That is, our method does not require any sense-tagged training data. We use SensEval-3 as gold-standard data for evaluation, as we mentioned in Section 8.1.

As we described in Section 8.4, the method we propose relies on selectional preferences. Selectional preferences are modeled using LDA. We use automatic clustering based on the preference arguments, which is extracted from WordNet information, to create a sense inventory customized to our task.

Through several experiments in Section 8.5, we show that our method achieves very good performance, with an overall accuracy of 0.83, which represents a significant improvement over three competitive baselines. In +effect label, even though the precision in one baseline (*BL1:SenseRelate*) is higher than our method, we show a significant improvement in the recall. In -effect label, our method outperforms all baseline systems with all measures. With a majority baseline, since the majority is Null, the recall of Null is 1.0. Although our system has lower recall, we show better precision and f-measure. Also, we show that each argument

type as selectional preferences is helpful to our task.

Moreover, we present the role of word sense clustering in Section 8.5.3. In our experiments, the variable sense grouping (i.e., allow more than one sense group with the same label) outperforms the fixed sense grouping (i.e., one sense group for each label) and fine-grained WSD (i.e., no grouping). Since it can have purer sense groups with the variable sense grouping, the topic model is able to better model the selectional preferences and provide more accurate results.

In addition, in Section 8.5.4, we show that good performance can still be obtained using some automatic labels to fill out coverage.

9.0 JOINT EXTRACTION OF +/-EFFECT EVENTS AND AFFECTED ENTITY

The ultimate goal of the opinion inference system is to develop a fully automatic system capable of recognizing inferred attitudes such as +/-effect events and affected entities. This information is necessary for not only opinion inferences but also connotation frames introduced by [Rashkin et al., 2016].

Consider the sentence (3):

(3) *The bill would curb skyrocketing health care costs.*

We know that the writer expresses an explicit **negative sentiment** toward *health care costs* because of *skyrocketing*. Since the event, *curb*, is a -effect on the theme, *skyrocketing health care costs*, we can infer that the writer has **positive sentiment** toward the event because it has a negative effect on the theme toward which the writer is negative. We can also infer that the writer has **positive sentiment** toward *the bill* which is the agent of the event. In this example, for opinion inferences, we have to know that *curb* is a -effect event and the affected entity is *skyrocketing health care costs*. (Of course, we have to know that *skyrocketing* is a negative term. Since there are several explicit sentiment analysis system such as OpinionFinder [Akkaya et al., 2011] and Sentiment Treebank [Socher et al., 2013], we can easily get this information.)

Thus, in this chapter, we present a pilot study to extract both +/-effect events and their affected entities. That is, there are two tasks such as the +/-effect event detection, which consists of two sub-tasks such as recognizing the span of +/-effect events and detecting the polarity of these events, and the affected entity identification.

For the +/-effect event detection, we already present that +/-effect events have substantial sense ambiguity (i.e. some words have mixtures of +effect, -effect, and Null) in Section 4.2.2 and created the sense-level +/-effect event lexicon, called EffectWordNet, in Chapter 6 and Chapter 7. Also, we investigated WSD for +/-effect events to utilize such sense-level +/-effect lexicons in Chapter 8. Based on this information, we have to extract +/-effect events in a given sentence, that is, first recognize a span of +/-effect events and then detect the polarity of +/-effect events.

For the affected entity identification, as we mentioned in Section 4.2.4, the information about which entities are affected is important since the sentiment toward an entity can be different. While the affected entity is the theme in many cases, it is the agent or other entity sometimes.

In the sentence (3), the given event, *curb*, is -effect on the theme (i.e., the affected entity is the theme), and the writer's sentiment toward the theme is negative. Thus, we know that the writer has a positive sentiment toward the event, and the sentiment toward the agent is positive.

However, consider the following example:

(7) *Yay! John's team lost the first game.*

We know that the writer expresses an explicit positive sentiment toward the event because of *Yay!*. The event, *lost*, has a **negative effect** on the agent, *John's team*, since he fails to win. That is, in this example, the affected entity is the agent, not the theme. We can infer that the writer has **negative sentiment** toward the agent because the event, toward which the writer is positive, has a negative effect on the agent.

Compared to the sentence (3), even though both are -effect events and the writer has a positive sentiment toward these events, the sentiment toward the agent is different according to what the affected entity is. As illustrated by these examples, it is important to know which entities are affected by the event in opinion inferences. Thus, we have to identify which entities are affected by the given events. At that time, we have to cover all cases of affected entities, not only themes but also agents or other entities.

These two tasks (i.e. the +/-effect event detection and the affected entity identification) might be regarded as independent tasks, so they can be placed in a pipeline system such as firstly extracting +/-effect events and then identifying their affected entities. [Deng et al., 2014, Deng and Wiebe, 2015] include such approach. In [Deng et al., 2014], they simply check the presence of +/-effect words in a word-level lexicon (not a sense-level lexicon) for the +/-effect event detection, and they adopt the semantic role labeler and generate simple rules to identify affected entities. In [Deng and Wiebe, 2015], they utilize EffectWordNet to recognize the polarity of +/-effect events, and they also adopt the semantic role labeler to identify affected entities. However, in these works, the span of +/-effect events is given. As we mentioned, for the ultimate goal, we have to recognize the span of +/-effect events automatically. In this dissertation, we present not only detecting the polarity of +/-effect events but also recognizing the span of +/-effect events. That is, while the inputs of their system are a sentence and a span of +/-effect events, the input of our system is only a sentence. In addition, while they consider only theme as an affected entity, we cover all cases of affected entities (i.e., not only a theme but also an agent or other entities). As we mentioned, depending on +/-effect events and contexts, an affected entity can be different (e.g., while the affected entity is the theme in the sentence (3), it is the agent in the sentence (7)). Recall that the information about which entities are affected is important because the sentiment toward an entity can differ.

We hypothesize that there are inter-dependencies between +/-effect events and their affected entities. As we mentioned in Section 4.2.2, since words can have a mixture of +effect, -effect and Null, it is important to grasp the meaning of the given word. So, contexts, especially affected entities, are important information to detect +/-effect events. For example, in the sentence (7), because the affected entity is *John's team*, we can know the meaning of *lost* is *to fail to win* which is a -effect event. On the other hand, to identify the affected entity, +/-effect event information is also important. For instance, in the sentence (3), the affected entity is *health care costs*, which is the theme of the event, *curb*. However, in the sentence (7), since the event is *lost*, the affected entity is *John's team*, which is the agent, not *the first game*, which is the theme. Thus, detecting +/effect events and the affected entities can be mutually beneficial.

