OPINION SUMMARIZATION: AUTOMATICALLY CREATING USEFUL REPRESENTATIONS OF THE OPINIONS EXPRESSED IN TEXT

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Veselin Stoyanov Stoyanov

August 2009

OPINION SUMMARIZATION: AUTOMATICALLY CREATING USEFUL REPRESENTATIONS OF THE OPINIONS EXPRESSED IN TEXT

Veselin Stoyanov Stoyanov, Ph.D.

Cornell University 2009

Opinion analysis is concerned with extracting information about attitudes, beliefs, emotions, opinions, evaluations and sentiment expressed in texts. To date, research in the area of opinion analysis has focused on developing methods for the automatic extraction of opinions and their attributes. While this opinion information is useful, its true potential can be realized only after it is consolidated (summarized) in a meaningful way: the raw information contained in individual opinions is often incomplete and their number is overwhelming.

Until now, the task of domain-independent opinion summarization has received little research attention. We address this void by proposing methods for opinion summarization. Toward that end, we formulate new approaches for the problems of determining what opinions should be attributed to the same source (*source coreference resolution*) and whether opinions are on the same topic (*topic identification/coreference resolution*). Additionally, we introduce novel evaluation metrics for the quantitative evaluation of the quality of complete opinion summaries. Finally, we describe and evaluate OASIS, the first opinion summarization system known to us that produces domain-independent non-extract based summaries. Results for the individual components are encouraging and the overall summaries produced by OASIS outperform a competitive baseline by a large margin when we put more emphasis on computing an aggregate summary during evaluation.

BIOGRAPHICAL SKETCH

Veselin Stoyanov was born in Sofia, the capital city of Bulgaria. He graduated from high school in Sofia in 1996, following which he completed 18 months of mandatory military service in the Bulgarian army. Following discharge from the army, Veselin Stoyanov enrolled in Sofia University majoring in Informatics. After completing one year at Sofia university, he followed his family to the US and transferred to the University of Delaware majoring in Computer Science. In 2002 he was awarded Honors Bachelor of Science degree with Distinction by the University of Delaware graduating Summa Cum Laude. In the fall of the same year, Veselin Stoyanov enrolled in the PhD program at Cornell University's Department of Computer Science.

To Cristina and Martin.

In memory of our dear friend Caroline M. Coffey. Caroline, your dedication to science will always be an inspiration.

ACKNOWLEDGEMENTS

I have been very fortunate to be advised by a visionary and dedicated scientist (and an excellent technical writer), Claire Cardie. She has been instrumental to my professional development and an inspirational role model. But most of all I am thankful for her personal involvement and understanding.

I am indebted to the members of my committee, Lillian Lee, Thorsten Joachims and Shimon Edelman, for many useful comments on the research for my thesis in addition to teaching me many useful techniques and concepts. Thanks to Janice Wiebe at the University of Pittsburgh and Ellen Riloff at the University of Utah. Also thanks to members of the Cornell NLP group, Eric Breck, Regina Barzilay, Yejin Choi, Cristian Danescu-Niculescu-Mizil, Oren Kurland, Vincent Ng, Bo Pang, Ainur Yessenalina, for the many engaging discussions. I have also been fortunate to collaborate with Theresa Wilson, Diane Littman and Swapna Somasundaran at the University of Pittsburgh and Nathan Gilberth at the University of Utah. Thank you for the hard work and dedication.

I have spend the last two years of my tenure as a graduate student in New York City. I am thankful to Kathy McKeown and the NLP group at Columbia University for hosting me during that time and providing the support and inspiration to carry on with my research project. I also spent a summer at the Nara Institute of Science and Technology (NAIST). Thanks to Yuji Matsumoto, Kentaro Inui and members of the NAIST computational linguistics laboratory for hosting me there and making me feel at home.

I am also thankful to Larry Levy, David Pierce, Doran Mutsafi and Oran Leiba for allowing me to be part of an interesting adventure — a start-up company. Thanks for making my experience at Jodange so much fun. I have learned a great deal from being involved in the company.

Last but not least, I would not have been able to finish without the support and inspiration of my family and friends. I would like to thank my beautiful wife, Cristina, and my precious son, Martin. I also thank my parents, Sasha and Stoyan, for being an inspiration toward a career in science and for putting up with me for so many years. I am thankful to my sister, Yana, my friends Drago, Vladi and Adam as well as all other friends from Bulgaria, Delaware, Ithaca and New York City for their friendship and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The field of natural language processing (NLP) has exhibited rapid development in recent years, resulting in a number of practical tools. Many people around the world have become accustomed to using these tools in everyday life with great economic and social implications. To name two of the best known examples, we can hardly imagine our daily trip to the Web without using an information retrieval tool such as Google; in addition, speech recognition systems have helped telephone service companies to save millions of dollars.

Natural language technology research and systems, however, have primarily focused on the "factual" aspect of the analysis of the content of text (e.g. Baeza-Yates and Ribeiro-Neto (1999), Mani (2001), Cowie and Lehnert (1996)). Other aspects of text analysis, including pragmatics, point of view and style, have received much less attention. For many applications, however, to achieve an adequate understanding of a text, these aspects cannot and should not be ignored.

More specifically, many NLP applications might benefit from being able to represent and extract opinion information. Information retrieval systems (e.g. Baeza-Yates and Ribeiro-Neto (1999), Manning et al. (2008)), for instance, could be able to restrict retrieval to documents containing either factual or subjective information about a subject matter or to documents that express the point of view of a pre-specified entity. Document clustering (e.g. Zamir et al. (1997), Cutting et al. (1992)), which is a key component in a number of NLP applications, might form "better" clusters based on the opinion information; document summarization systems (e.g. Mani (2001), Kan et al. (2001)) might em-

ploy opinion information to produce more informative and accurate document summaries; and question answering (QA) systems (e.g. Ittycheriah et al. (2001), Moldovan et al. (2002)) might use opinion information both to produce more accurate answers to standard or factual questions for which they have been used so far, as well as to answer questions regarding opinions and perspectives.

In addition, being able to extract opinions and present them to the user in a way that makes it easy to comprehend and explore will be useful in its own right. Many professions (e.g. FBI analysts, company executives, and politicians) require dealing with opinions expressed in text as a part of the daily workload and this is presently done mostly manually. In the presence of a vast amount of information through the World Wide Web, the ability to quickly retrieve information about opinions is likely to be of interest even for the everyday user.

Motivated by these needs, the area of *opinion analysis*, concerned with automatically extracting attitudes, opinions, evaluations, and sentiment from text has received much recent research attention (see Related Work Chapter). To date, research in the area of opinion analysis has concentrated on developing methods for the automatic extraction of opinions. While opinion information as extracted by these methods (i.e. *raw opinion information*) can be useful, the true potential of this information can be realized only after the raw information is aggregated in a meaningful way. We will use the term *opinion summarization* to describe the process of meaningfully aggregating opinions and *opinion summary* to describe the resulting representation of the opinions. The ways of aggregating opinions and the resulting opinion summaries are described in more detail in Section 1.3.

Until now, the task of domain-independent opinion summarization has re-

ceived little research attention. This thesis intends to address this void. *The goal of this thesis is to develop effective methods for opinion summarization.* Specifically, we define two general forms for opinion summaries dictated by different application needs, identify the problems that need to be addressed by an opinion summarization system, develop methods to address these problems, introduce novel quantitative evaluation metrics for opinion summaries and construct and evaluate full opinion summaries for the documents in a standard opinion-oriented corpus (Wiebe et al., 2005b).

The rest of this chapter is organized as follows. We begin with a brief introduction to the field of opinion analysis in Section 1.1. We discuss our notion of opinion summary in Section 1.2, followed by a discussion of two opinion summary forms in Section 1.3. Next, we discuss the research challenges that need to be addressed by opinion summarization systems in Section 1.4. Finally, we summarize the contributions of this thesis in Section 1.5 and conclude the chapter with a roadmap for the rest of the thesis in Section 1.6.

1.1 Opinion Analysis

As previously defined, the area of opinion analysis is concerned with automatically extracting attitudes, opinions, evaluations and sentiment from text (e.g. Wiebe et al. (2005b), Bethard et al. (2004), Kim and Hovy (2004)). Research in the area (see Related Work section) can be split in two main subareas: *coarse-grained opinion analysis*, which is concerned with extracting sentiment orientation of whole documents (e.g. Pang et al. (2002), Turney (2002)) and *fine-grained opinion analysis*, which is concerned with extracting opinions at or below the sentence level – at the level of sentences, clauses, or individual expressions of

opinions (e.g. Bethard et al. (2004), Kim and Hovy (2004)). The work in this thesis falls in the latter area of fine-grained sentiment analysis.

To date, researchers have shown that fine-grained opinions as well as other aspects of opinions (such as their sources) can be extracted to a reasonable degree of accuracy (e.g. Bethard et al. (2004), Choi et al. (2006), Breck et al. (2007), Wilson et al. (2005), Kim and Hovy (2005)). This thesis assumes that we can rely on automatically extracted fine-grained opinions and their attributes. More precisely, we assume that each fine-grained opinion has the following four attributes:

1. Trigger – the word or phrase that signal the expression of opinion in the text. Opinion can be expressed either directly by words such as "said,"[1] "believes," or "argued" or indirectly through the choice of style and words in the language used (e.g. in the sentence "Saddam has repressed his people." the choice of the word "repressed" signals the author's negative opinion of Saddam).

2. Source – the entity to which the opinion is to be attributed. More precisely, we assume that automatic opinion extraction systems can recover the span of text (generally a noun phrase or pronoun) that specifies the entity to which the opinion is to be attributed. Researchers have also used *opinion holder* to refer to the source of an opinion. We consider both terms equally expressive and will use *source* for brevity.

3. Topic – the topic or target of the opinion. This could be either an entity (e.g. "Sue dislikes **John**") or a general topic (e.g. "I don't think that **lending**

[1]Many reporting verbs such as "said" can be expressing factual information. We follow other researchers (Wiebe and Riloff, 2005) and consider these reporting verbs to be opinion triggers only when the context signals expression of opinion.

money to close friends is a good idea").

4. Polarity – the sentiment (favorability) expressed in the opinion. For simplicity, we assume the polarity to be either positive (favorable opinion), negative (unfavorable opinion), or neutral (a non-judgmental opinion that does not express a favorable or unfavorable attitude).

Previous work has addressed extracting fine-grained opinion triggers, sources and polarity. This thesis assumes that it can rely on automatic extractors for fine-grained opinions with these three attributes. As discussed in Chapters 2 and 5, the problem of identifying topics of domain-independent fine-grained opinions lacks effective approaches. We address this problem in Chapter 5.

1.2 Opinion Summaries

While fine-grained opinion information can be useful as extracted by existing systems, researchers have argued that individual expressions of opinions will have to be aggregated into a summary representation to be fully useful (Cardie et al., 2003). An example of an opinion summary is shown in Figure 1.1. The example shows a text segment containing fine-grained opinions (above) and a summary of those opinions (below). In the text, sources and targets of opinions are bracketed; opinion expressions are shown in italics and bracketed with their associated polarity, either positive (+) or negative (-). In the summary, entities involved in opinions are shown as nodes and aggregated opinions are shown as directed edges. Opinions from the same source on the same topic are combined, statistics are computed for each source/topic, and multiple opinions from the same source on the same topic are aggregated.

[Source Zacarias Moussaoui] [− *complained*] at length today about [Target his own lawyer], telling a federal court jury that [Target he] was [− *more interested in achieving fame than saving Moussaoui's life*].

Mr. Moussaoui said he was appearing on the witness stand to tell the truth. And one part of the truth, [Source he] said, is that [Target sending him to prison for life] would be "[− *a greater punishment*] than being sentenced to death."

"[− [Target *You*] *have put your interest ahead of* [Source *my*] *life*]," [Source Mr. Moussaoui] told his court-appointed lawyer Gerald T. Zerkin.

...

But, [Source Mr. Zerkin] pressed [Target Mr. Moussaoui], was it [− *not true*] that he told his lawyers earlier not to involve any Muslims in the defense, not to present any evidence that might persuade the jurors to spare his life?

...

[Source Zerkin] seemed to be trying to show the jurors that while [Target the defendant] is generally [+ *an honest individual*], his conduct shows [Target he] is [− *not stable mentally*], and thus [− *undeserving*] of [Target the ultimate punishment].

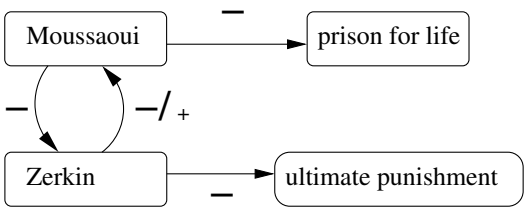

Figure 1.1: Example text containing opinions (above) and a summary of the opinions (below).

Opinion summaries similar to the one from the example allow opinion information to be presented to the user in a manner that is intuitive, concise and easy to explore and manipulate. Additionally, consolidated opinion information such as the one in Figure 1.1 is, arguably, more useful for NLP applications that take advantage of opinion information.

1.3 Opinion Summary Forms

We expect that applications will use summaries of fine-grained opinion information in two distinct ways, giving rise to two distinct summary formats — an *aggregate opinion* summary and an *opinion set* summary. Each type of summary relies on a different mechanism for combining multiple opinions from the same source about the same topic.

1.3.1 Aggregate opinion summary

In an *aggregate opinion* summary, multiple opinions from a source on a topic are merged into a single aggregate opinion that represents the cumulative opinion of the source on that topic considering the document as a whole. We discuss several different ways to aggregate individual opinions in Chapter 7. Note that Figure 1.1 depicts an aggregate opinion summary for the accompanying text.

Aggregate opinion summaries allow applications or users to access as a single opinion in a standardized format the overall view expressed in a collection of opinions by a source on a topic. They will be needed by applications such as question answering (QA). A QA system, for example, might need to answer questions such as "What is X's opinion toward Y?" Rather than report all of the places in the text where X expresses opinions on topic Y, the QA system only needs to report the overall accumulated opinion from X toward Y in a clean "database" form (although it will generally keep pointers to all of the contributing opinions as support for the answer). Aggregate opinions might also be employed for opinion-oriented information retrieval, clustering, opinion tracking, and document-level opinion exploration.

1.3.2 Opinion set summary

In an *opinion set* summary, multiple opinions from a source on a topic are simply collected into a single set (without analyzing them for the overall trend). An opinion set summary of the example in Figure 1.1 would include, for example, three directed links from *Moussaoui* toward *Zerkin* — one for each of the three expressions of negative opinion.

Opinion set summaries support fined-grained information extraction of opinions as well as user-directed exploration of the opinions in a document. In particular, they can be used to (1) identify all places in a text where entity E expresses an opinion (even though different expressions are used to refer to E), or to (2) identify all places in the text where an opinion on topic T is expressed. Systems that are concerned with mining the perceived strengths/weaknesses of a given entity (e.g. a product in the case of product reviews) or the arguments in favor/against a given topic rather than only the sentiment (useful, for example, for analysing public opinion for the benefit of politicians, or foreign entities' opinions for the benefit of intelligence analysts) can similarly use opinion set summaries to drive their analyses.

Although the two types of opinion summary are related, evaluating summaries geared toward each type requires different methodology. A detailed discussion of opinion summaries and their evaluation appears in Chapter 6.

1.4 Challenges in Opinion Summarization

Creating and evaluating opinion summaries requires solving a number of research challenges. These challenges are not specific to our the study of opinion summarization; they also extend to other NLP and ML tasks. In this section we set the stage for this thesis by briefly discussing each of the main research challenges in a subsection. More details for each of the problems, our proposed approaches, connections to other NLP and ML tasks are given in later chapters of this thesis.

1.4.1 Source Coreference Resolution

A big part of constructing the opinion summaries consists of determining which sources of opinions refer to the same real-world entity. We refer to this task as *source coreference resolution*.

1.4.2 Topic Determination/Coreference Resolution

Equally important is to determine which opinions discuss the same topic. This problem, which we refer to as *topic coreference resolution*, is further complicated by the lack of general opinion corpora that contain information about the topic of fine-grained opinions.

1.4.3 Evaluation

In order to be able to compare empirically different approaches to opinion summarization, this thesis develops methods and measures for quantitatively assessing the quality of opinion summaries. Using these methods, we can com-

pare an automatically generated summary to a gold standard opinion summary constructed from the available manually annotated fine-grained opinion information. The purpose of the evaluation measures is to compare automatic summaries to the gold standard and assign a numeric score to the summary that reflects the summary's "goodness."