Therefore, we propose a joint model to extract both +/-effect events and their affected entities. We adopt the structured perceptron suggested by [Collins, 2002] for a joint model. Structured perceptron is a machine learning algorithm for structured prediction problems. Since our input (i.e., a sentence) has structures and our output (i.e., +/-effect events and their affected entities) also has structures such as sequences and trees, we hypothesize that the approach for the structured prediction is appropriate for our task.

First, we explain our data for training and test in Section 9.1 and describe our evaluate metrics in Section 9.2. Then, our task is defined in Section 9.3. Our joint model is explained in Section 9.4 and features are described in Section 9.5. The experimental results are presented in Section 9.6, and related work is discussed in Section 9.7. Finally, our summary is given in Section 9.8.

9.1 DATA

As we mentioned in Section 4.1.1, Deng et al. [Deng et al., 2013] presented an annotation scheme for +/-effect events and for the sentiment of the writer toward their agents and objects. Each event is representable as a triple of text spans, ⟨*agent, +/-effect event, object*⟩. The agent should be a noun phrase or *implicit* when the given text doesn't have the agent information explicitly. The object also should a noun phrase. Another component is the influencer, a word whose effect is to either retain or reverse the polarity of +/-effect event. However, since our task is to extract +/-effect events and their affected entities, we ignore the annotations related to influencers.

Based on this annotation scheme, +/-effect corpus is created. This corpus is based on the arguing corpus [Conrad et al., 2012], which consists of 134 documents from blogs and editorials about a controversial topic, *the Affordable Care Act*. We already present the reliability of the annotation scheme through the agreement study in Section 4.1.1.

We adopt +/-effect corpus as the training and test data. Based on the annotation scheme, objects are affected entities of +/-effect events. In this corpus, 1,372 +/-effect events and their arguments are annotated. Among them, 592 events are +effect and 780 events are -effect.

9.2 EVALUATION METRICS

In this chapter, we calculate the accuracy, precision, recall, and f-measure to evaluate our system such as Chapter 6. The accuracy is the degree of closeness of detected value to an actual or correct value. It is calculated as follows:

$$Accuracy = \frac{\text{Number of correctly detected synsets}}{\text{Number of all synsets in test data}} \tag{9.1}$$

However, with the accuracy, we cannot evaluate the performance for each label. For example, if there is a predominant class, the base rate is close to the accuracy of predicting the predominant class. In this case, even though the performances for other labels that are not predominant labels are not good, the accuracy can be high. In our task, not only the accuracy but also the performance for each label is important. Thus, to evaluate for each label, we calculate precision, recall, and f-measure for all three labels.

The precision presents how many of detected instances are correct in each label. It is also called as positive predictive value. The precision for a given label is calculated as:

$$Precision_{label} = \frac{\text{Number of correctly detected synsets as a given label}}{\text{Number of all synsets detected as a given label}} \tag{9.2}$$

On the other hand, the recall indicates how many of relevant instances for each label is detected by the system. The recall is measured as follows:

$$Recall_{label} = \frac{\text{Number of correctly detected synsets as a given label}}{\text{Number of all synsets of a given label in test data}} \tag{9.3}$$

These two measures can be used together in the f-measure to provide a single measurement such as:

$$\text{F-measure}_{label} = 2 \cdot \frac{Precision_{label} \cdot Recall_{label}}{Precision_{label} + Recall_{label}} \tag{9.4}$$

9.3 TASK DEFINITION

The task in this work is to recognize +/-effect events and their affected entities. For that, there are two tasks such as:

- **+/-Effect Event Detection**: To recognize the span of +/-effect event instances and to detect whether the detected event instances are used with +effect, -effect, or Null (i.e., neither).
- **Affected Entity Identification**: To identify affected entities of detected +/-effect events.

Thus, given a sentence, our system should detect the span of +/-effect events with a polarity (i.e., +/-effect) and identify their affected entities. That it, if the input is the sentence (3), our system extracts *curb* as a -effect event and *skyrocketing health care costs* as its affected entity.

Specifically, the gold standard consists of triples of the form $\langle e, p, a \rangle$, where e is a +/-effect event term, p is a polarity of the given event term, and a is an affected entity of the given event. For instance, in the case of the sentence (3), *The bill would curb skyrocketing health care costs.*, the gold standard contains ⟨curb, -effect, skyrocketing health care costs⟩. As we mentioned in Section 9.1, we adopt +/-effect corpus as the training and test data.

9.4 JOINT EXTRACTION USING STRUCTURED PERCEPTRON

We hypothesize that there are inter-dependencies between +/-effect events and their affected entities. Thus, we suggest a joint extraction of +/-effect events and their affected entities that co-occur in the same sentence.

In this work, we handle this problem as a structured prediction problem. Structured prediction is similar with multiple classifications. While the output of the classification is a single label, the output of structured prediction is a set of inter-related labels or structures. Since outputs of our task (i.e., +/-effect events and their affected entities) are inter-related labels, we assume that the structured prediction problem is appropriate for our task. Thus, we adopt structured perceptron to jointly extract +/-effect events and their affected entities.

9.4.1 Representation

Let $x = \langle E, A \rangle$ denote the sentence instance, where $E = \{v_1, v_2, ..., v_m\}$ is the set of +/-effect event candidates in the sentence and $A = \{a_1, a_2, ..., a_n\}$ is the set of affected entity candidates. In this work, we assume that +/-effect events are verbs and affected entities are noun phrases (NPs). Thus, all verbs are regarded as candidates of +/-effect events, and all NPs in a sentence are candidates of affected entities. To denote the output structure, we define such as:

	+/-Effect Label	Entity Candidates		
		$\mathbf{a}_1$	...	$\mathbf{a}_n$
$\mathbf{v}_1$	e_1	$r_{1,1}$	...	$r_{1,n}$
...	...	...	...	...
$\mathbf{v}_m$	e_m	$r_{m,1}$	...	$r_{m,n}$

Table 23: The structure of inputs and outputs in our system. v_i and a_j are inputs and e_i and $r_{i,j}$ are outputs.

$$y = (e_1, r_{1,1}, .., r_{1,n}, ..., e_m, r_{m,1}, ..., r_{m,n}) \tag{9.5}$$

where e_i represents the +/-effect polarity (i.e., +effect, -effect, or Null) for the event candidate v_i, and $r_{i,j}$ represents the affected entity label of the affected entity candidate a_j when the given event is v_i. There are two possible affected entity labels: AffectedEntity and None which means that the given candidate is not an affected entity. To better understand, Table 23 shows the structure of inputs and outputs in our system.

For example, in the sentence, *Improving care for seniors after they leave the hospital*, there are two verbs (i.e., *improving* and *leave*) and three NPs (i.e., *care for seniors*, *they*, and *the hospital*). According to our assumption, all verbs are event candidates, and all NPs are affected entity candidates. Among them, *improving* is a +effect event and its affected entity is *care for seniors*. In this case, *leave* is Null since the meaning of *leave* in this sentence is *going away from a place*. This type of event does not positively or negatively affects the entity. Table 24 shows the representation of this example.