1.5 Contributions

To the best of our knowledge, this thesis is the first work that addresses the problem of creating opinion summaries for general, domain-independent fine-grained opinions. As a result, our work contributes the first extended discussion of different aspects of opinion summarization such as the form of the summaries and the research problems involved in creating opinion summaries. More importantly, the thesis addresses the identified research problems concerning the creation of opinion summaries:

Usability Study. Like other work in the area of fine-grained sentiment analysis, our work is based on the hypothesis that fine-grained opinion information can be used successfully in NLP applications. While previous work has argued in favor of this hypothesis, this conjecture has been supported by little empirical evidence. Thus, we deem it important to empirically confirm the usefulness of fine-grained opinion information for NLP applications. This thesis includes one of the first experimental studies that shows empirically that such opinion information can be useful for an NLP application. More precisely, we show that fine-grained opinion information can be used successfully for the task of Multi-Perspective Question Answering (MPQA). The study is described in Chapter

3.

Source Coreference Resolution. One of the steps in opinion summarization is linking together opinions that belong to the same source – *source coreference resolution*. This thesis includes the first approach to the problem of source coreference resolution. In particular, we define and treat source coreference resolution as a partially supervised version of noun phrase coreference resolution. The partially supervised nature of the problem leads us to approach it as the more general, but also novel, problem of partially supervised clustering. In Chapter 4, we propose and evaluate a new algorithm for the task of source coreference resolution that outperforms competitive baselines.

Topic Identification. Topic identification has received little research attention due to both the difficulty of the task and the lack of appropriately annotated resources. This thesis addresses the problem of topic identification for fine-grained opinion analysis of general text. We provide a new, operational definition of *opinion topic* in which the topic of an opinion depends on the context in which its associated opinion expression occurs. We also present a novel method for general-purpose opinion topic identification that, following our new definition, treats the problem as an exercise in topic coreference resolution. We add manual annotations that encode topic information to an existing opinion corpus and use it for evaluation. Our approach achieves topic coreference scores that statistically significantly outperform two topic segmentation baselines across three different coreference resolution evaluation measures. Topic identification and coreference is the subject of Chapter 5.

Evaluation Measures. There are no "natural" evaluation metrics that quantitatively assess the quality of an automatically generated opinion summary as compared to a gold standard. Additionally, we are not aware of any previous work that has suggested evaluation metrics for structures similar to those of the opinion summaries. In this thesis, we propose two evaluation metrics for opinion summaries inspired by evaluations in information extraction and noun phrase coreference resolution. These are presented in Chapter 6.

Generating and Evaluating Complete Opinion Summaries. To the best of our knowledge, this thesis contains the first published work that generates and evaluates rich domain-independent opinion summaries. An overview and evaluation of our complete system is presented in Chapter 7.

1.6 Roadmap

In this chapter, we gave a brief overview of opinion analysis, focusing on fine-grained opinion analysis. We also discussed the need for opinion summaries, the form of these summaries and the remaining problems that need to be addressed in order to create completely automatic opinion summaries. We concluded by presenting the contributions of this thesis.

The rest of this thesis is organized as follows. We first overview related work in the area of opinion analysis in Chapter 2. We continue by describing in Chapter 3 the results of our experimental usability study which empirically shows that fine-grained opinion information is useful for an NLP application. The results of this study confirm the importance of the work described in the thesis. We then discuss our work on opinion summarization addressing the problems

of source coreference resolution in Chapter 4 and topic identification in Chapter 5. In Chapter 6 we address the issues of quantitative evaluation of opinion summaries by describing our novel evaluation metrics. Finally, in Chapter 7 we describe and evaluate our system, OASIS, which generates complete automatic opinion summaries for documents, paragraphs, or arbitrary text snippets.

CHAPTER 2

RELATED WORK

In this chapter we describe existing research in the area of opinion analysis. Work related to other problems that we address (i.e., source coreference resolution, topic identification and evaluation) is discussed in the appropriate chapters.

As mentioned in Chapter 1, the area of *opinion analysis* is an area of NLP concerned with automatically extracting attitudes, opinions, evaluations and sentiment (Wiebe et al., 2005a; Bethard et al., 2004; Pang and Lee, 2008)[1]. Other terms used to refer to opinion analysis include *opinion mining*, *sentiment analysis*, *sentiment extraction*, *subjectivity analysis*, *appraisal extraction* and has some connections to *affective computing* (see Pang and Lee (2008) for an interesting in-depth discussion of terminology). All of these terms are roughly equivalent, but carry somewhat different connotations with respect to the task that is being attempted. We prefer the name *opinion analysis* (or *opinion mining*) for our work to emphasize that, in addition to sentiment-bearing opinions (e.g. "Joe likes New York"), the task that we attempt includes extraction of opinions that may not carry sentiment or where the sentiment may be difficult to determine. For example the sentence "I believe that all bears are brown" contains an opinion, but, arguably, not any particular sentiment.

As discussed previously, opinion analysis research can be split into two general categories based on the granularity of the extracted opinions. These broad categories are coarse-grained opinion classification concerned with opinions at

[1]More precisely, building on other work in the area (e.g. Wiebe et al. (2005a), Bethard et al. (2004), Kim and Hovy (2004)) we are interested in extracting information about opinions in text signaled through the use of *subjective language*.

the document level and fine-grained opinion extraction concerned with opinions at the sentence level or below. Since these two levels of granularity have been subject of different approaches, we will follow this distinction in our discussion and devote a section for each of these two categories below.

2.1 Coarse-grained Opinion Extraction

Some of the pioneering work in opinion analysis was done in the area of coarse-grained opinion extraction. Work in this area has been approached as a text categorization task in which the goal is to assign to a document either positive ("thumbs up") or negative ("thumbs down") polarity or as a regression task in which the goal is to associate a favorability rating (e.g. number of stars) with a document (e.g. Das and Chen (2001), Pang et al. (2002), Turney (2002), Dave et al. (2003), Pang and Lee (2004)). Researchers have cleverly taken advantage of available electronic texts (mostly in the form of product reviews[2] from the Web) that contain numerical ratings (sometimes in the form of stars) to inexpensively create several corpora (e.g. Das and Chen (2001), Pang et al. (2002), Turney (2002)). Helped by the availability of these corpora a number of different approaches to sentiment classification have been proposed. Since our work falls in the area of fine-grained opinion analysis, it is less closely related to coarse-grained opinion extractions, so we review only a few of the pioneering works in the area of sentiment classification. See Pang and Lee (2008) for an in-depth discussion of coarse-grained (and fine-grained) opinion analysis.

Turney (2002) uses a simple unsupervised learning method to classify re-

[2]We will use the term *product review* to refer to reviews of a wide range of consumer products (e.g. electronics such as digital cameras, cars, CDs), services (e.g. hotel rooms and restaurants) as well as other entities such as movies.

views as recommended (*thumbs up*) or not recommended (*thumbs down*). Turney computes the orientation of a review by averaging the semantic orientation of phrases in the review that contain adjectives and adverbs. The semantic orientation of a phrase is based on its semantic relatedness to positive and negative terms. More precisely, for each adjective and adverb phrase, Turney computes the pointwise mutual information as the mutual information between the given phrase and the word "excellent" minus the mutual information between the given phrase and the word "poor" as determined by a web search engine. A review is classified as recommended if the average semantic orientation of its phrases is positive. Turney uses for evaluation 410 reviews from Epinions.com taken from four domains (automobiles, banks, movies, and travel destinations) and achieves an average accuracy of 74%. The accuracy ranges from 84% for automobile reviews to 66% for movie reviews.

Pang et al. (2002) perform a similar task – positive/negative review classification – in the domain of movie reviews. Using a bag-of-words representation and three learning algorithms (Naive Bayes, maximum entropy classification and support vector machines) they achieve accuracy of 84%, which outperforms human-constructed baselines. The performance of the classifiers, however, is not as good as for traditional topic-based categorization. Pang et al. conclude that the task of sentiment classification is more challenging than traditional classification.

In a subsequent effort, Pang and Lee (2004) improve the performance of the classifier to 86% by extracting review summaries. Their notion of summaries is quite different from ours – for them, a summary consists of the set of subjective sentences in a document. Like Pang and Lee, we identify the subjective

sentences in a document, but, in contrast, we aim to summarize those sentences rather than return the set as the summary. To identify the subjective sentences in a document, Pang and Lee train a classifier using a large, automatically created web corpus. Predictions from the classifier are incorporated in a minimum-cut formulation for the purpose of enforcing cross-sentence contextual constraints. Because their corpus has no supervisory "objective" vs. "subjective" labels, Pang and Lee do not evaluate directly the performance of the sentence level subjectivity classifier.

Tong (1999) is concerned with a slightly different aspect of coarse-grained opinion summarization. He is interested in the "buzz" surrounding a movie for the purpose of marketing research. His system relies on hand-built lexicons of terms, the proximity of lexicon terms and mentions of movies and ordering rules to construct a timeline of the buzz and sentiment surrounding a movie.

Das and Chen (2001) develop methods for extracting small investor sentiment from stock message boards. For the task they rely on several different classifiers combined through a voting scheme. Empirical evaluation shows some relation with stock values at the sector level – the aggregate sentiment of a sector is found to predict the sector index levels, but not the prices of individual stocks.

Dave et al. (2003) develop a method for automatically classifying product reviews from the web (Amazon and C|Net) into positive and negative. Their methods draw on information retrieval techniques starting with simple unigram models and applying a variety of techniques to develop more complex models. Adding a variety of semantic and syntactic information proves ineffective, but adding N-gram features and feature weighting show some improve-

ment. In addition, Dave et al. apply their approach on individual sentences collected from web searches and find that the performance is limited due to the limited textual content and lack of redundancy.

2.2 Fine-grained Opinion Extraction

Work in this thesis falls in the area of fine-grained opinion extraction, which is concerned with sentiment analysis at or below the sentence level. Research in the area includes a wide variety of approaches adapted for different definitions of opinions, domains and aspects of opinions. For the ease of presentation, we organize our presentation of fine-grained opinion research into several categories. We begin by discussing fine-grained opinion extraction from product reviews, which, we argue, has been tackled using very different approaches and definitions due to several domain differences that we outline. We continue with a discussion of methods for general, domain-independent fine-grained opinion analysis (we will also use *domain-independent opinion analysis* to refer to the latter), which is intimately related to our work. We conclude by discussing research on opinion summarization including both the product review and general news and editorial genres.

2.2.1 Fine-grained Opinion Analysis of Product Reviews

Fine-grained opinion analysis of product reviews is also referred to as *review mining*. Most of the coarse-grained opinion extraction efforts discussed in Section 2.1 can be considered to fall in the area of review mining, although the two problems differ in the way that they are typically approached. Fine-grained review mining is similar to domain-independent fine-grained opinion analysis –

both aim to identify fine-grained opinions. However, due to some specificity of how opinions are expressed in product reviews, review mining has employed approaches that can be considered special cases of domain-independent opinion analysis. The following list contains some of the most important differences:

1. **Sources are known.** Most opinions of interest in product reviews can be attributed to the author of the review. Therefore, extracting sources in fine-grained review mining is a rather trivial task as is source coreference resolution.

2. **Topics are limited.** Review-mining approaches are interested only in those opinions that are about a specific product and its features or attributes (often labeled *aspects*). Furthermore, reviews in product review corpora are almost always labeled with the product that is being discussed in the review. These two properties simplify significantly the task of extracting opinion topics in review mining. Effectively, topics can been limited to a list of features for the specific product or product class and product topic extraction can be conducted by looking up words or phrases in a lexicon of attribute terms, which can be constructed either manually, automatically or semi-automatically.

3. **The opinion lexicon is domain-dependent.** In review mining, the sets of words that express positive and negative sentiment differ based on the product being reviewed. For example, "big" is a positive term when discussing digital camera screens, but negative when talking about cell phones, for example. Some of the work in review mining has been able to take advantage of this domain specificity, crucially relying on methods to automatically induce an opinion lexicon for each domain. In

contrast, domain-specific methods are much less prominent in domain-independent fine-grained opinion analysis.

Due to the above differences, work in fine-grained review mining employs methods that are very dissimilar from those used for general fine-grained opinion analysis. Therefore, in the rest of this subsection, we review only a few example works in fine-grained review mining to illustrate the methods that are used.

Kobayashi et al. (2004) propose a semi-automatic method for collecting evaluative patterns that are used to extract sentiment about products, with each expression of sentiment being a triple of <Subject, Attribute, Value> corresponding to product, product feature and polarity. Their method relies on co-occurrence patterns of evaluated subjects, focused attributes and value expressions. In a subsequent effort Kobayashi et al. (2005) use the same representation of opinions as triples and propose a computational method for extracting these opinions. This is done by splitting the task into the subtask of extracting Attribute-Value pairs and the subtask of judging whether an extracted pair expresses an opinion. Kobayashi et al. use machine learning techniques for both tasks.

Kanayama et al. (2004) use machine translation technology to extract opinions about products represented as opinion triples similar to Kobayashi et al. (2004). Defining the problem as translating from text documents to sentiment units, Kanayama et al. are able to develop a surprisingly accurate system at low development cost. In another work, Kanayama and Nasukawa (2006) propose a method for the unsupervised building of domain-dependent lexicons that can be used to detect clauses that convey positive or negative sentiment.

The method is bootstrapped with a domain-independent lexicon and depends on context coherency, i.e., the tendency for the same polarities to appear in successive contexts.

Popescu and Etzioni (2005) decompose the problem of review mining into four subtasks: (1) identify product features, (2) identify opinions regarding product features, (3) determine the polarity of opinions, and, (5) rank opinions based on their strength. They introduce OPINE, an unsupervised information extraction system that follows the above decomposition and includes a component for each of the above subtasks.

2.2.2 Domain-Independent Fine-Grained Opinion Analysis

In this subsection we discuss work in the area of domain-independent fine-grained opinion analysis, in which our work falls. We begin by giving a brief overview of the different definitions of opinion and some publicly available general fine-grained corpora. We continue by discussing some of the approaches employed for fine-grained opinion extraction.

Definitions of Opinion and Fine-Grained Opinion Corpora

Contrary to popular belief, defining what constitutes an expression of opinion is not an easy task. The literature contains several comprehensive definitions, which we discuss briefly in this subsection. Together with the definitions, we talk about language resources that have been created using the corresponding definition.

Wiebe et al. (2005b) center their definition of *subjective expression* around the notion of private state, a general term that covers opinions, beliefs, thoughts,

feelings, emotions, goals, evaluations, and judgments. As Quirk et al. (1985) define it, a private state is a state that is not open to objective observation or verification. Furthermore, Wiebe et. al view private states in terms of their functional components, which correspond to the opinion attributes that we list in Chapter 1. More precisely, their text anchor, source, target and attitude type roughly correspond to our trigger, source, topic and polarity, respectively.

Using their definition of opinion, Wiebe et al. (2005b) propose an extensive annotation scheme for subjective expressions and use it to create the MPQA corpus. The MPQA corpus in its first release (Version 1.2) contains 535 documents manually annotated for phrase-level expressions of opinions, their sources, polarities, and intensities. The second release of the corpus (Version 2.0) adds 157 documents for a total of 692 and adds target span annotations.

Bethard et al. (2004) define opinions as a sentence or part of a sentence that would answer the question "What does X feel about Y?". They define a propositional opinion as an opinion localized in the propositional argument of a verb (generally functioning as the sentential complement of a predicate). The goal of Bethard et al. is to identify propositional opinions and their holders. Examples of propositional opinions are the complements of predicates such as *believe*, *realize*, and *reply*, as in the underlined part of the sentence "I *believe* Tom is honest."

Bethard et al. (2004) create a corpus containing 5,139 sentences annotated for opinions at the sentence level. Each sentence is labeled with three tags NON-OPINION, OPINION-PROPOSITION and OPINION-SENTENCE to indicate that the sentence contains no opinion, an opinion contained in an propositional verb argument and an opinion outside of such an argument, respectively. The annotations also contain information about the holders (sources) of some of

the propositional opinions.

Kim and Hovy (2004) use Bethard et al.'s (2004) definition of opinion. Similar to us, Kim and Hovy define an opinion as a quadruple [Topic, Holder, Claim, Sentiment] in which the Holder (corresponding to our Source) believes a Claim (the equivalent of our opinion trigger) about the Topic, and in many cases associates a Sentiment (i.e. Polarity), such as good or bad, with the belief. Kim et al. create a small (e.g. 100 sentences) manually annotated corpus with sentence-level tags.