	+/-Effect Label	Entity Candidates		
		care for seniors	**they**	**the hospital**
improving	+Effect	AffectedEntity	None	None
leave	Null	None	None	None

Table 24: The representation of the sentence, *Improving care for seniors after they leave the hospital.*

9.4.2 Structured Perceptron with Beam Search

Structured perceptron is an extension of linear perceptron to handle structured predictions problems [Collins, 2002]. Given input $x \in X$, the prediction function of structured perceptron is such as:

$$\hat{y} = \arg\max_{y \in Y(x)} w \cdot \Phi(x, y) \tag{9.6}$$

where $Y(x)$ denotes the label set, w is a parameter vector (weight vector), and $\Phi(x, y)$ represents the feature vector for an instance x along with y. $w \cdot \Phi(x, y)$ is the inner product as follows:

$$\hat{y} = \arg\max_{y \in Y(x)} \sum_{s} w_s \Phi_s(x, y) \tag{9.7}$$

The learning task is to set the parameter vector w using the training data. The decoding algorithm is a method for searching for the arg max in Equation 9.6.

The perceptron learns the parameter vector w by online learning. The structured perceptron iterates over training instances $\{x, y\}$. In each iteration, with the current parameter vector w, the algorithm finds a prediction $\hat{y}$ given the input x. If $\hat{y}$ is incorrect, the parameter vector is updated as follows:

$$w = w + \Phi(x, y) - \Phi(x, \hat{y}) \tag{9.8}$$

In the standard perceptron, since it returns the most recent parameter vector, it might be overfitting to the last few instances. To reduce this overfitting, we adopt averaged perceptron that returns the average of all parameter vectors [Collins, 2002].

The important part in the structured perceptron is the decoding procedure, which searches for the structure with the maximal score in structured inference. There are two different categories for the decoding: exact search and inexact search. With the simple task, exact search can be performed well. However, since we have to joint model for +/-effect events and their affected entities, exact search is intractable. Thus, we adopt inexact search suggested by [Collins and Roark, 2004]. They proposed the incremental perceptron, which is a variant on the structured perceptron. Their idea is to replace the arg max with a beam search algorithm to find the maximal score under the parameter model.

During beam search, if the partial output y' ranks too low and falls out from the beam, there is no possibility of output y being in the final set. To handle it, [Collins and Roark, 2004] proposed an early update strategy, and [Huang et al., 2012] proved its convergence. In each step of the beam search, when the prefix of an output falls out of the beam, the top result in the beam is returned for an early update.

Algorithm 1 describes the learning algorithm for averaged structured perceptron with beam search and early update. $y_{[1:i]}$ denotes the prefix of y with length i.

Algorithm 1 Learning algorithm for averaged structured perceptron with beam search and early update.

Input: Training data D, Number of iterations T

Output: Parameter vector w

Initialization: Set $w = 0$, $w_c = 0$, $c = 1$
for $t \in 1...T$ **do**
  for $(x, y) \in D$ **do**
    $\hat{y} \leftarrow$ beamSearch(x,y,w)
    if $\hat{y} \neq y$ **then**
      $w \leftarrow w + \Phi(x, y_{[1:i]}) - \Phi(x, \hat{y})$
      $w_c \leftarrow w_c + c\Phi(x, y_{[1:i]}) - c\Phi(x, \hat{y})$
    end if
    $c \leftarrow c + 1$
  end for
end for
return $w - w_c/c$

9.4.3 Beam Search Decoder

As we mentioned, we have two sub-tasks: +/-effect event detection and affected entity identification. Thus, to jointly detect +/-effect events and their affected entities, we adopt the decoding algorithm with multiple beam search suggested by [Li et al., 2013].

They suggested the beam search decoding algorithm for joint event extraction that predicts the event triggers and arguments simultaneously. Their method is for ACE event extraction task, so there are 33 event subtypes. However, in our case, we only have three labels, +effect, -effect, and Null. Even though each event subtype may have distinguishing features, our event labels are more broad concepts. Also, since the task is different, we need the different features for +/-effect events. Features are explained in Section 9.5. Moreover, their system has constraints about arguments according to the ACE annotation guideline. For instance, the *Attacker* argument can only be one of *PER*, *ORG*, and *GPE*. However, we don't have any constraints for affected entities. We consider all noun phrases as candidates.

Algorithm 2 shows the decoding algorithm with multiple beam search for the joint +/-effect event and affected entity extraction. There are two sub-steps:

- **+/-Effect Event Labeling**: We consider all possible +/-effect labels for the given event candidate v_i. In the algorithm, $Append(b, l)$ means that label l is appended to the end of b. That is, each label is appended to existing partial assignments in one of the previous beams, and new assignment is generated. These assignments are saved in buf_v. Then, the top k results are selected to the beam B. To calculate a score, we use the linear model defined in Equation 9.6.

- **Affected Entity Labeling**: In this step, we traverse all results in the beam B. If we find the instance that +/-effect label is not Null (i.e., the label is +effect or -effect), the algorithm labels each argument candidate and creates new assignments. These are saved in buf_a. After scoring, we select the k best results to the beam B.

Algorithm 2 Beam search decoding algorithm for a joint +/-effect event and affected entity extraction.

Input: Instance $x = \langle E, A \rangle$, Beam size K,

+/-Effect label set L_e, Affected entity label set L_a

Output: Best $\hat{y}$ for x

Initialization: Set empty beam B
for $v_i \in E$ **do**
  $buf_v \leftarrow \{Append(b, l) | b \in B, l \in L_e\}$
  $B \leftarrow Best_k(buf_v)$
  for $a_j \in A$ **do**
    $buf_a \leftarrow \emptyset$
    for $b \in B$ **do**
      if $b_{v_i} \neq$ Null **then**
        $buf_a \leftarrow buf_a \cup \{Append(b, l) | l \in L_a\}$
      end if
    end for
    $B \leftarrow Best_k(buf_a)$
  end for
end for
return $B[0]$

9.5 FEATURES

In this section, we describe several features used in our system. There are three feature types: basic features, features for EffectWordNet, and features for relations between +/-effect events and affected entities. The basic features in Section 9.5.1 indicate lexical and syntactic features for both +/-effect events and affected entities. The features for EffectWordNet described in Section 9.5.2 are for only +/-effect event detection. Since the concept of +/-effect events are too broad, it is difficult to detect +/-effect events with the small set of training data. Thus, we utilize EffectWordNet. Section 9.5.3 is to represent lexical and syntactic relations between +/-effect events and their affected entities. As we mentioned, we hypothesize there are dependencies between them, so they can help each other. Thus, we present several features for relations between them.

9.5.1 Basic Features

The basic features are related on both +/-effect events and affected entities. There are seven basic features:

- Unigram of the current word.
- Lemma of the current word.
- Part-Of-Speech (POS) of the current word.
- Synonyms of the current word.
- Context words of the current word within 3 windows size.
- Dependent and governor words of the current word.
- Dependency types of the current word.

The following is one example of unigram feature function for +/-effect event:

$$f_{b1}(v_i, y_{e_i}) = \begin{cases} 1 & \text{if } v_i = \text{pass and } y_{e_i} = \text{+Effect} \\ 0 & \text{otherwise} \end{cases}$$

In this feature, if the text of v_i is *pass* and the label of y_{e_i} is +Effect, it is triggered. To create the dependency parse and to get lemma and POS, we use the Stanford coreNLP [Manning et al., 2014]. For synonyms, we utilize WordNet.

9.5.2 Features for EffectWordNet

For this work, we utilize EffectWordNet described in Chapter 6, not Enhanced EffectWordNet described in Chapter 7 since the joint extraction system automatically extracts the affected entity. While Enhanced EffectWordNet considers only the case that the affected entity is a theme or an agent, EffectWordNet can cover all cases (including the case that the affected entity is an entity which is neither a theme nor an agent).

We utilize EffectWordNet in various ways. Let S be the set of all senses of the current word. The score for each +/-effect label l is calculated such as:

$$Score_l(S) = \sum_{s_k \in S} EffectWN(s_k, l) \tag{9.9}$$

where

$$EffectWN(s_k, l) = \begin{cases} 1 & \text{if the lable of} s_k \text{ is } l \\ 0 & \text{otherwise} \end{cases}$$

There are seven types of features for +/-effect events.

- $Score_l(S)$ for each +/-effect label.
- Label that has the maximum score in Equation (9.5).
- Label of the detected sense by the word sense disambiguation system.
- Synsets of S.
- Whether the current word is monosemous (i.e., the current word has only one sense).
- Whether the current word has mixed +/-effect polarity.
- Whether the current word has Null sense.

9.5.3 Features for Relations between +/-Effect Events and Affected Entities

There are nine types of lexical or syntactic features for relations between +/-effect events and affected entity.

- Lexical distance between +/-effect event and affected entity.
- Dependency path between +/-effect event and affected entity.
- Length of the path between +/-effect event and affected entity in dependency tree.
- Common root node of +/-effect event and affected entity.
- Whether it is the nearest affected entity from +/-effect event among candidates.
- Whether +/-effect event and affected entity are in the same clause.
- Whether affected entity is the subject of the +/-effect event.
- Whether affected entity is the object of the +/-effect event.
- Semantic role label of the affected entity of the +/-effect event.

The following is one simple example feature for relations between a +/-effect event and an affected entity:

$$f_{r1}(v_i, a_j, y_{e_i}, y_{r_{i,j}}) = \begin{cases} 1 & \text{if } SRole(v_i, a_j) = \text{A1}, \\ & y_{e_i} = \text{+Effect, and} \\ & y_{r_{i,j}} = \text{AffectedEntity} \\ 0 & \text{otherwise} \end{cases}$$

In this feature, if the label of y_{e_i} is +Effect, the label of $y_{r_{i,j}}$ is AffectedEntity, and $SRole(v_i, a_j)$, which indicates the semantic role of a_j when the predicate is v_i, is A1, it is triggered.

For the semantic role labeler, we adopt SENNA[1] semantic role labeling system [Collobert et al., 2011].

9.6 EXPERIMENTS

In this section, we first describe baselines for comparison (Section 9.6.1). Then, we provide our experimental results (Section 9.6.2).

9.6.1 Baseline System

Since our task is the first work for jointly recognizing the span of +/-effect events, detecting the polarity of them, and identifying their affected entities, there is no existing system for a comparison. Since our task is different from the one addressed by [Deng et al., 2014] (recall

[1]SENNA, http://ronan.collobert.com/senna/

the span of +/-effect events is given in their system), we cannot compare our system with their system. Also, since [Deng and Wiebe, 2015] performed entity-level sentiment analysis and utilized +/-effect information to recognize implicit sentiments, they didn't provide any results related our task. Thus, as the baseline system, we adopt a pipeline method with two different systems for each sub-task: +/-effect event detection and affected entity identification.

As we mentioned, we created EffectWordNet, which is the sense-level +/-effect lexicon. For +/-effect event detection, we utilize EffectWordNet. However, since it is sense-level, we need Word Sense Disambiguation (WSD) to pinpoint the sense. Thus, for +/-effect event detection, we first conduct WSD and assign +/-effect labels based on EffectWordNet. For WSD system, we utilize WordNet::SenseRelate [Patwardhan et al., 2005] which is an unsupervised approach. Since word senses are not annotated in the data we use, we don't have the training data for WSD. Thus, we adopt an unsupervised model.

For affected entity identification, we adopt SENNA semantic role labeling system [Collobert et al., 2011]. Deng et al. [Deng et al., 2014] used the output of SENNA to extract agent and theme candidates. We utilize their rules to detect the affected entity. SENNA has two different labels related to the affected entity: A1 (object), and A2 (indirect object). We consider A1 of the +/-effect event as the affected entity. If A1 is not labeled but A2 is labeled, we consider A2 of the +/-effect event as the affected entity.

9.6.2 Experimental Results

For experiments, we set the number of iteration T to 50 and the beam size K to 5. We conduct 10-fold cross validation with the given data.

Table 25 shows the results of +/-effect event detection of the baseline system and our system. It gives accuracy, precision, recall, and f-measure for all three labels.

Our system shows higher accuracy and f-score than the baseline system. In general, the performance is low. One of reasons is that because EffectWordNet is created automatically, it can provide wrong information. Moreover, although our system assumes +/-effect

events are verbs, a few annotated +/-effect events in the corpus are not verbs. For instance, in a sentence, *It is a moral obligation to end this indefensible neglect of hard-working Americans*, *neglect* is annotated as a -effect event, but it is not a verb. Since our system considers only verbs as +/-effect events, these cases cannot be detected by our system.

In addition, as we mentioned, we consider that all verbs are candidates of +/-effect events, so we regard verbs that are not annotated in the corpus as Null. However, among verbs that are not annotated in the corpus, some +/-effect events are missed by human annotators. Thus, in our experiments, the recall of Null is too low and the precision of +/-effect is also low (i.e., we consider verbs as Null since they are not annotated, but they are detected as +effect or -effect in the system). If we consider only +/-effect (i.e., ignoring Null cases), our system achieves the accuracy of 0.563 while the baseline system shows 0.480 accuracy value.