Work in this thesis relies heavily on the MPQA corpus – we are not aware of any other corpus that rivals the scale and depth of the MPQA corpus, including the corpora discussed in this section. By using the MPQA corpus, we indirectly rely on Wiebe et al.'s (2005b) definition of opinion. Approaches presented in this thesis, however, aim to be applicable with most available definitions of opinions. Our only assumptions are that expressions of opinions can be defined in a way that allows for reliable annotation and that opinions are defined in terms of the four components that we discussed previously: opinion trigger, source, topic and polarity.

Extraction of Fine-Grained Opinions and Their Attributes

A number of research efforts in the area of fine-grained opinion extraction have approached the problem as one of classification (e.g. Bethard et al. (2004), Riloff and Wiebe (2003), Wiebe and Riloff (2005), Wilson et al. (2004)). Those works have attempted to classify sentences, clauses, phrases, or words on one of two (related) axes: subjective vs. objective (with possibly different degrees of subjectivity) and expressing positive vs. negative sentiment (again with possible

different strengths and optionally including neutral sentiment). Next we give an overview of several such classification methods.

Nasukawa and Yi (2003) describe a system for analyzing the sentiment (positive or negative) toward specific subjects that relies on semantic analysis based on a syntactic parser and sentiment lexicon. Their system achieves precision of 75 to 95% on a corpus of web pages and news documents. An evolution of the system, described in Yi et al. (2003), utilizes components that perform topic-specific feature extraction, sentiment extaction, and (subject, seniment) relationship analysis. This system exhibits improved performance both on web pages and news articles as well as on a corpus of product reviews.

Bethard et al. (2004) define and attempt a new task in opinion analysis – identifying opinion-bearing propositions as well as the holders of these opinions. Their definition of propositional opinions is discussed in Section 2.2.2. Using machine learning techniques (which include a one-tiered and a two-tiered classification architecture), a number of linguistic resources such as FrameNet, PropBank, and opinion word lists, Bethard et al. achieve F-measure in the 50's on the task of propositional opinion identification. They also implement an opinion holder identifier, which is only slightly less accurate due to the fact that in 90% of the propositional opinions in their corpus, the opinion holder is at the same syntactic position in relation to the proposition.

Riloff and Wiebe (2003) develop a method for extracting subjective expressions from unannotated text using bootstrapping. Their method makes use of lexico-syntactic patterns, which have been used successfully for information extraction. Riloff and Wiebe's system extracts an initial set of subjective sentences by using high-precision subjectivity classifiers, which rely on a list of subjective

lexical items. Subsequently, the initial set is used to learn extraction patterns, which are then used to expand the initial set. This iteration is repeated until no more patterns can be added to the set. Empirical evaluation shows that the final set of extraction patterns can be used as a high precision sentence-level subjectivity classifier (with precision of over 90% and recall between 32.9% and 40.1% for two different implementations).

In a subsequent effort, Wiebe and Riloff (2005) use their high-precision sentence-level subjectivity recognizer as well as their lexicon of subjective clues to create a sentence-level subjective/objective classifier from unannotated data. Their method uses the presence/absence of subjective words to classify sentence as subjective or objective, and an approach similar to Riloff and Wiebe (2003) to learn subjective and objective extraction patterns. The lexicon and extraction patterns are combined into one classifier, the predictions on the training data of which are used to train a Naive Bayes classifier. Using self training, the predictions of the Naive Bayes classifier are then used to retrain the extraction pattern learner and the whole process is repeated. The final classifier achieves F-measure of 78.1 for the task of subjective sentence classification and 73.4 for objective sentence classification.

Wilson et al. (2004) are interested in recovering the strength of opinions at the clause level (including deeply nested clauses). Their approach makes use of a set of previously established subjectivity clues such as a subjectivity lexicon and the extraction patterns from Riloff and Wiebe (2003). They introduce a new set of syntactic clues developed for opinion recognition. Using boosting, rule-learning, and a support vector regression algorithm, Wilson et al. achieve reasonable levels of accuracy at all levels of nesting.

Several research efforts are concerned with classifying words and/or phrases with respect to their *sentiment orientation*, which is a measurement of the a priori sentiment that the words or phrases express taken out of context (e.g. "great" has positive semantic orientation, while "insufficient" has negative). Takamura et al. (2005) use a spin model to extract the semantic orientation of words. Their method starts with a small number of seed words and uses an energy minimization method (mean field approximation) to compute the semantic orientation of the non-seed words. In another effort, Takamura et al. (2006) compute the semantic orientation of phrases, using latent variable models and expectation-maximization (EM) based methods.

Other researchers have approached domain-independent fine-grained opinion analysis as an information extraction task. In this setting, the goal is to extract "opinion templates" – the equivalent of information extraction templates. Opinion templates constitute of the slots that need to be filled, such as the expressions of opinions in text together with a set of attributes (e.g., the source and/or the polarity of the opinion). For example, a template could contain three slots <Opinion Trigger, Source, Polarity>. Note that the opinion five-tuple <Opinion Trigger, Source, Topic, Polarity, Strength> on which our research relies is conceptually equivalent to a five-slot opinion template.

The distinction between classification approaches discussed above and extraction approaches is rather superficial because the two tasks can be often cast in terms of each other. Nevertheless, the opinion templates on which the in-

formation extraction definition relies are intimately related to our requirements for extracted opinions. Next, we review several of the most important works in fine-grained opinion analysis as an information extraction task.

Similar to us, Kim and Hovy (2004) are interested in opinions as quadruples of [Topic, Holder, Claim, Sentiment]. More specifically, Kim and Hovy address the following problem: given a Topic and a set of documents on the Topic, find the Sentiments and Claims expressed about the Topic as well as the Holders of the sentiments. The problem resembles a subproblem of opinion summarization – create an opinion summary of all opinions on a given topic (assuming opinions in each document address one topic). Kim and Hovy simplify the problem by only identifying expressions of positive, negative and neutral sentiment together with their holders, while ignoring the full topic extraction problem. The algorithm proposed by Kim and Hovy works in four stages. It begins by selecting sentences that contain both the topic phrase and holder candidates (the only candidates for holders are noun phrases that are tagged PERSON and ORGANIZATION by a named-entity finder). Next, the holder-based regions of the opinion are delimited. Then Kim and Hovy employ a word-level sentiment classifier to calculate the polarity of each sentiment-bearing word in isolation. Finally, they sum sentiment orientation for individual words to form cumulative sentiment for the sentence and return the result.

Kim and Hovy (2004) use for evaluation a small manually annotated corpus (mentioned in Section 2.2.2). They evaluate their system on sentiment classification at the word and sentence level. For the overall opinion identification task, they judge an opinion to be correctly identified if the system finds both the correct holder and the appropriate sentiment within the sentence (topic identi-

fication is not judged as the sentences are assumed relevant to the topic). Under this evaluation, Kim and Hovy's best model performed at 81% accuracy when provided with manually identified (gold standard) holders and at 67% when automatically identifying the holders.

Subsequently, Kim and Hovy (2006b) present an effort in which they are concerned with identifying judgment opinions. For this work, general opinions are split into two (overlapping) categories – 1) beliefs about the world, which can have values such as true, false, and likely; and 2) judgments about the world with values such as good, bad, neutral, wise and foolish. Based on their belief that judgment opinions are more easily identifiable, which was confirmed in a NIST-sponsored pilot study, Kim and Hovy concentrate only on the second kind of opinion, which they term *judgment opinions*. As in previous work, Kim and Hovy build a staged system, which begins by identifying opinion words and the valence (polarity) of the opinion and follow it with a module that identifies the opinion holder. The former module makes use of WordNet, while the latter is trained on the MPQA corpus. In this work, Kim and Hovy do not address the task of topic identification. Kim and Hovy's system achieves a F1 score in the fifties for overall opinion identification on a corpus of German emails.

Finally, Kim and Hovy (2006a) present a method for extracting opinions that include all four aspects [Topic, Holder, Claim, Sentiment]. Their approach is based on semantic role labeling as an intermediate step. As with previous approaches, opinion identification is performed in several steps beginning with identification of opinion words. In contrast to their previous algorithms, however, opinion word identification is followed by a step that labels the semantic roles of the words in each sentence utilizing an algorithm trained on data from

FrameNet. Opinion words are then mapped to frames, when possible, and the holder and topic of the opinion are extracted as the arguments that carry particular semantic roles for the particular frame based on the frame type. Evaluated on a manually created opinion corpus, Kim and Hovy's system achieves F1 between 55 and 64 on the task of opinion-bearing sentence identification and in the thirties for the tasks of opinion holder and topic extraction.

Breck et al. (2007) present an approach for identifying direct opinion expressions (i.e., opinion triggers) that uses conditional random fields and evaluate the approach using the MPQA corpus. Their approach achieves expression-level performance that is within 5% of the human interannotator agreement.

Choi et al. (2005) are interested in extracting sources of opinions. They start with two approaches: automatically acquire extraction patterns and learn a Conditional Random Field (CRF) segmenter, which approaches the task as a sequence labeling problem. Using a hybrid approach, which incorporates the extraction patterns as features of the CRF, Choi et al. achieve performance that is better than either approach alone. The resulting system identifies opinion sources with 79.3% precision and 59.5% recall using a head noun matching measure, and 81.2% precision and 60.6% recall using an overlap measure.

In a subsequent effort, Choi et al. (2006) combine the source extractor from (Choi et al., 2005) and the opinion trigger sequence tagger from (Breck et al., 2007). The combination is done by explicitly considering the linking relation between sources and opinion triggers and enforcing the consistency through the use of linear programming. Choi et al. show that global, constraint-based inference can significantly boost the performance of both the extraction of opinion-related entities (i.e. sources and opinion triggers) and relation extraction (i.e.

Nikos Fine Dining
Food **4/5** "Best fish in the city", "Excellent appetizers"
Decor **3/5** "Cozy with an old world feel", "Too dark"
Service **1/5** "Our waitress was rude", "Awful service"
Value **5/5** "Good Greek food for the $ ", "Great price!"

Figure 2.1: An example of feature-based opinion summary for a service.

the "source expresses opinion" relation). In addition, Choi et al. employ semantic role labeling to arrive at a system that achieves F-measures of 79 and 69 for entity and relation extraction, respectively.

2.2.3 Opinion Summarization

As we claim in Chapter 1, we are not aware of any previous work that attempts to perform domain-independent fine-grained opinion summarization in the form that we suggest. However, several efforts in opinion summarization have been published that differ either by being restricted to the product review domain or by targeting different representation for summaries. We discuss these efforts in this subsection starting with opinion summarization in the product mining domain and continuing with domain-independent efforts.

Several approaches have successfully constructed useful summaries in the product review domain (Hu and Liu, 2004; Popescu and Etzioni, 2005; Gamon et al., 2005; Carenini et al., 2006; Zhuang et al., 2006; Snyder and Barzilay, 2007; Titov and McDonald, 2008; Lerman et al., 2009). These summaries, sometimes referred to as *feature-based summaries* or *aspect summaries* are produced by using as input a corpus of product reviews for a product or a service and producing a set of relevant features (aspects), the aggregate sentiment for each feature

plus, optionally, a few supporting text segments. An example of feature-based summary from Titov and McDonald (2008) is shown in Figure 2.1. In the example, which comes from the restaurant domain, features include the quality of the food, the decor, etc. The goal of feature-based opinion summarization is to discover that, for example, food quality is a feature in this domain, aggregate the sentiment expressed in the corpus for the food quality of a particular restaurant and, optionally, give a few anecdotal examples of text that support the aggregated sentiment. Constructed in this way, opinion summaries are conceptually similar to the summaries that we propose – they group together and aggregate opinions on the same topic (e.g., same feature). However, due to the previously mentioned differences (see Section 2.2.1), approaches that are used in the product review domain are unlikely to be efficient in the general domain.

Efforts in opinion summarization in the general domain have been spurred by the inclusion of opinion tracks in the Text REtrieval Conference (TREC) (Ounis et al., 2007; Macdonald et al., 2008; Ounis et al., 2009) and subsequently in the Text Analysis Conference (TAC) (Dang, 2008). Opinion evaluation started in the 2006 TREC blog track with the *opinion-finding task* and continued with some changes in the 2007 and 2008 TREC blog tracks. We include the 2006-2008 TREC blog tracks in our discussion only for historical reasons since these tasks do not constitute fine-grained opinion summarization.

The TREC opinion-finding task aims to address a search scenario where the goal is to discover what bloggers think about topic X. Participants in the task are provided with a corpus of blog posts and a set of questions of the type "What do bloggers think of X?" and are expected to provide a set of relevant blog posts in response. Starting in TREC 2007, the opinion-finding task includes a polarity

extension – i.e., find the positive/negative opinions about X.

The TREC opinion-finding task does not constitute fine-grained opinion summarization – it is coarse-grained (the level of granularity is a blog post, similar to a document) and there is little done in the way of summarization. Nevertheless, this task is regarded as the predecessor of the Text Analysis Conference (TAC) 2008 Opinion Question Answering (QA) and Summarization tasks, which have inspired the only other works in domain-independent fine-grained opinion summarization of which we are aware.

The 2008 Opinion QA and Summarization tasks use for evaluation the Blog06 test collection from TREC 2006 (Ounis et al., 2007), which contains about 3.2 million blog posts from about 100,000 different blogs. Both tasks arguably perform some kind of opinion summarization, so we describe each of them below.

The 2008 TAC Opinion Question Answering (QA) task requires answering a series of questions about opinions on a given topic. Two examples of such series of questions are shown in Figure 2.2[3]. Answers to the questions are either RIGID LISTS, i.e., a list of unique (disjoint) named entities, and SQUISHY LISTS or complex concepts, which can overlap, may be expressed in different ways and where boundaries of the concepts are not well defined. An example of a squishy list answer is shown in Figure 2.3. RIGID LISTS are evaluated on precision and recall of the system's extraction as compared to a gold-standard list of named entities. SQUISHY LISTS, in contrast, are evaluated by specifying a set of required information nuggets for each question and borrowing an evaluation measure from the field of summarization – the Pyramid F-score (Nenkova et al., 2007).

[3]All examples for the 2008 Opinion QA Task are taken from Hoa Trang Dang's presentation available at `http://www.nist.gov/tac/publications/2008/agenda.html`.

TARGET 1018: **Myth Busters**
1018.1 *RIGID LIST* Who likes Mythbuster's?
1018.2 *SQUISHY LIST* Why do people like Mythbuster's?
1018.3 *RIGID LIST* Who do people like on Mythbuster's?

TARGET 1047: **Trader Joes**
1047.1 *RIGID LIST* Who likes Trader Joe's?
1047.2 *SQUISHY LIST* Why do people like Trader Joe's?
1047.3 *RIGID LIST* Who doesn't like Trader Joe's?
1047.4 *SQUISHY LIST* Why don't people like Trader Joe's?

Figure 2.2: Two example question series from the 2008 TAC Opinion Question Answering task.

1047.2 SQUISHY LIST **Why do people like Trader Joes?**
BLOG06-3227 Trader Joes is your destination if you prefer Industrial wines (unlike Whole Foods).
BLOG06-2494 Everytime I walk into a Trader Joes it's a fun filled experience, and I always learn something new...
BLOG06-4400 Sure, we have our natural food stores, but they are expensive and don't have the variety that Trader Joe's has.
BLOG06-2494 Then I went to Trader Joe's and they have all the good stuff for cheap.

Figure 2.3: Answers to one of the example questions in the 2008 TAC Opinion Question Answering task.

The TAC 2008 Opinion Summarization task is a natural extension of the Opinion QA task for questions with answers of the SQUISHY LIST type. Systems are provided with a target such as "Trader Joe's" and 1 or 2 SQUISHY LIST questions. In response, systems are expected to produce one fluent summary per target that summarizes the answers to all the questions for the target. Summaries are scored for their content using the aforementioned pyramid score and manually scored along five dimensions: grammaticality, non-redundancy,

structure/coherence, overall readability and overall responsiveness (content + readability). In other words, the TAC 2008 Opinion Summarization task assigns importance both to the content of the opinion summary as well as its fluency and readability.

The TAC 2008 Opinion QA task on RIGID LIST questions is quite similar to standard QA tasks and, not surprisingly, participating systems have adopted standard QA approaches (Razmara and Kosseim, 2008; Li et al., 2008). In contrast, the QA task on SQUISHY LIST questions and the Opinion Summarization task, which share certain similarities, have been approached by participating systems through different techniques. Generally, systems attempt to identify relevant text segments (i.e. sentences or snippets of a given length) from the blogs, re-rank the set of relevant segments using some form of opinion classification and remove redundant text segments (Razmara and Kosseim, 2008; Li et al., 2008; Seki, 2008; Balahur et al., 2008). QA systems then output the resulting set of text segments, while summarization participants attempt to produce a fluent, readable summary.