		Baseline	**Our System**
Accuracy		0.109	**0.391**
+Effect	Precision	0.130	0.354
	Recall	0.334	0.250
	F-measure	0.187	**0.293**
-Effect	Precision	0.373	0.439
	Recall	0.403	0.671
	F-measure	0.387	**0.531**
Null	Precision	1.000	0.207
	Recall	0.007	0.102
	F-measure	0.013	**0.137**

Table 25: Results of +/-Effect Event Detection.

Table 26 shows the results of affected entity identification. It also gives accuracy, precision, recall and f-measure for all two labels. Since the baseline system provides one affected entity according to rules, we only calculate accuracy. Our system outperforms the baseline system. Since all candidates that are not annotated as affected entities are considered as None, the number of None labels is larger than the number of AffectedEntity labels. Thus, the precision of None is high while the precision of AffectedEntity is low. Because our task is not a easy task, the overall performance is low. However, our evaluations show that a joint model is promising for extracting +/-effect events and their affected entities.

		Baseline	**Our System**
Accuracy		0.242	**0.427**
AffectedEntity	Precision		0.198
	Recall		0.684
	F-measure		0.307
None	Precision		0.836
	Recall		0.369
	F-measure		0.511

Table 26: Results of Affected Entity Identification.

9.7 RELATED WORK

While our dissertation is the first research for +/-effect event detection and affected entity identification, there are several works for event extraction.

The event extraction task is to extract events and their arguments. In NLP, the event extraction task has received significant attention. Most research about event extraction has been conducted with the Automatic Content Extraction (ACE) data [Doddington et al.,

2004]. In early work, researchers present a pipeline system that first extracts event triggers and then identifies their arguments [Grishman et al., 2005, Ahn, 2006, Ji and Grishman, 2008, Liao and Grishman, 2010, Li et al., 2012].

Many researchers also focus on biomedical event extraction that extracts information of molecular events from text. BioNLP shared tasks [Kim et al., 2009] is one of evaluation tasks for biomedical event extraction. Such as ACE data, most works are adopted a pipeline approach [Bui and Sloot, 2011, Le Minh et al., 2011].

There are several works for a joint model to extract event triggers and their arguments simultaneously. [Poon and Vanderwende, 2010] uses Markov Logic for a joint inference from biomedical data, and McClosky et al. [McClosky et al., 2011] utilizes the dependency parsing for relations between a biomedical event and an argument. [Li et al., 2013] adopts a structured perceptron to extract ACE event triggers and their argument jointly. [Araki and Mitamura, 2015] also presents a structured perceptron for a joint event trigger identification and event coreference resolution.

As we mentioned, our task is new. Even though there are several event detection systems for ACE data and BioNLP shared tasks, these are incompatible with our task. In ACE data and BioNLP shared tasks, each event has a specific definition, but the concept of +/-effect is broader. In addition, we consider all noun phrases as affected entity candidates, not specific types.

9.8 SUMMARY

In this chapter, we present a pilot study to jointly extract +/-effect events and their affected entities.

The ultimate goal of the opinion inference system is to develop a fully automatic system capable of recognizing inferred attitudes such as +/-effect events and affected entities. Also, this information is necessary for not only opinion inferences but also connotation frames introduced by [Rashkin et al., 2016].

As we mentioned in Section 9.3, there are two tasks such as the +/-effect event detection, which consists of two sub-tasks such as recognizing the span of +/-effect events and detecting the polarity of these events, and the affected entity identification.

These two tasks might be regarded as independent tasks, so they can be placed in a pipeline system such as firstly extracting +/-effect events and then identifying their affected entities. However, in this dissertation, we hypothesize that there are inter-dependencies between +/-effect events and their affected entities.

Therefore, we suggest a joint model to extract both +/-effect events and their affected entities. Since our input (i.e., a sentence) has structures and our output (i.e., +/-effect events and their affected entities) also has structures such as sequences and trees, we hypothesize that the approach for the structured prediction is appropriate for our task. Thus, we adopt the structured perceptron suggested by [Collins, 2002] for a joint model, as we mentioned in Section 9.4. In Section 9.5, we describe several features used in our system such as basic features, features for EffectWordNet, and features for relations between +/-effect events and affected entities.

To our knowledge, this work is the first work to extract +/-effect events and their affected entities jointly. In Section 9.6, the experiments show that our joint model is promising to extract +/-effect events and their affected entities jointly.

10.0 CONCLUSION AND FUTURE WORK

Past research in sentiment analysis has mainly addressed explicit sentiment expressions, ignoring implicit opinions expressed via implicatures. Recently, Deng et al., [Deng et al., 2013, Deng et al., 2014] introduce the opinion implicatures framework for inferring such implicit expressions. A fully automatic implementation of the framework requires that +/-effect event information should be recognized in a text. Thus, in this dissertation, we focus on +/-effect event information for opinion inferences. As we mentioned, ours is the first NLP research into developing a lexicon for events that have positive or negative effected on entities.

Due to significant sense ambiguity as we mentioned in Section 4.2.2, we need a sense-level approach to acquire +/-effect lexicon knowledge. We first present the feasibility of using WordNet for sense-level +/-effect lexicon acquisition with the bootstrapping method in Chapter 5. Our goal of this work is that starting from the seed set we explore how +/-effect events are organized in WordNet via semantic relations and expand the seed set based on those semantic relations. Our evaluations show the WordNet is promising for expanding sense-level +/-effect lexicons. Even though the seed set is completely independent from the corpus, the expanded lexicons coverage of the corpus is not small. The accuracy of the expanded lexicon is substantially higher. Also, the results of the agreement study are positive, providing evidence that the annotation task is feasible and that the concept of +/-effect gives us a natural coarse-grained grouping of senses.

Then, we address methods for creating a lexicon of +/-effect events with WordNet, called EffectWordNet in Chapter 6. One of our goals is to develop the method that applied to many verb synsets; and another goal is to build a lexicon with a small number of seed data. Also, we want to investigate whether the +/-effect property tends to be shared among semantically-related synsets. For that, we adopt a graph-based learning method

which is seeded by entries culled from FrameNet and then expanded by exploiting semantic relations in WordNet. Through experiments, we show that WordNet relations are useful for the polarity propagation in the graph model. Moreover, to maximize the effectiveness of each type of information, we combine a graph-based method using WordNet relations and a standard classifier using gloss information. A hybrid method gives the best results in +effect and -effect labels although the performance for the Null label is dropped. In addition, we present that the graph-based model is appropriate for WordNet relation information and the classifier is proper for gloss information in our task. Further, we provide evidence that the model is an effective way to guide manual annotation to find +/-effect words that are not in the seed word-level lexicon. This is important, as the likelihood that a random WordNet sense (and thus word) is +effect or -effect is not large.

Moreover, we present a graph-based method for constructing a sense-level +/-effect lexicon with consideration of affected entities called Enhanced EffectWordNet in Chapter 7. As we mentioned, the information about which entities are affected is important since the sentiment can be different in opinion inferences. Thus, we refine EffectWordNet with consideration of affected entities. Our experiments show that considering the information about which entities are affeced is helpful to construct more refined sense-level +/-effect lexicon.

To extract +/-effect events with a constructed sense-level lexicon, we have to carry out WSD. In this dissertation, we investigate a knowledge-based coarse-grained +/-effect WSD approach, which identifies the +/-effect of a word sense based on its surrounding context. The method we proposed relies on selectional preferences, and does not require any sense-tagged training data. Selectional preferences are modeled using LDA. We use automatic clustering based on the preference arguments, which is extracted from WordNet information, to create a sense inventory customized to our task. Through several experiments on a test dataset consisting of sense tagged data drawn from Senseval-3, we show that our method achieves very good performance, with an overall accuracy of 0.83, which represents a significant improvement over three competitive baselines. Moreover, we present the role of word sense clustering. In our experiments, the variable sense grouping (i.e., allow more than one sense group with the same label) outperforms the fixed sense grouping (i.e., one sense group for

each label) and fine-grained WSD (i.e., no grouping). Since it can have purer sense groups with the variable sense grouping, the topic model is able to better model the selectional preferences and provide more accurate results. In addition, we show that good performance can still be obtained using some automatic labels to fill out coverage.

Finally, we conduct a pilot study to jointly extract +/-effect events and their affected entities. These two tasks might be regarded as independent tasks, so they can be placed in a pipeline system such as firstly extracting +/-effect events and then identifying their affected entities. However, in this dissertation, we hypothesize that there are inter-dependencies between +/-effect events and their affected entities. Therefore, we propose a joint model to extract +/-effect events and their affected entities. For a joint model, we adopt the structured perceptron. The experiments show that our joint model is promising to extract +/-effect evetns and their affected entities simultaneously. To our knowledge, this research is the first work to extract +/-effect events and their affected entities jointly.

In all the evaluations, the accuracy of the lexicon construction is substantially high, but there is still room for improvement, especially for the +/-effect events of the lexicon. We believe that +/-effect judgments of word senses could be effectively crowd-sourced using a service such as Amazon Mechanical Turk (AMT); [Akkaya et al., 2010], for example, effectively used AMT for similar coarse-grained judgments. The idea would be to use automatic expansion methods to create a sense-level lexicon, and then have AMT workers judge the entries in which we have least confidence (e.g., +effect entries identified using the troponym relation). This would be much more time- and cost-effective than having workers judge senses randomly chosen from WordNet as a whole.

Moreover, since WordNet cannot cover all cases, corpus-based methods that have been applied to develop sentiment and connotation lexicons can be used to identify candidate words; annotating their senses could be crowd-sourced.

In addition, since this is the first work to extract +/-effect events and their affected entities jointly, there are several avenues for future work. Since the corpus for +/-effect events is not as large as other event corpora such as Automatic Content Extraction (ACE) [Doddington et al., 2004], it is hard to get a good result in the supervised learning methods. Thus, we need more annotated data containing implicit opinions based on the guideline

provided by Deng et al. [Deng et al., 2013]. Otherwise, we are interesting in adopting semi-supervised or unsupervised learning methods.

Another future work is to figure out more prominent features for +/-effect events. Since +/-effect events are broader concepts as we mentioned, it is difficult to detect them. As in Choi et al. [Choi and Wiebe, 2014, Choi et al., 2014], WordNet hierarchy information might be helpful. Also, it would be promising to utilize other resources such as FrameNet. In addition, although we consider only verbs as +/-effect events, we need to include other types of +/-effect events.

BIBLIOGRAPHY

[Agirre and Martínez, 2000] Agirre, E. and Martínez, D. (2000). Exploring automatic word sense disambiguation with decision lists and the web. In *Proceedings of the COLING-2000 Workshop on Semantic Annotation and Intelligent Content*, pages 11–19.

[Agirre et al., 2006] Agirre, E., Martínez, D., de Lacalle, O. L., and Soroa, A. (2006). Two graph-based algorithms for state-of-the-art wsd. In *Proceedings of EMNLP 2006*, pages 585–593.

[Ahn, 2006] Ahn, D. (2006). The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning About Time and Events*, ARTE '06, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

[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.

[Akkaya et al., 2011] Akkaya, C., Wiebe, J., Conrad, A., and Mihalcea, R. (2011). Improving the impact of subjectivity word sense disambiguation on contextual opinion analysis. In *Proceedings of CoNLL 2011*, pages 87–96.

[Akkaya et al., 2009] Akkaya, C., Wiebe, J., and Mihalcea, R. (2009). Subjectivity word sense disambiguation. In *Proceedings of EMNLP 2009*, pages 190–199.

[Akkaya et al., 2014] Akkaya, C., Wiebe, J., and Mihalcea, R. (2014). Iterative constrained clustering for subjectivity word sense disambiguation. In *Proceedings of the 14th EACL*, page 269278.

[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*.

[Araki and Mitamura, 2015] Araki, J. and Mitamura, T. (2015). Joint event trigger identification and event coreference resolution with structured perceptron. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2074–2080, Lisbon, Portugal. Association for Computational Linguistics.

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

[Azran, 2007] Azran, A. (2007). The rendezvous algorithm: Multiclass semi-supervised learning with markov random walks. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 49–56, New York, NY, USA. ACM.

[Baccianella et al., 2010] Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of LREC*, pages 2200–2204.

[Baker et al., 1998] Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Banerjee and Pedersen, 2003] Banerjee, S. and Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th IJCAI*, pages 805–810.

[Bao, 2012] Bao, H. T. (2012). Graphical models and topic modeling. University Lecture.

[Barbosa and Feng, 2010] Barbosa, L. and Feng, J. (2010). Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36–44, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Basile et al., 2014] Basile, P., Caputo, A., and Semeraro, G. (2014). An enhanced lesk word sense disambiguation algorithm through a distribution semantic model. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600.

[Belkin et al., 2006] Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

[Blum and Chawla, 2001] Blum, A. and Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of ICML*, pages 19–26.

[Blum et al., 2004] Blum, A., Lafferty, J., Rwebangira, M. R., and Reddy, R. (2004). Semi-supervised learning using randomized mincuts. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 13–, New York, NY, USA. ACM.

[Boyd-Graber et al., 2007] Boyd-Graber, J., Blei, D., and Zhu, X. (2007). A topic model for word sense disambiguation. In *Proceedings of EMNLP-CoNLL 2007*, pages 1024–1033.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of WWW7*, pages 107–117.