The 2008 TAC Opinion QA and Opinion Summarization tasks bear certain resemblance to our work:

- Some of the RIGID LIST questions in the Opinion QA task require identifying sources of opinions on certain topic; we are interested in grouping together all opinions on the same topic.
- SQUISHY LIST questions require grouping together opinions on the same topic; we are also interested in grouping together opinions on the same topic.

However, our work differs along a number of dimensions from the 2008 TAC Opinion tasks:

- **Sources.** We are always grouping together opinions that belong to the same source, while TAC 2008 tasks do not always require that sources of opinions are identified.
- **Topics.** We are interested in grouping together opinions that are on the same topic, while the topics for the 2008 TAC Opinion tasks are pre-specified and typically involve a single named entity. Thus, TAC tasks can substitute topic extraction with relevance judgment while we can not.
- **Polarity.** TAC tasks do not always require polarity. They also do not require polarities of individual opinions to be aggregated.
- **Summary form.** We aim for an abstract, graph-based representation of opinions, while the TAC Opinion Summary task aims for a fluent natural language summary. The latter type of summaries are generally harder to produce, but TAC summaries require less in the way of understanding the expressed opinions. For example, a TAC-style summaries can be generated effectively without any need to determine who is the opinion holder or what is the polarity of the expressed opinions.

These differences make the problem of producing TAC 2008 Opinion Summaries fundamentally different from the opinion summarization problem discussed in this thesis. In fact, we regard the tasks of TAC Opinion Summarization and the fine-grained opinion summarization task that we propose as complementary to each other. It is easy to imagine how a summary in the format that we propose can be used gainfully by a system targeting the TAC Opinion tasks.

On the other hand, our summaries would benefit from a system that can transform the summaries into a fluent, human-readable text output for some applications. Finally, we are not aware of any system from the TAC 2008 evaluation that attempts to solve any of the problems discussed in this thesis.

2.3 Chapter Summary

In this chapter, we gave an overview of related work on opinion analysis. We started with a brief discussion of coarse-grained opinion analysis, describing some of the pioneering work in the area. We continued with an overview of fine-grained opinion analysis. We first described fine-grained opinion analysis work in the product review domain and then introduced work in domain-independent fine-grained opinion analysis, discussing definitions of opinion and opinion corpora, efforts in fine-grained opinion analysis and previous work on opinion summarization.

In the next chapter, we introduce the results of our empirical study that show that fine-grained opinion information is useful for a particular NLP application.

CHAPTER 3

USABILITY STUDY

The general-domain fine-grained opinion summaries that we propose in this thesis are only as usable as the fine-grained opinion information on which they are based. While several researchers have argued that fine-grained opinion information is indeed useful, these claims are supported by little empirical evidence. Therefore, before embarking on a substantial research effort, we deemed it important to empirically assess the usefulness of fine-grained opinion information. Toward this end, we pick one of the applications for which opinion information is arguably useful – Multi-Perspective Question Answering (MPQA) – and design a study to evaluate our hypothesis that fine-grained opinion information is both necessary and useful for this application. Our study is described in this Chapter, portions of which have appeared in Stoyanov et al. (2004) and Stoyanov et al. (2005).

3.1 Introduction

In recent years the field of NLP has made much progress in what we will refer to as *fact-based* question answering (QA), which is automatic, open-domain question answering (e.g., Voorhees (2002), Voorhees (2001), Voorhees and Buckland (2003)). Fact-based QA addresses questions such as:

- When did McDonald's open its first restaurant?
- Who was the first woman to successfully climb Mount Everest?
- What is the Kyoto Protocol?

On the other hand, relatively little research has been done in the area of Multi-Perspective Question Answering (MPQA), which targets questions of the following sort:

- How is Bush's decision not to ratify the Kyoto Protocol looked upon by Japan and other US allies?
- How do the Chinese regard the human rights record of the United States?
- What is Mugabe's opinion about the West's attitude and actions toward the 2002 Zimbabwe election?
- How did ordinary Venezuelans feel about the 2002 coup and subsequent events?

Due to the relative novelty of MPQA, there is little understanding of the properties of questions and answers in MPQA as compared to fact-based question answering (QA). Nevertheless, MPQA targets questions about opinions and, therefore, we hypothesise that successful approaches to MPQA would have to rely on opinion information. More precisely, using MPQA as a potential application that can benefit from relying on fine-grained opinion information, our usability study aims to:

1. Compare the properties of answers to opinion vs. fact questions to evaluate the hypothesis that there are significant differences between the two, deeming traditional QA techniques less effective for Multi-Perspective questions.
2. Evaluate the hypothesis that MPQA systems can be helped by fine-grained opinion information.

To address these issues we created the *OpQA* corpus of opinion questions and answers. Using the corpus, we compare and contrast the properties of fact and opinion questions and answers. We find that text spans identified as answers to opinion questions: (1) are approximately twice as long as those of fact questions, (2) are much more likely (37% vs. 9%) to represent *partial* answers rather than complete answers, (3) vary much more widely with respect to syntactic category – covering clauses, verb phrases, prepositional phrases, and noun phrases; in contrast, fact answers are overwhelming associated with noun phrases, and (4) are roughly half as likely to correspond to a single syntactic constituent type (16-38% vs. 31-53%).

Based on the disparate characteristics of opinion vs. fact answers, we argue that traditional fact-based QA approaches may have difficulty in an MPQA setting without modification. Instead, we propose that MPQA systems should rely on fine-grained opinion information. We use experiments to evaluate the usefulness of fine-grained opinion information in opinion question answering using the OpQA corpus. We find that filtering potential answers using machine learning and rule-based NLP opinion filters substantially improves the performance of an end-to-end MPQA system according to both a mean reciprocal rank (MRR) measure (0.59 vs. a baseline of 0.42) and a metric that determines the mean rank of the first correct answer (MRFA) (26.2 vs. a baseline of 61.3). Further, we find that requiring opinion answers to match the requested opinion source (e.g., does <source> approve of the Kyoto Protocol) dramatically improves the performance of the MPQA system on the hardest questions in the corpus.

In the remainder of this Chapter we describe the OpQA corpus (Section 3.2) and then use the OpQA corpus to identify potentially problematic issues for handling opinion vs. fact questions (Section 3.3). Sections 3.4 and 3.5 explore the use of opinion information in the design of MPQA systems.

3.2 OpQA Corpus

To support our research in MPQA, we created the OpQA corpus of opinion and fact questions and answers.

3.2.1 Documents and Questions

The OpQA corpus consists of 98 documents from the MPQA corpus. Each of the documents addresses one of four general topics: *kyoto*, concerning President Bush's alternative to the Kyoto protocol; *mugabe*, concerning 2002 elections in Zimbabwe and Mugabe's reelection; *humanrights*, discussing the US annual human rights report; and *venezuela*, which describes the 2002 coup d'etat in Venezuela. The documents were automatically selected from a bigger set of over 270,000 documents as being relevant to one of the four topics using the SMART information retrieval system. The OpQA corpus contains between 19 and 33 documents for each topic.

Fact and opinion questions for each topic were added to the OpQA corpus by a volunteer not associated with the research project. He was given two randomly selected documents on each topic along with a set of instructions for creating fact vs. opinion questions, which are shown in Appendix A. The complete set of 30 questions is shown in Table 3.2.1. The set contains an equal number of

opinion (o) and fact (f) questions for each topic.

Once the documents and questions were obtained, answers for the questions in the supporting documents had to be identified: we manually added *answer* annotations for every text segment in the OpQA corpus that constituted an answer to any question. The *answer* annotations include attributes to indicate the **topic** of the associated question, the **question number** within that topic, and the annotator's **confidence** that the segment actually answered the question. Documents were annotated by two separate annotators, each of which annotated the questions associated with two separate topics. Annotators did not have access to the fine-grained opinion annotations during answer annotation. Instructions for adding answer annotations that were used by the annotators are enclosed in Appendix B.

3.2.2 Difficulties in Corpus Creation

This section summarizes some of the difficulties encountered during creation of the OpQA corpus.

Question Creation.

Despite that the question creation instructions instructed against it, it appears that some questions were reverse-engineered from the available documents. These questions are answered in only one or two of the documents, which presents some challenges when using the collection for evaluation. Nevertheless, the setting is not unrealistic since the situation in which questions find support in only a few documents is often present in real-world QA systems.

Additionally, the classification associated with each question — fact or opin-

Table 3.1: Questions in the OpQA collection by topic.

	Kyoto
1 f	What is the Kyoto Protocol about?
2 f	When was the Kyoto Protocol adopted?
3 f	Who is the president of the Kiko Network?
4 f	What is the Kiko Network?
5 o	Does the president of the Kiko Network approve of the US action concerning the Kyoto Protocol?
6 o	Are the Japanese unanimous in their opinion of Bush's position on the Kyoto Protocol?
7 o	How is Bush's decision not to ratify the Kyoto Protocol looked upon by Japan and other US allies?
8 o	How do European Union countries feel about the US opposition to the Kyoto protocol?
	Human Rights
1 f	What is the murder rate in the United States?
2 f	What country issues an annual report on human rights in the United States?
3 o	How do the Chinese regard the human rights record of the United States?
4 f	Who is Andrew Welsdan?
5 o	What factors influence the way in which the US regards the human rights records of other nations?
6 o	Is the US Annual Human Rights Report received with universal approval around the world?
	Venezuela
1 f	When did Hugo Chavez become President?
2 f	Did any prominent Americans plan to visit Venezuela immediately following the 2002 coup?
3 o	Did anything surprising happen when Hugo Chavez regained power in Venezuela after he was removed by a coup?
4 o	Did most Venezuelans support the 2002 coup?
5 f	Which governmental institutions in Venezuela were dissolved by the leaders of the 2002 coup?
6 o	How did ordinary Venezuelans feel about the 2002 coup and subsequent events?
7 o	Did America support the Venezuelan foreign policy followed by Chavez?
8 f	Who is Vice-President of Venezuela?
	Mugabe
1 o	What was the American and British reaction to the reelection of Mugabe?
2 f	Where did Mugabe vote in the 2002 presidential election?
3 f	At which primary school had Mugabe been expected to vote in the 2002 presidential election?
4 f	How long has Mugabe headed his country?
5 f	Who was expecting Mugabe at Mhofu School for the 2002 election?
6 o	What is the basis for the European Union and US critical attitude and adversarial action toward Mugabe?
7 o	What did South Africa want Mugabe to do after the 2002 election?
8 o	What is Mugabe's opinion about the West's attitude and actions toward the 2002 Zimbabwe election?

ion — did not always seem appropriate. For instance, the following opinion question "What is the basis for the European Union and US critical attitude and adversarial action toward Mugabe?" could arguably be classified as fact-based, since the question is not actually asking about European union and US's opinion, but rather about the basis for it. Similarly, the factual question "Did any prominent Americans plan to visit Venezuela soon immediately following the 2002 coup?" could be judged as asking about the opinion of prominent Americans.

Annotating Answers.

The most frequently encountered problem in answer annotation is a well-known problem from fact-based QA, namely the difficulty of deciding what constitutes an answer to a question. The problem was further amplified by the presence of opinion questions. For instance, the question "Did most Venezuelans support the 2002 coup?" had potential answers such as "Protesters...failed to gain the support of the army" and "... thousand of citizens rallied the streets in support of Chavez." Both segments hint that most Venezuelans did not support the coup that forced Chavez to resign. Both passages, however, state it in a very indirect way. It is hard even for humans to conclude whether the above two passages constitute answers to the question.

A related issue is that opinionated documents often express answers to the questions only very indirectly, by using word selection and style of language (*expressive subjectivity*). While we can annotate such expressions as at least contributing to the answer to a question, our current answer annotation structure has no means for indicating how to map that expression onto an actual answer.

An additional problem is that opinion questions often ask about opinions of certain entities, such as countries, governments, and popular opinions. During the annotation phase, the difficulty of recognizing opinions of such collective entities became clear. It was hard for human annotators to judge what can be considered an expression of the opinion of collective entities and often the conjecture required a significant amount of background information.

3.3 Characteristics of opinion answers

Next, we use the OpQA corpus to analyze and compare the characteristics of fact vs. opinion questions. Based on our findings, we believe that QA systems based solely on traditional QA techniques are likely to be less effective at MPQA than they are at traditional fact-based QA.

3.3.1 Traditional QA architectures

Despite the wide variety of approaches implied by modern QA systems, almost all systems rely on the following two steps (subsystems), which have empirically proven to be effective:

- **IR module.** The QA system invokes an IR subsystem that employs traditional text similarity measures (e.g., tf/idf-weighted cosine similarity) to retrieve and rank document fragments (sentences or paragraphs) with respect to the question (query).
- **Linguistic filters.** QA systems employ a set of filters and text processing components to discard some document fragments. The following filters have empirically proven to be effective and are used universally:

Semantic filters prefer an answer segment that matches the semantic class(es) associated with the question type (e.g., *date* or *time* for *when* questions; *person* or *organization* for *who* questions).

Syntactic filters are also based on the type of question. The most common and effective syntactic filters select a specific constituent (e.g., noun phrase) according to the question type (e.g., *who* question).

QA systems typically interleave the above two subsystems with a variety of different steps processing both the question and the answer. The goal of the processing is to identify text fragments that contain an answer to the question. Typical QA systems do not perform any further text processing; they return the text fragment as it occurred in the text.[1]

3.3.2 Corpus-based analysis of opinion answers

We hypothesize that QA systems that conform to this traditional architecture will have difficulty handling opinion questions without non-trivial modifications. In support of this hypothesis, we provide statistics from the OpQA corpus to illustrate some of the characteristics that distinguish answers to opinion vs. fact questions, and discuss their implications for a traditional QA system architecture.

Answer length. We see in Table 3.2 that the average length of opinion answers in the OpQA corpus is 9.24 tokens, almost double that of fact answers. Unfortunately, longer answers could present problems for some traditional QA

[1]This architecture is seen mainly in QA systems designed for TREC's "factoid" and "list" QA tracks. Systems competing in the relatively new "definition" or "other" tracks have begun to introduce new approaches. However, most such systems still rely on the IR step and return the text fragment as it occurred in the text.

Table 3.2: Number of answers, average answer length (in tokens), and number of partial answers for fact/opinion questions.

	Number of answers	Length	Number of partials
fact	124	5.12	12 (9.68%)
opinion	415	9.24	154 (37.11%)

systems. In particular, some of the more sophisticated algorithms that perform **additional processing** steps such as logical verifiers (Moldovan et al., 2002) may be less accurate or computationally infeasible for longer answers. More importantly, longer answers are likely to span more than a single syntactic constituent, rendering the syntactic filters, and very likely the semantic filters, less effective.

Partial answers. Table 3.2 also shows that over 37% of the opinion answers were marked as partial vs. 9.68% of the fact answers. The implications of partial answers for the traditional QA architecture are substantial: an MPQA system will require an **answer generator** to (1) distinguish between partial and full answers; (2) recognize redundant partial answers; (3) identify which subset of the partial answers, if any, constitutes a full answer; (4) determine whether additional documents need to be examined to find a complete answer; and (5) assemble the final answer from partial pieces of information.

Syntactic constituent of the answer. As discussed in Section 3.3.1, traditional QA systems rely heavily on the predicted syntactic and semantic class of the answer. Based on answer lengths, we speculated that opinion answers are unlikely to span a single constituent and/or semantic class. This speculation is confirmed by examining the phrase type associated with OpQA answers using Abney's (1996) CASS partial parser.[2] For each question, we count the number of

[2]The parser is available from `http://www.vinartus.net/spa/`.

times an answer segment for the question (in the manual annotations) matches each constituent type. We consider four constituent types – noun phrase (n), verb phrase (v), prepositional phrase (p), and clause (c) – and three matching criteria:

1. The **ex**act match criterion is satisfied only by answer segments whose spans exactly correspond to a constituent in the CASS output.
2. The **up** criterion considers an answer to match a CASS constituent if the constituent completely contains the answer and no more than three additional (non-answer) tokens.
3. The **up/dn** criterion considers an answer to match a CASS constituent if it matches according to the **up** criterion or if the answer completely contains the constituent and no more than three additional tokens.