[Bui and Sloot, 2011] Bui, Q.-C. and Sloot, P. M. A. (2011). Extracting biological events from text using simple syntactic patterns. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, BioNLP Shared Task '11, pages 143–146, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Cai et al., 2007] Cai, J. F., Lee, W. S., and Teh, Y. W. (2007). Improving word sense disambiguation using topic features. In *Proceedings of EMNLP-CoNLL 2007*, pages 1015–1023.

[Carroll and McCarthy, 2000] Carroll, J. and McCarthy, D. (2000). Word sense disambiguation using automatically acquired verbal preferences. *Computers and the Humanities*, 34:109–114.

[Che and Liu, 2010] Che, W. and Liu, T. (2010). Jointly modeling wsd and srl with markov logic. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling)*, pages 161–169.

[Chen et al., 2014] Chen, X., Liu, Z., and Sun, M. (2014). A unified model for word sense representation and disambiguation. In *Proceedings of EMNLP 2014*, pages 1024–1035.

[Choi et al., 2014] Choi, Y., Deng, L., and Wiebe, J. (2014). Lexical acquisition for opinion inference: A sense-level lexicon of benefactive and malefactive events. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, pages 107–112.

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

[Collins, 2002] Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Collins and Roark, 2004] Collins, M. and Roark, B. (2004). Incremental parsing with the perceptron algorithm. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

[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.

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

[Davidov et al., 2010] Davidov, D., Tsur, O., and Rappoport, A. (2010). Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 241–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

[Deng et al., 2013] Deng, L., Choi, Y., and Wiebe, J. (2013). Benefactive/malefactive event and writer attitude annotation. In *Proceedings of 51st ACL*, pages 120–125.

[Deng and Wiebe, 2014] Deng, L. and Wiebe, J. (2014). Sentiment propagation via implicature constraints. In *Proceedings of EACL*.

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

[Deng 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*, page 7988.

[Doddington et al., 2004] Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The automatic content extraction (ace) program tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004)*, Lisbon, Portugal. European Language Resources Association (ELRA). ACL Anthology Identifier: L04-1011.

[Esuli and Sebastiani, 2006] Esuli, A. and Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of 5th LREC*, pages 417–422.

[Esuli and Sebastiani, 2007] Esuli, A. and Sebastiani, F. (2007). Pageranking wordnet synsets: An application to opinion mining. In *Proceedings of ACL*, pages 424–431.

[Feng et al., 2011] Feng, S., Bose, R., and Choi, Y. (2011). Learning general connotation of words using graph-based algorithms. In *Proceedings of EMNLP*, pages 1092–1103.

[Fillmore, 1977] Fillmore, C. J. (1977). The case for case reopened. *Syntax and Semantics 8: Grammatical Relations*, pages 59–81.

[Finkel and Manning, 2009] Finkel, J. R. and Manning, C. D. (2009). Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334, Boulder, Colorado. Association for Computational Linguistics.

[Go et al., 2009] Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classication using distant supervision. Stanford Digital Library Technical report.

[Godbole et al., 2007] Godbole, N., Srinivasaiah, M., and Skiena, S. (2007). Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

[Goyal et al., 2010] Goyal, A., Riloff, E., and DaumeIII, H. (2010). Automatically producing plot unit representations for narrative text. In *Proceedings of EMNLP*, pages 77–86.

[Griffiths and Steyvers, 2002] Griffiths, T. and Steyvers, M. (2002). A probabilistic approach to semantic representation. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, pages 381–386.

[Griffiths and Steyvers, 2004] Griffiths, T. and Steyvers, M. (2004). Finding scientific topics. In *Proceedings of the National Academy of Sciences 101 (Suppl. 1)*, pages 5228–5235.

[Griffiths and Steyvers, 2003] Griffiths, T. L. and Steyvers, M. (2003). Prediction and semantic association. pages 11–18.

[Grishman et al., 2005] Grishman, R., Westbrook, D., and Meyers, A. (2005). Nyus english ace 2005 system description. Technical report, Department of Computer Science, New York University.

[Gyamfi et al., 2009] Gyamfi, Y., Wiebe, J., Mihalcea, R., and Akkaya, C. (2009). Integrating knowledge for subjectivity sense labeling. In *Proceedings of NAACL HLT 2009*, pages 10–18.

[Hatzivassiloglou and McKeown, 1997] Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of ACL*, pages 174–181.

[Hatzivassiloglou and Wiebe, 2000] Hatzivassiloglou, V. and Wiebe, J. M. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, COLING '00, pages 299–305, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Hofmann, 1999] Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA. ACM.

[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*, KDD '04, pages 168–177, New York, NY, USA. ACM.

[Huang et al., 2012] Huang, L., Fayong, S., and Guo, Y. (2012). Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 142–151, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Ji and Grishman, 2008] Ji, H. and Grishman, R. (2008). Refining event extraction through cross-document inference. In *Proceeding ot ACL-08: HLT*, pages 254–262.

[Jiang and Conrath, 1997] Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of COLING*.

[Kang et al., 2014] Kang, J. S., Feng, S., Akoglu, L., and Choi, Y. (2014). Connotationwordnet: Learning connotation over the word+sense network. In *Proceedings of the 52nd ACL*, page 15441554.

[Kim et al., 2009] Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., and Tsujii, J. (2009). Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, BioNLP '09, pages 1–9, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Kim and Hovy, 2004] Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Kim and Hovy, 2006] Kim, S.-M. and Hovy, E. (2006). Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, SST '06, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Klapafits and Manandhar, 2010] Klapafits, I. P. and Manandhar, S. (2010). Word sense induction disambiguation using hierarchical random graphs. In *Proceedings of EMNLP 2010*, pages 745–755.

[Klein et al., 2002] Klein, D., Toutanova, K., Ilhan, H. T., Kamvar, S. D., and Manning, C. D. (2002). Combining heterogeneous classifiers for word-sense disambiguation. In *Proceedings of the ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions - Volume 8*, pages 74–80.

[Le Minh et al., 2011] Le Minh, Q., Truong, S. N., and Bao, Q. H. (2011). A pattern approach for biomedical event annotation. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, BioNLP Shared Task '11, pages 149–150, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Lee and Ng, 2002] Lee, Y. K. and Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of EMNLP 2000*, pages 41–48.

[Lesk, 1986] Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC 1986*, pages 24–26.

[Li et al., 2010] Li, L., Roth, B., and Sporleder, C. (2010). Topic models for word sense disambiguation and token-based idiom detection. In *Proceedings of the 48th ACL*, pages 1138–1147.

[Li et al., 2012] Li, P., Zhou, G., Zhu, Q., and Hou, L. (2012). Employing compositional semantics and discourse consistency in chinese event extraction. In *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1006–1016.