The counts for the analysis of answer segment syntactic type for fact vs. opinion questions are summarized in Table 3.3.2. Results for the 15 fact questions are shown in the left half of the table, and for the 15 opinion questions in the right half. The leftmost column in each half provides the question topic and number, and the second column indicates the total number of answer segments annotated for the question. The next three columns show, for each of the **ex**, **up**, and **up/dn** matching criteria, respectively, the number of annotated answer segments that match the majority syntactic type among answer segments for that question/criterion pair. Using a traditional QA architecture, the MPQA system might filter answers based on this majority type. The *syn type* column indicates the majority syntactic type using the exact match criterion; two values in the column indicate a tie for majority syntactic type, and an empty syntactic type indicates that no answer exactly matched any of the four constituent

Table 3.3: Syntactic Constituent Type for Answers in the OpQA Corpus

Fact						Opinion					
Question	# of answers	Matching Criteria ex	up	up/dn	syn type	Question	# of answers	Matching Criteria ex	up	up/dn	syn type
H 1	1	0	0	0		H 3	15	5	5	5	c
H 2	4	2	2	2	n	H 5	24	5	5	10	n
H 4	1	0	0	0		H 6	123	17	23	52	n
K 1	48	13	14	24	n	K 5	3	0	0	1	
K 2	38	13	13	19	n	K 6	34	6	5	12	c
K 3	1	1	1	1	c n	K 7	55	9	8	19	c
K 4	2	1	1	1	n	K 8	25	4	4	10	v
M 2	3	0	0	1		M 1	74	10	12	29	v
M 3	1	0	0	1		M 6	12	3	5	7	n
M 4	10	2	2	5	n	M 7	1	0	0	0	
M 5	3	1	1	2	c	M 8	3	0	0	1	
V 1	4	3	3	4	n	V 3	1	1	0	1	c
V 2	1	1	1	1	n	V 4	13	2	2	2	c
V 5	3	0	1	1		V 6	9	2	2	5	c n
V 8	4	2	4	4	n	V 7	23	3	1	5	
Coverage	124	39 31%	43 35%	66 53%		Coverage	415	67 16%	70 17%	159 38%	

types. With only a few exceptions, the **up** and **up/dn** matching criteria agreed in majority syntactic type.

Results in Table 3.3.2 show a significant disparity between fact and opinion questions. For fact questions, the syntactic type filter would keep 31%, 35%, or 53% of the correct answers, depending on the matching criterion. For opinion questions, there is unfortunately a two-fold reduction in the percentage of correct answers that would remain after filtering — only 16%, 17% or 38%, depending on the matching criterion. More importantly, the majority syntactic type among answers for fact questions is almost always a noun phrase, while no single constituent type emerges as a useful syntactic filter for opinion questions (see the **syn phrase** columns in Table 3.3.2). Finally, because semantic class information is generally tied to a particular syntactic category, the effectiveness

of traditional semantic filters in the MPQA setting is unclear.

In summary, identifying answers to questions in an MPQA setting within a traditional QA architecture will be difficult. First, the implicit and explicit assumptions inherent in standard linguistic filters are consistent with the characteristics of fact, rather than opinion-oriented QA. In addition, the presence of relatively long answers and partial answers will require a much more complex **answer generator** than is typically present in current QA systems.

In Sections 3.4 and 3.5, we evaluate the hypothesis that fine-grained opinion information may be used successfully in systems for MPQA. In particular, we propose and evaluate two types of **opinion filters** for MPQA: **subjectivity filters** and **opinion source filters**. Both types of linguistic filters rely on fine-grained opinion information, which has been manually annotated in our corpus. Documents in our OpQA corpus come from the larger MPQA corpus, which, as discussed in Chapter 2, contains manual opinion annotations. The annotation framework is described in more detail in Wiebe et al. (2005b) and in Chapter 2. As a brief reminder, the MPQA corpus contains annotations for expression-level opinions with several attributes, including the source of the opinion.

3.4 Subjectivity Filters for MPQA Systems

This section describes three **subjectivity filters** based on fine-grained opinion information. Below (in Section 3.4.3), the filters are used to remove fact sen-

tences from consideration when answering opinion questions, and the OpQA corpus is used to evaluate their effectiveness.

3.4.1 Manual Subjectivity Filter

Much previous research on automatic extraction of opinion information performs classifications at the sentence level. Therefore, we define sentence-level opinion classifications in terms of the phrase-level annotations. For our gold standard of manual opinion classifications (dubbed MANUAL for the rest of the paper) we will follow Riloff and Wiebe's (2003) convention (also used by Wiebe and Riloff (2005)) and consider a sentence to be *opinion* if it contains at least one opinion of intensity *medium* or higher, and to be *fact* otherwise.

3.4.2 Two Automatic Subjectivity Filters

Several research efforts have attempted to perform automatic opinion classification on the clause and sentence level (see Chapter 2). We investigate whether such information can be useful for MPQA by using the automatic sentence level opinion classifiers of Riloff and Wiebe (2003) and Wiebe and Riloff (2005).

As discussed in Chapter 2, Riloff and Wiebe (2003) use a bootstrapping algorithm to perform a sentence-based opinion classification on the MPQA corpus. They use a set of high precision subjectivity and objectivity clues to identify subjective and objective sentences. This data is then used in an algorithm similar to AutoSlog-TS (Riloff, 1996) to automatically identify a set of extraction patterns. The acquired patterns are used iteratively to identify a larger set of subjective and objective sentences. In our experiments we use the classifier that was created by the reimplementation of this bootstrapping process in Wiebe and Riloff

Table 3.4: Precision, recall, and F-measure for the two classifiers.

		precision	recall	F
MPQA corpus	RULEBASED	90.4	34.2	46.6
	NAIVE BAYES	79.4	70.6	74.7

(2005). We will use RULEBASED to denote the opinion information output by this classifier.

In addition, Wiebe and Riloff used the RULEBASED classifier to produce a labeled data set for training. They trained a Naive Bayes subjectivity classifier on the labeled set. We will use NAIVE BAYES to refer to Wiebe and Riloff's naive Bayes classifier.[3] Table 3.4.2 shows the performance of the two classifiers on the MPQA corpus as reported by Wiebe and Riloff.

3.4.3 Experiments

We performed two types of experiments using subjectivity filters.

3.4.4 Answer rank experiments

Our hypothesis motivating the first type of experiment is that subjectivity filters can improve the answer identification phase of an MPQA system. We implement the IR subsystem of a traditional QA system, and apply subjectivity filters to the IR results. Specifically, for each opinion question in the corpus [4], we do

[3] Specifically, the one they label *Naive Bayes 1*.

[4] We do not evaluate the subjectivity filters on the 15 fact questions. Since opinion sentences are defined as containing at least one opinion of intensity medium or higher, opinion sentences can contain factual information and sentence-level opinion filters are not likely to be effective for fact-based QA.

the following:

1. Split all documents in our corpus into sentences.
2. Run an information retrieval algorithm[5] on the set of all sentences using the question as the query to obtain a *ranked list* of sentences.
3. Apply a subjectivity filter to the *ranked list* to remove all fact sentences from the *ranked list*.

We test each of the MANUAL, RULEBASED, and NAIVE BAYES subjectivity filters. We compare the rank of the first answer to each question in the *ranked list* before the filter is applied, with the rank of the first answer to the question in the *ranked list* after the filter is applied.

Results

Results for the opinion filters are compared to a simple baseline, which performs the information retrieval step with no filtering. Table 3.5 gives the results on the 15 opinion questions for the baseline and each of the three *subjectivity filters*. The table shows two cumulative measures – the mean reciprocal rank (MRR) across the top five answers in the *ranked list*[6] and the mean rank of the first answer (MRFA).[7]

[5]We use the Lemur toolkit's standard tf.idf implementation available from `http://www.lemurproject.org/`.

[6]The MRR is computed as the average of $1/r$, where r is the rank of the first answer.

[7]MRR has been accepted as the standard performance measure in QA, since MRFA can be strongly affected by outlier questions. However, the MRR score is dominated by the results in the high end of the ranking. Thus, MRFA may be more appropriate for our experiments because the filters are an intermediate step in the processing, the results of which other MPQA components may improve.

Table 3.5: Results for the subjectivity filters.

Topic	Qnum	Baseline	Manual	NaiveBayes	Rulebased
Kyoto	5	1	1	1	1
	6	5	4	4	3
	7	1	1	1	1
	8	1	1	1	1
Human Rights	3	1	1	1	1
	5	10	6	7	5
	6	1	1	1	1
Venezuela	3	106	81	92	35
	4	3	2	3	1
	6	1	1	1	1
	7	3	3	3	2
Mugabe	1	2	2	2	2
	6	7	5	5	4
	7	447	291	317	153
	8	331	205	217	182
MRR :		0.4911	0.5189	0.5078	0.5856
MRFA:		61.3333	40.3333	43.7333	26.2

Table 3.5 shows that all three *subjectivity filters* outperform the baseline: for all three filters, the first answer in the filtered results for all 15 questions is ranked at least as high as in the baseline. As a result, the three subjectivity filters outperform the baseline in both MRR and MRFA. Surprisingly, the best performing subjectivity filter is RULEBASED, surpassing the gold standard MANUAL, both in MRR (0.59 vs. 0.52) and MRFA (40.3 vs. 26.2). Presumably, the improvement in performance comes from the fact that RULEBASED identifies subjective sentences with the highest precision (and lowest recall). Thus, the RULEBASED subjectivity filter discards non-subjective sentences most aggressively.

Table 3.6: Answer probability results.

			sentence	
			fact	opinion
question	Manual	fact	56 (46.67%)	64 (53.33%)
		opinion	42 (10.14%)	372 (89.86%)
	Naive Bayes	fact	49 (40.83%)	71 (59.17%)
		opinion	57 (13.77%)	357 (86.23%)
	Rulebased	fact	96 (80.00%)	24 (20.00%)
		opinion	184 (44.44%)	230 (55.56%)

3.4.5 Answer probability experiments

The second experiment, *answer probability*, begins to explore whether opinion information can be used in an **answer generator**. This experiment considers correspondences between (1) the classes (i.e., opinion or fact) assigned by the subjectivity filters to the sentences containing answers, and (2) the classes of the questions the answers are responses to (according to the OpQA annotations). That is, we compute the probabilities (where *ans* = answer):

P(*ans* is in a *C1* sentence | *ans* is the answer to a *C2* question) for all four combinations of *C1*=opinion, fact and *C2*=opinion, fact.

Results

Results for the answer probability experiment are given in Table 3.6. The rows correspond to the classes of the questions to which the answers responds, and the columns correspond to the classes assigned by the subjectivity filters to the sentences containing the answers. The first two rows, for instance, give the results for the MANUAL criterion. MANUAL placed 56 of the answers to fact ques-

tions in fact sentences (46.67% of all answers to fact questions) and 64 (53.33%) of the answers to fact questions in opinion sentences. Similarly, MANUAL placed 42 (10.14%) of the answers to opinion questions in fact sentences, and 372 (89.86%) of the answers to opinion questions in opinion sentences.

The answer probability experiment sheds some light on the subjectivity filter experiments. All three subjectivity filters place a larger percentage of answers to opinion questions in opinion sentences than they place in fact sentences. However, the different filters exhibit different degrees of discrimination. Answers to opinion questions are almost always placed in opinion sentences by MANUAL (89.86%) and NAIVE BAYES (86.23%). While that aspect of their performance is excellent, MANUAL and NAIVE BAYES place more answers to fact questions in opinion rather than fact sentences (though the percentages are in the 50s). This is to be expected, because MANUAL and NAIVE BAYES are more conservative and err on the side of classifying sentences as opinions: for MANUAL, the presence of any subjective expression makes the entire sentence opinion, even if parts of the sentence are factual; NAIVE BAYES shows high recall but lower precision in recognizing opinion sentences (see Table 3.4.2). Conversely, RULEBASED places 80% of the fact answers in fact sentences and only 56% of the opinion answers in opinion sentences. Again, the lower number of assignments to opinion sentences is to be expected, given the high precision and low recall of the classifier. But the net result is that, for RULEBASED, the off-diagonals are all less than 50%: it places more answers to fact questions in fact rather than opinion sentences (80%), and more answers to opinion questions in opinion rather than fact sentences (56%). This is consistent with its superior performance in the subjectivity filtering experiment.

In addition to explaining the performance of the subjectivity filters, the answer rank experiment shows that the automatic opinion classifiers can be used directly in an **answer generator** module – the two automatic classifiers rely on evidence in the sentence to predict the class (the information extraction patterns used by RULEBASED and the features used by NAIVE BAYES).

3.5 Opinion Source Filters for MPQA Systems

In addition to subjectivity filters, we also define an opinion *source filter* based on the manual opinion annotations. This filter removes all sentences that do not have an opinion annotation with a source that matches the source of the question[8]. For this filter we only used the MANUAL source annotations. We employ the same Answer Rank experiment as in 3.4.4, substituting the source filter for a subjectivity filter.

Results. Results for the source filter are mixed. The filter outperforms the baseline on some questions and performs worst on others. As a result the MRR for the source filter is worse than the baseline (0.4633 vs. 0.4911). However, the source filter exhibits by far the best results using the MRFA measure, a value of 11.267. The performance improvement is due to the filter's ability to recognize the answers to the hardest questions, for which the other filters have the most trouble (questions *mugabe* 7 and 8). For these questions, the rank of the first answer improves from 153 to 21, and from 182 to 11, respectively. With the exception of question *venezuela* 3, which does not contain a clear source (and is problematic altogether because there is only a single answer in the corpus and

[8]We manually identified the sources of each of the 15 opinion questions.

the question's qualification as opinion is not clear) the *source filter* always ranked an answer within the first 25 answers. Thus, *source filters* can be especially useful in systems that rely on the presence of an answer within the first few ranked answer segments and then invoke more sophisticated analysis in the **additional processing** phase.

3.6 Chapter summary

In this Chapter we discussed issues concerning the usability of fine-grained opinion information for NLP applications. We used a particular application, multi-perspective question answering (MPQA). We began by describing the OpQA corpus – a corpus of fact- and opinion-based questions and their manually annotated answers – created for the purpose of evaluation. Using the corpus, we compared the characteristics of answers to fact and opinion questions. Based on the different characteristics, we surmise that traditional QA approaches may not be as effective for MPQA as they have been for fact-based QA. Finally, we showed that fine-grained opinion information can be successfully used in an MPQA system. In summary, empirical evidence confirmed our hypothesis that (1) traditional QA approaches are unlikely to be successful for MPQA and, (2), fine-grained opinion information (even when extracted automatically) can be used successfully by MPQA systems. Thus, we have shown that fine-grained opinon information is useful for at least one NLP application.

CHAPTER 4

SOURCE COREFERENCE RESOLUTION

In this chapter, we address the problem of *source coreference resolution* — the task of determining which mentions of opinion sources refer to the same entity. Parts of this chapter are published in Stoyanov and Cardie (2006a) and Stoyanov and Cardie (2006b).

As argued in Chapter 1, source coreference resolution constitutes an integral step in the process of generating full opinion summaries. For the example of Figure 1.1 (re-printed in this chapter as Figure 4.1), the task of source coreference resolution includes recognizing that source mentions "Zacarias Moussaoui", "he", "my", and "Mr. Moussaoui" all refer to the same person; and that source mentions "Mr. Zerkin" and "Zerkin" refer to the same person.

At first glance, source coreference resolution appears equivalent to the task of noun phrase coreference resolution (discussed in Section 4.1.1) and therefore amenable to traditional coreference resolution techniques (e.g. Ng and Cardie (2002), Morton (2000)). We hypothesize in Section 4.2, however, that the task is likely to be subject of a better solution by treating it in the context of a new machine learning setting that we refer to as *partially supervised clustering*. In particular, due to high coreference annotation costs, data sets that are annotated with opinion information (like the MPQA corpus) do not typically include supervisory coreference information for *all* noun phrases in a document (as would be required for the application of traditional coreference resolution techniques), but only for noun phrases that act as opinion sources (or targets).

[Source Zacarias Moussaoui] [– *complained*] at length today about [Target his own lawyer], telling a federal court jury that [Target he] was [– *more interested in achieving fame than saving Moussaoui's life*].

Mr. Moussaoui said he was appearing on the witness stand to tell the truth. And one part of the truth, [Source he] said, is that [Target sending him to prison for life] would be "[– *a greater punishment*] than being sentenced to death."

"[– [Target *You*] *have put your interest ahead of* [Source *my*] *life*]," [Source Mr. Moussaoui] told his court-appointed lawyer Gerald T. Zerkin.

...

But, [Source Mr. Zerkin] pressed [Target Mr. Moussaoui], was it [– *not true*] that he told his lawyers earlier not to involve any Muslims in the defense, not to present any evidence that might persuade the jurors to spare his life?

...

[Source Zerkin] seemed to be trying to show the jurors that while [Target the defendant] is generally [+ *an honest individual*], his conduct shows [Target he] is [– *not stable mentally*], and thus [– *undeserving*] of [Target the ultimate punishment].

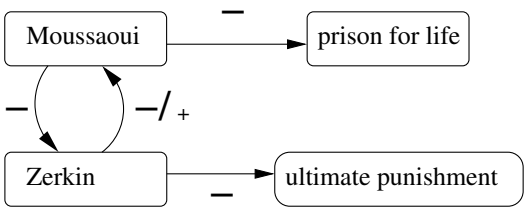

Figure 4.1: (Re-print of Figure 1.1) Example text containing opinions (above) and a summary of the opinions (below).

As a result, we define the task of *partially supervised clustering*, the goal of which is to learn a clustering function from a set of partially specified clustering examples (Section 4.4). We are not aware of prior work on the problem of partially supervised clustering and argue that it differs substantially from that of semi-supervised clustering. We propose an algorithm for partially supervised clustering that extends a rule learner with structure information and is generally applicable to problems that fit the partially supervised clustering definition

(Section 4.5). We apply the algorithm to the source coreference resolution task and evaluate its performance on a the MPQA corpus, which includes source coreference chains (Section 4.6). We find that our algorithm outperforms highly competitive baselines by a considerable margin – B^3 score of 83.2 vs. 81.8 and 67.1 vs. 60.9 F1 score for the identification of positive source coreference links.

4.1 Related Work

In addition to the work in sentiment analysis discussed in Chapter 2, there are two other areas of research relevant to the problem of source coreference resolution – traditional noun phrase coreference resolution and supervised and weakly supervised clustering. Related work in the former area is summarized briefly below. Supervised and weakly supervised clustering approaches are discussed in Subsection 4.4.

4.1.1 Coreference resolution.

Coreference resolution is a relatively well studied NLP problem (e.g. Morton (2000), Ng and Cardie (2002), Iida et al. (2003), McCallum and Wellner (2003)). Coreference resolution is defined as the problem of deciding which noun phrases in the text (*mentions*) refer to the same real world entities (*are coreferent*). Generally, successful approaches to coreference resolution have relied on supervised classification followed by clustering. For supervised classification these approaches learn a pairwise function to predict whether a pair of noun phrases is coreferent. Subsequently, when making coreference resolution decisions on unseen documents, the learnt pairwise NP coreference classifier is run, followed by a clustering step to produce the final clusters (coreference chains)

of coreferent NPs. For both training and testing, coreference resolution algorithms rely on feature vectors for pairs of noun phrases that encode linguistic information about the NPs and their local context. Our general approach to source coreference resolution is inspired by the state-of-the-art performance of one such approach to coreference resolution, which relies on a rule learner and single-link clustering as described in Ng and Cardie (2002).

4.2 Problem Definition

In this section we introduce the problem of source coreference resolution in the context of opinion summarization and argue for the need for novel methods for the task.

The task of *source coreference resolution* is to decide which mentions of opinion sources refer to the same entity. Much like traditional coreference resolution, we employ a learning approach; however, our approach differs from traditional coreference resolution in its definition of the learning task. Motivated by the desire to utilize unlabeled examples (discussed later), we define training as an integrated task in which pairwise NP coreference decisions are learned together with the clustering function as opposed to treating each NP pair as a training example. Thus, our training phase takes as input a set of documents with manually annotated opinion sources together with coreference annotations for the sources; it outputs a classifier that can produce source coreference chains for previously unseen documents containing marked (manually or automatically) opinion sources. More specifically, the source coreference resolution training phase proceeds through the following steps:

1. **Source-to-NP mapping:** We preprocess each document by running a tokenizer, sentence splitter, POS tagger, parser, and an NP finder. Subsequently, we augment the set of NPs found by the NP finder with the help of a system for named entity detection. We then map the sources of opinions to the automatically extracted NPs using a set of heuristics.

2. **Feature vector creation:** We extract a feature vector for every pair of NPs from the preprocessed corpus. We use the features introduced by Ng and Cardie (2002) for the task of coreference resolution.

3. **Classifier construction:** Using the feature vectors from step 2, we construct a training set containing one training example per document. Each training example consists of the feature vectors for all pairs of NPs in the document, including those that do not map to sources, together with the available coreference information for the *source noun phrases* (i.e. the noun phrases to which sources are mapped). The training instances are provided as input to a learning algorithm (see Section 4.5), which constructs a classifier that can take the instances associated with a new (previously unseen) document and produce a clustering over all NPs in the document.

The testing phase employs steps 1 and 2 as described above, but replaces step 3 by a straightforward application of the learnt classifier. Since we are interested in coreference information only for the source NPs, we simply discard the non-source NPs from the resulting clustering.

The approach to source coreference resolution described here would be identical to traditional coreference resolution when provided with training examples containing coreference information for all NPs. However, opinion corpora in general, and our corpus in particular, contain no coreference information about

general NPs. Nevertheless, after manual sources are mapped to NPs in step 1 above, our approach can rely on the available coreference information for the source NPs. Due to the high cost of coreference annotation, we desire methods that can work in the presence of only this limited amount of coreference information.

A possible workaround for the absence of full NP coreference information is to train a traditional coreference system only on the labeled part of the data (indeed that is one of the baselines against which we compare). However, we believe that an effective approach to source coreference resolution has to utilize the unlabeled noun phrases because links between sources might be realized through non-source mentions. This problem is illustrated in Figure 4.1. The underlined *Moussaoui* is coreferent with all of the Moussaoui references marked as sources, but, because it is used in an objective sentence rather than as the source of an opinion, the reference would be omitted from the *Moussaoui* source chain. Unfortunately, this proper noun phrase might be critical in establishing the coreference of the final source reference *he* with the other mentions of the source *Moussaoui*.

As mentioned previously, in order to utilize the unlabeled data, our approach differs from traditional coreference resolution, which uses NP pairs as training instances. We instead follow the framework of supervised clustering (Finley and Joachims, 2005; Li and Roth, 2005) and consider each document as a training example. As in supervised clustering, this framework has the additional advantage that the learning algorithm can consider the clustering algorithm when making decisions about pairwise classification, which could lead to improvements in the classifier. We devote the next section to step 1 above, de-

Table 4.1: Statistics for matching sources to noun phrases.

	Single Match	Multiple Matches	No Match
Total	7811	3461	50
Exact	6242	1303	0

scribing the difficulties associated with mapping sources to NPs and the set of heuristics that we employ to perform the mapping. We follow that by describing our approach to classifier construction for step 3 and compare our problem to traditional weakly supervised clustering, characterizing it as an instance of the novel problem of partially supervised clustering.

4.3 Mapping sources to noun phrases

This section describes our method for heuristically mapping sources to NPs. In the context of source coreference resolution we consider a noun phrase to correspond to (or match) a source if the source and the NP cover the exact same span of text. Unfortunately, the annotated sources did not always match exactly a single automatically extracted NP. We discovered the following problems:

1. **Inexact span match.** We discovered that often (3777 out of the 11322 source mentions in the MPQA corpus) there is no noun phrase whose span matches exactly the source although there are noun phrases that overlap the source. In most cases this is due to the way spans of sources are marked in the data. For instance, in some cases determiners are not included in the source span (e.g. "*Venezuelan people*" vs. "*the Venezuelan people*"). In other cases, differences are due to mistakes by the NP extractor (e.g. "*Muslim rulers*" was not recognized, while "*Muslim*" and "*rulers*"

were recognized). Yet in other cases, manually marked sources do not match the definition of a noun phrase. This case is described in more detail next.

2. **Multiple NP match.** For 3461 of the 11322 source mentions more than one NP overlaps the source. In roughly a quarter of these cases the multiple match is due to the presence of nested NPs (introduced by the NP augmentation process introduced in Section 4.2). In other cases the multiple match is caused by source annotations that spanned multiple NPs or included more than only NPs inside its span. There are three general classes of such sources. First, some of the marked sources are appositives such as "*the country's new president, Eduardo Duhalde*". Second, some sources contain an NP followed by an attached prepositional phrase such as "*Latin American leaders at a summit meeting in Costa Rica*". Third, some sources are conjunctions of NPs such as "*Britain, Canada and Australia*". Treatment of the latter is still a controversial problem in the context of coreference resolution as it is unclear whether conjunctions represent entities that are distinct from the conjuncts. For the purpose of our current work we do not attempt to address conjunctions.

3. **No matching NP.** Finally, for 50 of the 11322 sources there are no overlapping NPs. Half of those (25 to be exact) included marking of the relative pronoun "*who*" such as in the sentence "*Carmona named new ministers, including two military officers* **who** *rebelled against Chavez*". From the other 25, 19 included markings of non-NPs including question words, qualifiers, and adjectives such as "*many*", "*which*", and "*domestically*". The remaining six are rare NPs such as "*lash*" and "*taskforce*" that are mistakenly not recognized by the NP extractor.

Counts for the different types of matches of sources to NPs are shown in Table 4.1. We determine the match in the problematic cases using a set of heuristics:

1. If a source matches any NP exactly in span, match that source to the NP; do this even if multiple NPs overlap the source – we are dealing with nested NP's.
2. If no NP matches matches exactly in span then:
    - If a single NP overlaps the source, then map the source to that NP. Most likely we are dealing with differently marked spans.
    - If multiple NPs overlap the source, determine whether the set of overlapping NPs include any non-nested NPs. If all overlapping NPs are nested with each other, select the NP that is closer in span to the source – we are still dealing with differently marked spans, but now we also have nested NPs. If there is more than one set of nested NPs, then most likely the source spans more than a single NP. In this case we select the outermost of the last set of nested NPs before any preposition in the span. We prefer: the outermost NP because longer NPs contain more information; the last NP because it is likely to be the head NP of a phrase (this also handles the case of explanation followed by a proper noun); NP's before preposition, because a preposition signals an explanatory prepositional phrase.
3. If no NP overlaps the source, select the last NP before the source. In half of the cases we are dealing with the word *who*, which typically refers to the last preceding NP.

Following mapping opinion sources to NPs and feature vector creation (steps 1 and 2), we aim to learn a classifier that can predict correctly the clusters of coreferent sources (step 2). We discuss our approach to learning such a classifier through the novel definition of the problem as partially supervised clustering next.

4.4 Partially Supervised Clustering

In our desire to perform effective source coreference resolution we arrive at the following learning problem – the learning algorithm is presented with a set of partially specified examples of clusterings and acquires a function that can cluster accurately an unseen set of items, while taking advantage of the unlabeled information in the examples.

This setting is to be contrasted with semi-supervised clustering (or clustering with constraints), which has received much research attention (e.g. Demiriz et al. (1999), Wagstaff and Cardie (2000), Basu (2005), Davidson and Ravi (2005)). Semi-supervised clustering can be defined as the problem of clustering a set of items in the presence of limited supervisory information such as pairwise constraints (e.g. two items must/cannot be in the same cluster) or labeled points. In contrast to our setting, in the semi-supervised case there is no training phase – the algorithm receives all examples (labeled and unlabeled) at the same time together with some distance or cost function and attempts to find a clustering that optimizes a given measure (usually based on the distance or cost function).

Source coreference resolution might alternatively be approached as a supervised clustering problem. Traditionally, methods or supervised clustering have

treated the pairwise link decisions as a classification problem. These approaches first learn a distance metric that optimizes the pairwise decisions; and then follow the pairwise classification with a clustering step. However, these traditional approaches have no obvious way of utilizing the available unlabeled information.

In contrast, we follow recent approaches to supervised clustering that propose ways to learn the distance measure in the context of the clustering decisions (Li and Roth, 2005; Finley and Joachims, 2005; McCallum and Wellner, 2003). This provides two advantages for the problem of source coreference resolution. First, it allows the algorithm to take advantage of the complexity of the rich structural dependencies introduced by the clustering step. Viewed traditionally as a hurdle, the structural complexity of clustering may be beneficial in the partially supervised case. We believe that provided with a few partially specified clustering examples, an algorithm might be able to generalize from the structural dependencies to infer correctly the clustering over all of the items. Second, considering pairwise decisions in the context of the clustering can arguably lead to more accurate classifiers.

Unfortunately, none of the supervised clustering approaches is readily applicable to the partially supervised case. However, by adapting the formal supervised clustering definition, which we do next, we can develop approaches to partially supervised clustering that take advantage of the unlabeled portions of the data.

4.4.1 Formal definition.

For partially supervised clustering we extend the formal definition of supervised clustering given by Finley and Joachims (2005). In the fully supervised setting, an algorithm is given a set S of n training examples $(x_1, y_1), ..., (x_n, y_n) \in X \times Y$, where X is the set of all possible sets of items and Y is the set of all possible clusterings of these sets. For a training example (x, y), $x = \{x_1, x_2, ..., x_k\}$ is a set of k items and $y = \{y_1, y_2, ..., y_r\}$ is a clustering of the items in x with each $y_i \subseteq x$. Additionally, each item can be in no more than one cluster ($\forall i, j . y_i \cap y_j = \emptyset$) and in the fully supervised case each item is in at least one cluster ($x = \bigcup y_i$). The goal of the learning algorithm is to acquire a function $h : X \rightarrow Y$ that can accurately cluster a (previously unseen) set of items.

In the context of source coreference resolution the training set contains one example for each document. The items in each training example are the NPs and the clustering over the items is the equivalence relation defined by the coreference information. For source coreference resolution, however, clustering information is unavailable for the non-source NPs. Thus, to be able to deal with this unlabeled component of the data we arrive at the setting of partially supervised clustering, in which we relax the condition that each item is in at least one cluster ($x = \bigcup y_i$) and replace it with the condition $x \supseteq \bigcup y_i$. The items with no linking information (items in $x \setminus \bigcup y_i$) constitute the unlabeled (unsupervised) component of the partially supervised clustering.

4.5 Structured Rule Learner

We develop a novel method for partially supervised clustering, which is motivated by the success of a rule learner (RIPPER) for coreference resolution (Ng and Cardie, 2002). We extend RIPPER so that it can learn rules in the context of single-link clustering, which is a clustering algorithm that is both theoretically suitable for our task (i.e. pronouns link to their single antecedent) and has exhibited good performance for coreference resolution (Ng and Cardie, 2002). We begin with a brief overview of RIPPER followed by a description of the modifications that we implemented. For ease of presentation, we assume that we are in the fully supervised case. We end this section by describing the changes for the partially supervised case.

4.5.1 The RIPPER Algorithm

RIPPER (for Repeated Incremental Pruning to Produce Error Reduction) was introduced by Cohen (1995) as an extension of an existing rule induction algorithm. Cohen (1995) showed that RIPPER produces error rates competitive with C4.5, while exhibiting better running times. RIPPER consists of two phases – a ruleset is grown and then optimized.

The ruleset creation phase begins by randomly splitting the training data into a rule-growing set (2/3 of the training data) and a pruning set (the remaining 1/3). A rule is then grown on the former set by repeatedly adding the *antecedent* (the feature value test) with the largest information gain until the accuracy of the rule becomes 1.0 or there are no remaining potential antecedents. Next the rule is applied to the pruning data and any rule-final sequence that reduces the accuracy of the rule is removed.

The optimization phase uses the full training set to first grow a replacement rule and a revised rule for each rule in the ruleset. For each rule, the algorithm then considers the original rule, the replacement rule, and the revised rule, and keeps the rule with the smallest description length in the context of the ruleset. After all rules are considered, RIPPER attempts to grow residual rules that cover data not already covered by the ruleset. Finally, RIPPER deletes any rules from the ruleset that reduce the overall minimum description length of the data plus the ruleset. RIPPER performs two rounds of this optimization phase.

4.5.2 The StRip Algorithm

The property of partially supervised clustering that we want to explore is the structured nature of the decisions. That is, each decision of whether two items (say a and b) belong to the same cluster has an implication for all items a' that belong to a's cluster and all items b' that belong to b's cluster.

We target modifications to RIPPER that will allow StRip (for Structured RIPPER) to learn rules that produce good clusterings in the context of single-link clustering. We extend RIPPER so that every time it makes a decision about a rule, it considers the effect of the rule on the overall clustering of items (as opposed to considering the instances that the rule classifies as positive/negative in isolation). More precisely, we precede every computation of rule performance (e.g. information gain or description length) by a transitive closure (i.e. single link clustering) of the data with respect to to the pairwise classifications. Following the transitive closure, all pairs of items that are in the same cluster are considered covered by the rule for performance computation.

The ruleset creation phase of the StRip algorithm is given in figure 4.2, with

modifications to the original RIPPER algorithm shown in bold.

Partially supervised case.