[Li and Ji, 2014] Li, Q. and Ji, H. (2014). Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

[Li et al., 2013] Li, Q., Ji, H., and Huang, L. (2013). Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

[Liao and Grishman, 2010] Liao, S. and Grishman, R. (2010). Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 789–797, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Liu, 2010] Liu, B. (2010). *Sentiment Analysis and Subjectivity.* Invited Chapter for the Handbook of Natural Language Processing, Second Edition.

[Liu et al., 2012] Liu, W., Wang, J., and Chang, S.-F. (2012). Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE*, 100(9):2624–2638.

[Manning et al., 2014] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

[McCarthy and Carroll, 2003] McCarthy, D. and Carroll, J. (2003). Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, 29(4):639–654.

[McClosky et al., 2011] McClosky, D., Surdeanu, M., and Manning, C. D. (2011). Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1626–1635, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Mei et al., 2007] Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C. (2007). Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 171–180, New York, NY, USA. ACM.

[Mihalcea, 2004] Mihalcea, R. (2004). Co-training and self-training for word sense disambiguation. In *Proceedings of CoNLL-2004*, pages 33–40.

[Miller et al., 1990] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography*, 13(4):235–312.

[Mohammad and Turney, 2010] Mohammad, S. M. and Turney, P. D. (2010). Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*.

[Mooney, 1996] Mooney, R. J. (1996). Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of EMNLP 1996*, pages 82–91.

[Navigli, 2009] Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

[Navigli and Lapata, 2010] Navigli, R. and Lapata, M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):678–692.

[Navigli and Ponzetto, 2012] Navigli, R. and Ponzetto, S. P. (2012). Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

[Ng, 1997] Ng, H. T. (1997). Getting serious about word sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, pages 1–7.

[O'Connor et al., 2010] O'Connor, B., Balasubramanyan, R., Routledge, B. R., and Smith, N. A. (2010). From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of ICWSM*.

[Paltoglou and Thelwall, 2012] Paltoglou, G. and Thelwall, M. (2012). Twitter, myspace, digg: Unsupervised sentiment analysis in social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4).

[Pang and Lee, 2005] Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd*

Annual Meeting on Association for Computational Linguistics, ACL '05, pages 115–124, Stroudsburg, PA, USA. Association for Computational Linguistics.

[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*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Patwardhan et al., 2005] Patwardhan, S., Banerjee, S., and Pedersen, T. (2005). Senserelate::targetword - a generalized framework for word sense disambiguation. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 73–76.

[Pedersen and Kolhatkar, 2009] Pedersen, T. and Kolhatkar, V. (2009). Wordnet::senserelate::allwords - a broad coverage word sense tagger that maximimizes semantic relatedness. In *Proceedings of NAACL/HLT 2009*, pages 17–20.

[Peng and Park, 2011] Peng, W. and Park, D. H. (2011). Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization. In *Proceedings of ICWSM*.

[Poon and Vanderwende, 2010] Poon, H. and Vanderwende, L. (2010). Joint inference for knowledge extraction from biomedical literature. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 813–821, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Rashkin et al., 2016] Rashkin, H., Singh, S., and Choi, Y. (2016). Connotation frames: A data-driven investigation. In *Association for Computational Linguistics (ACL)*.

[Resnik, 1995] Resnik, P. (1995). Using information content to evaluate semantic similarity. In *Proceedings of 14th IJCAI*, pages 448–453.

[Resnik, 1996] Resnik, P. (1996). Selectional constraints: an information-theoretic model and its computational realization. *Cognition*, 61:127–159.

[Riloff et al., 2013] Riloff, E., Qadir, A., Surve, P., Silva, L. D., Gilbert, N., and Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of EMNLP*, pages 704–714.

[Riloff and Wiebe, 2003] Riloff, E. and Wiebe, J. (2003). Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 105–112, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Riloff et al., 2005] Riloff, E., Wiebe, J., and Phillips, W. (2005). Exploiting subjectivity classification to improve information extraction. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI'05, pages 1106–1111. AAAI Press.

[Rooth et al., 1999] Rooth, M., Riezler, S., Prescher, D., Carroll, G., and Beil, F. (1999). Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of the 37th ACL*, pages 104–111.

[Sinha and Mihalcea, 2007] Sinha, R. and Mihalcea, R. (2007). Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In *Proceedings of the International Conference on Semantic Computing*, ICSC '07, pages 363–369, Washington, DC, USA. IEEE Computer Society.

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

[Stepinski and Mittal, 2007] Stepinski, A. and Mittal, V. (2007). A fact/opinion classifier for news articles. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 807–808, New York, NY, USA. ACM.

[Steyvers and Griffiths, 2007] Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models.

[Strapparava and Valitutti, 2004] Strapparava, C. and Valitutti, A. (2004). Wordnet-affect: An affective extension of wordnet. In *Proceedings of 4th LREC*, pages 1083–1086.

[Su and Markert, 2009] Su, F. and Markert, K. (2009). Subjectivity recognition on word senses via semi-supervised mincuts. In *Proceedings of NAACL HLT 2009*, pages 1–9.

[Subramanya and Bilmes, 2008] Subramanya, A. and Bilmes, J. (2008). Soft-supervised learning for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1090–1099, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Subramanya and Talukdar, 2014] Subramanya, A. and Talukdar, P. P. (2014). *Graph-based semi-supervised learning*. San Rafael, California: Morgan Claypool.

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

[Tong and Koller, 2001] Tong, S. and Koller, D. (2001). Support vector machin active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66.

[Tsatsaronis et al., 2007] Tsatsaronis, G., Vazirgiannis, M., and Androutsopoulos, I. (2007). Word sense disambiguation with spreading activation networks generated from thesauri. In *Proceedings of the 20th IJCAI*, pages 1725–1730.

[Turney and Littman, 2003] Turney, P. and Littman, M. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346.

[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*, ACL '02, pages 417–424, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Van de Cruys, 2014] Van de Cruys, T. (2014). A neural network approach to selectional preference acquisition. In *Proceedings of EMNLP 2014*, pages 26–35.

[Véronis, 2004] Véronis, J. (2004). Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252.

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

[Wiebe and Mihalcea, 2006] Wiebe, J. and Mihalcea, R. (2006). Word sense and subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 1065–1072, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Wiebe and Riloff, 2005] Wiebe, J. and Riloff, E. (2005). Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'05, pages 486–497, Berlin, Heidelberg. Springer-Verlag.

[Wiebe et al., 1999] Wiebe, J. M., Bruce, R. F., and O'Hara, T. P. (1999). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 246–253, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Wilson et al., 2005] Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Wilson et al., 2004] Wilson, T., Wiebe, J., and Hwa, R. (2004). Just how mad are you? finding strong and weak opinion clauses. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 761–767. AAAI Press.

[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, EMNLP '03, pages 129–136, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Zhou et al., 2004] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Scholkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16:321–329.

[Zhu et al., 2003] Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919.