So far we described StRip only for the fully supervised case. We use a very simple modification to handle the partially supervised setting: we exclude the unlabeled pairs when computing the performance of the rules. Thus, the unlabeled items do not count as correct or incorrect classifications when acquiring or pruning a rule, although they do participate in the transitive closure. Links in the unlabeled data are inferred entirely through the indirect links between items in the labeled component that they introduce. In the example of figure 1.1, the two problematic unlabeled links are the link between the source mention "he" and the underlined non-source NP "Mr. Moussaoui" and the link between the underlined "Mr. Moussaoui" to any source mention of *Moussaoui*. While StRip will not reward any rule (or rule set) that covers these two links directly, such rules will be rewarded indirectly since they put the source *he* in the chain for the source *Moussaoui*.

StRip running time.

StRip's running time is generally comparable to that of RIPPER. We compute transitive closure by using a Union-Find structure, which runs in time $O(log^* n)$, which for practical purposes can be considered linear ($O(n)$) [1]. However, when computing the best information gain for a nominal feature, StRip has to make a pass over the data for each value that the feature takes, while RIPPER can split the data into bags and perform the computation in one pass.

[1]For the transitive closure, n is the number of items in a document, which is $O(\sqrt{k})$, where k is the number of NP pairs. Thus, transitive closure is sublinear in the number of training instances.

```
procedure StRip(TrainData){
  GrowData, PruneData = Split(TrainData);
  //Keep instances from the same document together
  while(there are positive uncovered instances) {
    r = growRule(GrowData);
    r = pruneRule(r, PruneData);
    DL = relativeDL(Ruleset);
    if(DL ≤ minDL + d bits)
      Ruleset.add(r);
      Mark examples covered by r as +;
    else
      exit loop with Ruleset
  }
}
procedure grow(growData){
  r = empty rule;
  for(every unused feature f){
    if (f is nominal feature) {
      for(every possible value v of f) {
        mark all instances that have values of v for f with +;
        compute the transitive closure of the positive instances
        //(including instances marked + from previous rules);
        compute the infoGain for the future/value combination;
      }
    } else{ //Numeric feature
      create one bag for each feature value and split the instances into bags;
      do a forward and a backward pass over the bags keeping a running
      clustering and compute the information gain for each value;
    }
  }
  add the future/value pair with the best infoGain to r;
  growData = growData - all negative instances;
  return r;
}
procedure prune(r, pruneData){
  for(all antecedents a in the rule){
    apply all antecedents in r up to a to pruneData;
    compute the transitive closure of the positive instances;
    compute A(a) -- the accuracy of the rule up to antecedent a;
  }
  Remove all antecedents after the antecedent for which A(a) is maximum.
}
```

Figure 4.2: The StRip algorithm. Additions to RIPPER are shown in bold.

4.6 Evaluation and Results

This section describes the source coreference data set, the baselines, our implementation of StRip, and the results of our experiments.

4.6.1 Data set

For evaluation we use the aforementioned Version 1.2 of the MPQA corpus (Wiebe et al., 2005b). As a reminder, the corpus consists of 535 documents from the world press. All documents in the collection are manually annotated with phrase-level opinion information following the annotation scheme of Wiebe et al. (2005b). Discussion of the annotation scheme is carried in Chapter 2; for the purposes of the source coreference evaluation, it suffices to say that the annotations include the source of each opinion and coreference information for the sources (e.g. source coreference chains). The corpus contains no additional noun phrase coreference information.

For our experiments, we randomly split the data set into a training set consisting of 400 documents and a test set consisting of the remaining 135 documents. We use the same test set for all experiments, although some learning runs were trained on 200 training documents (see next Subsection). The test set contains a total of 4736 source NPs (average of 35.34 source NPs per document) split into 1710 total source NP chains (average of 12.76 chains per document) for an average of 2.77 source NPs per chain.

4.6.2 Implementation

We implemented the StRip algorithm by modifying JRip – the java implementation of RIPPER included in the WEKA toolkit (Witten and Frank, 2000). The WEKA implementation follows the original RIPPER specification. We changed the implementation to incorporate the modifications suggested by the StRip algorithm; we also modified the underlying data representations and data handling techniques for efficiency. Also due to efficiency considerations, we train StRip only on the 200-document training set.

4.6.3 Competitive baselines

We compare the results of the new method to three fully supervised baseline systems, each of which employs the same traditional coreference resolution approach. In particular, we use the aforementioned algorithm proposed by Ng and Cardie (2002), which combines a pairwise NP coreference classifier with single-link clustering.

For one baseline, we train the coreference resolution algorithm on the *MPQA src* corpus — the labeled portion of the MPQA corpus (i.e. NPs from the source coreference chains) with unlabeled instances removed.

The second and third baselines investigate whether the source coreference resolution task can benefit from NP coreference resolution training data *from a different domain*. Thus, we train the traditional coreference resolution algorithm on the *MUC6* and *MUC7* coreference-annotated corpora[2] that contain documents similar in style to those in the MPQA corpus (e.g. newspaper articles),

[2]We train each baseline using both the development set and the test set from the corresponding MUC corpus.

Table 4.2: Performance of the best runs. For SVMs, γ stands for RBF kernel with the shown γ parameter.

	Measure	Rank	Method and parameters	Instance selection	B^3	MUC score	Positive Identification			Actual Pos. Ident.		
							Prec.	Recall	F1	Prec.	Recall	F1
400 Training Documents	B^3	1	svm C10 γ0.01	none	**81.8**	71.7	80.2	43.7	56.6	57.5	62.9	60.2
		5	ripper asc L2	soon2	**80.7**	72.2	74.5	45.2	56.3	55.1	62.1	58.4
	MUC Score	1	svm C10 γ0.01	soon1	77.3	**74.2**	67.4	51.7	58.5	37.8	70.9	49.3
		4	ripper acs L1.5	soon2	78.4	**73.6**	68.3	49.0	57.0	40.0	69.9	50.9
	Positive ident.	1	svm C10 γ0.05	soon1	72.7	73.9	60.0	57.2	**58.6**	37.8	71.0	49.3
		4	ripper acs L1.5	soon1	78.9	73.6	68.8	48.9	**57.2**	40.0	69.9	50.9
	Actual pos. ident.	1	svm C10 γ0.01	none	81.8	71.7	80.2	43.7	56.6	57.5	62.9	**60.2**
		2	ripper asc L4	soon2	73.9	69.9	81.1	40.2	53.9	69.8	52.5	**60.0**
200 Training Documents	B^3	1	ripper acs L4	none	**81.8**	67.8	91.4	32.7	48.2	72.0	52.5	60.6
		9	svm C10 γ0.01	none	**81.4**	70.3	81.6	40.8	54.4	58.4	61.6	59.9
	MUC Score	1	svm C1 γ0.1	soon1	74.8	**73.8**	63.2	55.2	58.9	32.1	74.4	44.9
		5	ripper acs L1	soon1	77.9	**0.732**	71.4	46.5	56.3	37.7	69.7	48.9
	Positive ident.	1	svm C1 γ0.1	soon1	74.8	73.8	63.2	55.2	**58.9**	32.1	74.4	44.9
		4	ripper acs L1	soon1	75.3	72.4	69.1	48.0	**56.7**	33.3	72.3	45.6
	Actual pos. ident.	1	ripper acs L4	none	81.8	67.8	91.4	32.7	48.2	72.0	52.5	**60.6**
		10	svm C10 γ0.01	none	81.4	70.3	81.6	40.8	54.4	58.4	61.6	**59.9**

but emanate from different domains.

For all baselines we targeted the best possible systems by trying two pairwise NP classifiers (RIPPER and an SVM in the SVM^{light}implementation (Joachims, 1998)), many different parameter settings for the classifiers, two different feature sets, two different training set sizes (the full 400-document training set and a smaller training set consisting of 200 documents (half of the documents selected at random)), and three different instance selection algorithms[3]. This variety of classifier and training data settings was motivated by reported differences in performance of coreference resolution approaches with respect to these variations (Ng and Cardie, 2002). In the experiments below we give detailed results for the first baseline (trained on the MPQA src corpus) in order to observe trends across parameters. For the rest of the baselines, we report the best performance of each of the algorithms on the MPQA test data.

Table 4.3: Results for Source Coreference. *MPQA src* stands for the MPQA corpus limited to only source NPs, while *MPQA all* contains the unlabeled NPs.

ML Framework	Training set	Classifier	B^3	precision	recall	F1
Fully supervised	MUC6	SVM	81.2	*72.6*	*52.5*	60.9
		RIPPER	80.7	*57.4*	*63.5*	60.3
	MUC7	SVM	81.7	*65.6*	*55.9*	60.4
		RIPPER	79.7	*71.6*	*48.5*	57.9
	MPQA src	SVM	81.8	*57.5*	*62.9*	60.2
		RIPPER	81.8	*72.0*	*52.5*	60.6
		StRip	82.3	*76.5*	*56.1*	64.6
Partially supervised	MPQA all	StRip	**83.2**	*77.1*	*59.4*	**67.1**

4.6.4 Evaluation

In addition to the baselines described above, we evaluate StRip both with and without unlabeled data. That is, we train on the MPQA corpus StRip using either all NPs or just opinion source NPs.

Baseline trends

Table 4.2 lists the results of the best performing runs for the MPQA src trained baseline. The upper half of the table gives the results for the runs that were trained on 400 documents and the lower half contains the results for the 200-document training set. We evaluated using the two widely used performance measures for coreference resolution – MUC score (Vilain et al., 1995) and B^3 (Bagga and Baldwin, 1998) (for a detailed description of the performance measures, please refer to Section 6). In addition, we used performance metrics (precision, recall and F1) on the identification of the positive class. We compute the latter in two different ways – either by using the pairwise decisions as the clas-

[3]The goal of the instance selection algorithms is to balance the data, which contains many more negative than positive instances.

sifiers output them or by performing the clustering of the source NPs and then considering a pairwise decision to be positive if the two source NPs belong to the same cluster. The second option (marked *actual* in Table 4.2) should be more representative of a good clustering, since coreference decisions are important only in the context of the clusters that they create.

Table 4.2 shows the performance of the best RIPPER and SVM runs for each of the four evaluation metrics. The table also lists the rank for each run among the rest of the runs.

The absolute B^3 and MUC scores for source coreference resolution are comparable to reported state-of-the-art results for NP coreference resolutions. Results should be interpreted cautiously, however, due to the different characteristics of our data. Our documents contained 35.34 source NPs per document on average, with coreference chains consisting of only 2.77 NPs on average. The low average number of NPs per chain may be producing artificially high scores for the B^3 and MUC scores as the modest results on positive class identification indicate.

From the relative performance of our runs, we observe the following trends. First, SVMs trained on the full training set outperform RIPPER trained on the same training set as well as the corresponding SVMs trained on the 200-document training set. The RIPPER runs exhibit the opposite behavior – RIPPER outperforms SVMs on the 200-document training set and RIPPER runs trained on the smaller data set exhibit better performance. Overall, the single best performance is observed by RIPPER using the smaller training set.

Another interesting observation is that the B^3 measure correlates well with

good "actual" performance on positive class identification. In contrast, good MUC performance is associated with runs that exhibit high recall on the positive class. This confirms some theoretical concerns that the MUC score does not reward algorithms that recognize well the absence of links. In addition, the results confirm our conjecture that "actual" precision and recall are more indicative of the true performance of coreference algorithms. Due to these findings, for the rest of this Section we report only the B^3 measure and "actual" F1 (which we term simply F1 for the rest of the Section).

Results

Results are shown in Table 4.3. The first six rows of results correspond to the fully supervised baseline systems trained on different corpora — *MUC6*, *MUC7*, and *MPQA src*. The seventh row of results shows the performance of StRip using only labeled data. The final row of the table shows the results for partially supervised learning **with** unlabeled data. The table lists results from the best performing run for each algorithm.

Performance among the baselines trained on the *MUC* data is comparable. However, the two baseline runs trained on the *MPQA src* corpus (i.e. results rows five and six) show slightly better performance on the B^3 metric than the baselines trained on the *MUC* data, which indicates that for our task the similarity of the documents in the training and test sets appears to be more important than the presence of complete supervisory information. (Improvements over the RIPPER runs trained on the MUC corpora are statistically significant[4], while improvements over the SVM runs are not.)

[4]Using the Wilcoxon matched-pairs signed-ranks test ($p < 0.05$).

Table 4.3 also shows that StRip outperforms the baselines on both B^3 and F1 performance metrics. StRip's performance is higher than the baselines when trained on *MPQA src* (improvement not statistically significant, $p > 0.20$) and even better when trained on the full MPQA corpus, which includes the unlabeled NPs (improvement over the baselines and the former StRip run statistically significant). These results confirm our hypothesis that StRip improves due to two factors: first, considering pairwise decisions in the context of the clustering function leads to improvements in the classifier; and, second, StRip can take advantage of the unlabeled portion of the data.

StRip's performance is all the more impressive considering the strength of the SVM and RIPPER baselines, which represent the best runs across the 336 different parameter settings tested for SVM^{light} and 144 different settings tested for RIPPER. All four of the StRip runs using the full MPQA corpus (we vary the loss ratio for false positive/false negative cost) outperform those baselines.

Generally, StRip is applicable to other problems that fit in the partially supervised problem defintion. It is possible, for example, that StRip can be used for traditional NP coreference resolution to preserve annotation effort. Experimental results in this chapter show that StRip can utilize unsupervised examples to learn a better classification function as compared to using only fully supervised data for part of the items. However, our experiments do not provide any data on whether annotation effort can be reduced by annotating documents partially as opposed to annotating fewer documents fully. We leave that evaluation for future work.

4.7 Chapter Summary

In this chapter we discussed the problem of source coreference resolution. Due to similarities with noun phrase coreference resolution, we build on approaches to NP coreference resolution using a similar pairwise learner with similar features. The partially supervised nature of the problem, however, leads us to define and approach it as the novel problem of partially supervised clustering. We propose and evaluate StRip, a new algorithm for the task of source coreference resolution and empirically observe that StRip outperforms competitive baselines.

CHAPTER 5

TOPIC IDENTIFICATION

In this chapter we address the problem of topic identification for opinion summarization. Parts of this chapter appear in Stoyanov and Cardie (2008a) and Stoyanov and Cardie (2008b).

As mentioned in Chapter 1, extracting topics of fine-grained opinions has proven to be a difficult task. Previous work has failed to provide a definition of opinion topics that can be effectively operationalized and used for annotating fine-grained opinions with topics (Wilson, 2005; Wiebe, 2005). The lack of corpora containing annotations of topics of fine-grained opinions, in turn, has hindered the progress in formulating approaches for automatic extraction of topics of fine-grained opinions. Nonetheless, topics remain an important component of an opinion, and topic extraction remains a critical step for sentiment analysis systems.

We address the problem of topic identification by providing a new, operational definition of *opinion topic* in which the topic of an opinion depends on the context in which its associated opinion expression occurs. We use this definition to create a methodology for performing opinion topic annotation. We apply the methodology to extend the existing MPQA corpus (Wiebe et al., 2005b) with manually annotated topic information (and refer to the extended corpus hereafter as the $\text{MPQA}_{\text{TOPIC}}$ corpus). Inter-annotator agreement results for the manual annotations are reasonably strong across a number of metrics.

We also present a novel method for the automatic identification of general-purpose opinion topics that, following our new definition, treats the problem

as an exercise in topic coreference resolution. We evaluate the computational approach using the $\text{MPQA}_{\text{TOPIC}}$ corpus. The results of experiments that evaluate our topic identification method in the context of fine-grained opinion analysis are promising: using either automatically or manually identified topic spans, we achieve topic coreference scores that statistically significantly outperform two topic segmentation baselines across three coreference resolution evaluation measures (B^3, α and CEAF). For the B^3 metric, for example, the best baseline achieves a topic coreference score on the $\text{MPQA}_{\text{TOPIC}}$ corpus of 0.55 while our topic coreference algorithm scores 0.57 and 0.71 using automatically, and manually, identified topic spans, respectively.

In the remainder of the chapter, we define opinion topics (Section 5.1), present related work (Section 5.2), and motivate and describe the key idea of topic coreference that underlies our methodology for both the manual and automatic annotation of opinion topics (Section 5.3). Creation of the $\text{MPQA}_{\text{TOPIC}}$ corpus is described in Section 5.4 and our topic identification algorithm, in Section 5.5. The evaluation methodology and results are presented in Section 5.6 including inter-annotator agreement results in Section 5.6.3.

5.1 Definitions and Examples

Consider the following opinion sentences:

(1)[$_{\text{OH}}$ *John*] <u>*adores*</u> [$_{\text{TARGET+TOPIC SPAN}}$ *Marseille*] *and visits it often.*

(2)[$_{\text{OH}}$ *Al*] <u>*thinks*</u> *that* [$_{\text{TARGET SPAN}}$ [$_{\text{TOPIC SPAN?}}$ *the government*] *should* [$_{\text{TOPIC SPAN?}}$ *tax gas*] *more in order to* [$_{\text{TOPIC SPAN?}}$ *curb* [$_{\text{TOPIC SPAN?}}$ CO_2 *emissions*]]].

As discussed previously, fine-grained subjectivity analysis should identify: the OPINION EXPRESSION as "adores" in Example 1 and "thinks" in Example 2; the POLARITY as positive in Example 1 and neutral in Example 2; and the OPINION HOLDER (OH) as "John" and "Al", respectively. To be able to discuss the opinion TOPIC in each example, we begin with three definitions:

– **Topic.** The TOPIC of a fine-grained opinion is the real-world object, event or abstract entity that is the subject of the opinion as intended by the opinion holder.

– **Topic span.** The TOPIC SPAN associated with an OPINION EXPRESSION is the closest, minimal span of text that mentions the topic.

– **Target span.** In contrast, we use TARGET SPAN to denote the span of text that covers the syntactic surface form comprising the contents of the opinion.

In Example 1, for instance, "Marseille" is both the TOPIC SPAN and the TARGET SPAN associated with the city of Marseille, which is the TOPIC of the opinion. In Example 2, the TARGET SPAN consists of the text that comprises the complement of the subjective verb "thinks". Example 2 illustrates why opinion topic identification is difficult: within the single target span of the opinion, there are multiple potential topics, each identified with its own topic span. Without more context, however, it is impossible to know which phrase indicates the intended topic. If followed by sentence 3, however,

(3)*Although he doesn't like government-imposed taxes, he thinks that a fuel tax is the only effective solution.*

the topic of Al's opinion in 2 is much clearer — it is likely to be fuel tax, denoted

via the TOPIC SPAN "tax gas" or "tax".

With these related definitions of three key aspects associated with opinion topic in mind, we next discuss work related to the area of opinion topic extraction.

5.2 Related Work

As mentioned in Chapter 2, several research efforts have focused on the extraction of the topic of an opinion in the related area of opinion extraction from product reviews (e.g. Kobayashi et al. (2004), Yi et al. (2003), Popescu and Etzioni (2005), Hu and Liu (2004)). For this specialized text genre, it has been sufficient to limit the notion of topic to mentions of product names and components and their attributes. Thus, topic extraction has been effectively performed as a lexicon look-up and techniques have focused on how to learn or acquire an appropriate lexicon for the task. While the techniques have been very successful for this genre of text, they have not been applied outside the product reviews domain. Further, there are analyses (Wiebe et al., 2005b) and experiments (Turney, 2002; Wilson et al., 2005) that indicate that lexicon-lookup approaches to subjectivity analysis will have limited success on general texts because subjective language is highly dependent on the context in the documents where it appears. While product reviews are naturally separated into domains in which subjective language has clearer interpretation, the same cannot be done easily for general documents.

Outside the product review domain, there has been little effort devoted to opinion topic annotation. The MPQA corpus, for example, was originally in-

tended to include topic annotations, but the task was abandoned after confirming that it was very difficult (Wiebe, 2005; Wilson, 2005), although target span annotation was subsequently added to Version 2.0 of the corpus. While useful, target spans alone will be insufficient for many applications: they neither contain information indicating which opinions are about the same topic, nor provide a concise textual representation of the topics.

Due to the lack of appropriately annotated corpora, the problem of opinion topic extraction has been largely unexplored in NLP. A notable exception is the work of Kim and Hovy (2006a). They propose a model that extracts opinion topics for subjective expressions signaled by verbs and adjectives. Their model relies on semantic frames and extracts as the topic the syntactic constituent at a specific argument position for the given verb or adjective. In other words, Kim and Hovy extract what we refer to as the target spans, and do so for a subset of the opinion-bearing words in the text. Although on many occasions target spans coincide with opinion topics (as in Example 1), we have observed that on many other occasions this is not the case (as in Example 2). Furthermore, hampered by the lack of resources with manually annotated targets, Kim and Hovy could provide only a limited evaluation.

As we have defined it, opinion topic identification bears some resemblance to the notion of topic segmentation in discourse, the goal of which is to partition a text into a linear sequence of topically coherent segments. Existing methods for topic segmentation typically assume that fragments of text (e.g. sentences or sequences of words of a fixed length) with similar lexical distribution are about the same topic; the goal of these methods is to find the boundaries where the lexical distribution changes (e.g. Choi (2000), Malioutov and Barzilay (2006)).

Opinion topic identification differs from topic segmentation in that opinion topics are not necessarily spatially coherent — there may be two opinions in the same sentence on different topics, as well as opinions that are on the same topic separated by opinions that do not share that topic. Nevertheless, we will compare our topic identification approach to a state-of-the-art topic segmentation algorithm (Choi, 2000) in the evaluation.

Other work has successfully adopted the use of clustering to discover entity relations by identifying entities that appear in the same sentence and clustering the intervening context (e.g. Hasegawa et al. (2004), Rosenfeld and Feldman (2007)). This work, however, considers named entities and heads of proper noun phrases rather than topic spans, and the relations learned are those commonly held between NPs (e.g. senator-of-state, city-of-state, chairman-of-organization) rather than a more general coreference relation.

5.3 A Coreference Approach to Topic Identification

Given our initial definition of opinion topics (Section 5.1), the next task is to determine what approaches might be employed for manual annotation and automatic identification of opinion topics. We begin this exercise by considering some of the problematic characteristics of opinion topics.

Multiple potential topics. As noted earlier via Example 2, a serious problem in opinion topic identification is the mention of multiple potential topics within the target span of the opinion. Although an issue for all opinions, this problem is typically more pronounced in opinions that do not carry sentiment (as in Example 2). Our current definition of opinion topic requires the human annotator

(or the NLP system) to decide which of the entities described in the target span, if any, refers to the intended topic. This decision can be aided by the following change to our definition of opinion topic, which introduces the idea of a context-dependent information focus: *the* TOPIC *of an opinion is the real-world entity that is the subject of the opinion as intended by the opinion holder* **based on the discourse context**.

With this modified definition in hand, and given Example 3 as the succeeding context for Example 2, we argue that the intended subject, and hence the TOPIC, of Al's opinion in 2 can be quickly identified as the FUEL TAX, which is denoted by the TOPIC SPANS "tax gas" in 2 and "fuel tax" in 3.

Opinion topics not always explicitly mentioned. In stark contrast to the above, on many occasions the topic is not mentioned explicitly at all within the target span, as in the following example:

(5)[$_{\text{OH}}$ *John*] *identified the violation of Palestinian human rights as one of the main factors.* TOPIC**:** ISRAELI-PALESTINIAN CONFLICT

We have further observed that the opinion topic is often not mentioned within the same paragraph and, on a few occasions, not even within the same document as the opinion expression.

Our Solution: Topic Coreference. With the above examples and problems in mind, we hypothesize that the notion of *topic coreference* will facilitate both the manual and automatic identification of opinion topics: **We say that two opinions are topic-coreferent if they share the same opinion topic.** In particular, we conjecture that judging whether or not two opinions are topic-coreferent is

easier than specifying the topic of each opinion (due to the problems described above).

Relying on the notion of topic coreference, we next introduce a new methodology for the manual annotation of opinion topics in text.

5.4 Constructing the $\text{MPQA}_{\text{TOPIC}}$ Corpus

Our topic annotation process begins with a corpus annotated with respect to fine-grained expressions of opinions (we use the MPQA corpus). To facilitate the opinion annotation process we developed a set of annotation instructions (included in Appendix C) based on the preceding discussion and a graphical user interface (GUI) that helps the annotator to keep track of the existing topics. Aided by the GUI, an annotator proceeds as follows:

1. The annotator opens a document manually annotated for fine-grained opinions. The GUI shows three panels (i) a panel containing a list of all opinions that are yet to be annotated — initially all opinions in the document (where each opinion is characterized by the words that signal the expression of the opinion, its source and its polarity), (ii) an initially empty panel that contains the current set of topic-coreferent clusters and, (iii) a panel containing the text of the document.
2. The annotator proceeds down the list of opinions that are yet to be annotated. Looking at the clusters of topic-coreferent opinions in panel (ii) as well as the text in panel (iii), the annotator decides whether the current opinion is coreferent with the opinions in any of the existing clusters

or should start a new topic. The annotator then drags the opinion to the appropriate cluster in panel (ii).

3. After dropping all opinions into the appropriate cluster, the annotator assigns a label to name each cluster, based on the opinions in the cluster[1].

4. In addition, we require the annotator to mark the spans of text that contributed to the topic coreference decision, since learning algorithms may benefit from this information. More specifically, the annotator marks the topic spans, which we view as secondary information, but information that can still be important for training automatic opinion identifiers. We allow the annotator to mark the topic spans at any time during the annotation process and allow marked topic spans to be anywhere in the document.

5. Finally, the annotator saves the document. The GUI checks the annotations to make sure that all opinions are assigned to a topic cluster, that all clusters are labeled and that all opinions are assigned a topic span.

Using this procedure, one person annotated opinion topics for a randomly selected set of 150 of the 535 documents in Version 1.2 of the MPQA corpus to form the $\text{MPQA}_{\text{TOPIC}}$ corpus. In addition, 20 of the 150 documents were selected at random and annotated by a second annotator for the purposes of an inter-annotator agreement study, the results of which are presented in Section 5.6.3.

[1]In reality, the annotator may assign a label to a cluster before assigning all opinions in the document. Indeed, we encourage the annotator to maintain a working label for each cluster.

5.5 Automatic Topic Identification

We perform manual annotation of opinion topics for the purpose of supporting automatic identification. In this section, we describe our method for automatically identifying opinion topics.

As mentioned in Section 5.3, our computational approach to opinion topic identification is based on topic coreference: For each document (1) find the clusters of coreferent opinions, and (2) label the clusters with the name of the topic.

Topic coreference resolution resembles another well-known problem in NLP — noun phrase (NP) coreference resolution discussed in Chapter 4. Therefore, we adapt a standard machine learning-based approach to NP coreference resolution (Soon et al., 2001; Ng and Cardie, 2002) for our purposes. Our adaptation has three steps: (i) identify the topic spans; (ii) perform pairwise classification of the associated opinions as to whether or not they are topic-coreferent; (iii) cluster the opinions according to the results of (ii); and, (iv) label each cluster with the name of the topic. Each step is discussed in more detail below.

Step I: Identifying Topic Spans

Decisions about topic coreference should depend on the text spans that express the topic. Ideally, we would be able to recover the topic span of each opinion and use its content for the topic coreference decision. However, the topic span depends on the topic itself, so it is unrealistic that topic spans can be recovered with simple methods. Nevertheless, in this initial work, we investigate two simple methods for automatic topic span identification and compare them to two manual approaches:

- **Sentence.** Assume that the topic span is the whole sentence containing the opinion.
- **Automatic.** A rule-based method for identifying the topic span (developed using MPQA documents that are not part of $\text{MPQA}_{\text{TOPIC}}$). Rules depend on the syntactic constituent type of the opinion expression and rely on syntactic parsing and grammatical role labeling.
- **Manual.** Use the topic span marked by the human annotator. We included this method to provide an upper bound on performance of the topic span extractor.
- **Modified Manual.** Meant to be a more realistic use of the manual topic span annotations, this method returns the manually identified topic span only when it is within the sentence of the opinion expression. When this span is outside the sentence boundary, this method returns the opinion sentence.

Of the 4976 opinions annotated across the 150 documents of $\text{MPQA}_{\text{TOPIC}}$, the topic spans associated with 4293 were within the same sentence as the opinion; 3653 were within the span extracted by our topic span extractor. Additionally, the topic spans of 173 opinions were outside of the paragraph containing the opinion.

Step II: Pairwise Topic Coreference Classification

The heart of our method is a pairwise topic coreference classifier. Given a pair of opinions (and their associated polarity and opinion holder information), the goal of the classifier is to determine whether the opinions are topic-coreferent.

We use the manually annotated data to automatically learn the pairwise classifier. Given a training document, we construct a training example for every pair of opinions in the document (each pair is represented as a feature vector). The pair is labeled as a positive example if the two opinions belong to the same topic cluster, and a negative example otherwise.

Pairwise coreference classification relies critically on the expressiveness of the features used to describe the opinion pair. We use three categories of features: positional, lexico-semantic and opinion-based features.

Positional features These features are intended to exploit the fact that opinions that are close to each other are more likely to be on the same topic. We use six positional features:

- **Same Sentence/Paragraph**[2] True if the two opinions are in the same sentence/paragraph.
- **Consecutive Sentences/Paragraphs** True if the two opinions are in consecutive sentences/paragraphs.
- **Number of Sentences/Paragraphs** The number of sentences/paragraphs that separate the two opinions.

TOPIC SPAN**-based lexico-semantic features** The features in this group rely on the topic spans and are recomputed with respect to each of the four topic span methods. The intuition behind this group of features is that topic-coreferent opinions are likely to exhibit lexical and semantic similarity within the topic span.

[2]We use sentence/paragraph to describe two features – one based on the sentence and one on the paragraph.

- **tf.idf** The cosine similarity of the tf.idf weighted vectors of the terms contained in the two spans.
- **Word overlap** True if the two topic spans contain any contain words in common.
- **NP coref** True if the two spans contain NPs that are determined to be coreferent by a simple rule-based coreference system.
- **NE overlap** True if the two topic spans contain named entities that can be considered aliases of each other.

Opinion features The features in this group depend on the attributes of the opinion. In the current work, we obtain these features directly from the manual annotations of the $\text{MPQA}_{\text{TOPIC}}$ corpus, but they might also be obtained from automatically identified opinion information using the methods referenced in Section 5.2.

- **Source Match** True if the two opinions have the same opinion holder.
- **Polarity Match** True if the two opinions have the same polarity.
- **Source-Polarity Match** False if the two opinions have the same opinion holder but conflicting polarities (since it is unlikely that a source will have two opinions with conflicting polarities on the same topic).

We employ three classifiers for pairwise coreference classification – an averaged perceptron (Freund and Schapire, 1998), SVM^{light} (Joachims, 1998) and the rule-learner described in Chapter 4 – RIPPER (Cohen, 1995). However, we report results only for the averaged perceptron, which exhibited the best performance.

Step III: Clustering

Pairwise classification provides an estimate of the likelihood that two opinions are topic-coreferent. To form the topic clusters, we follow the pairwise classification with a clustering step. We selected a simple clustering algorithm – single-link clustering, which has shown good performance for NP coreference. Given a threshold, single-link clustering proceeds by assigning pairs of opinions with a topic-coreference score above the threshold to the same topic cluster and then performs transitive closure of the clusters.

As discussed above our choice of clustering algorithm was influenced by the success of single-link clustering for coreference resolution. It is worth nothing, however, that topic clusters and NP clusters have properties – topic clusters tend to be larger (i.e. there are more items per cluster on average). Arguably, different clustering algorithm can perform better for topic clustering given the different properties. To address this issue, we experimented using other clustering algorithms: complete-link, best-first and last-first. Single-link clustering was selected empirically as it showed results that were similar or better compared to the other clustering algorithms.

Step IV: Labeling Topic Clusters

We use a simple approach to assign topic labels – for each topic cluster, we collect all words in the topic spans of all opinions in the cluster; we clean the resulting list of words by removing stopwords; out of the words that remain in the list, we select the top three words in terms of *tf.idf* and assign them as the label of the cluster. Due to practical difficulties in evaluating the topic labels, we do not perform an evaluation of the label assignment part of the algorithm in

this chapter. Instead, label assignment is evaluated in Chapter 7 as part of the evaluation of complete summaries.

5.6 Evaluation Methodology and Results

For training and evaluation we use the 150-document $\text{MPQA}_{\text{TOPIC}}$ corpus. All machine learning methods were tested via 10-fold cross validation. In each round of cross validation, we use eight of the data partitions for training and one for parameter estimation (we varied the threshold for the clustering algorithm), and test on the remaining partition. We report results for three evaluation measures described in the next Section using the four topic span extraction methods introduced in Section 5.5. The threshold is tuned separately for each evaluation measure. As noted earlier, all runs obtain opinion information from the $\text{MPQA}_{\text{TOPIC}}$ corpus (i.e. in the evaluation in this Chapter we do not incorporate automatic opinion extraction as opposed to the evaluation in Chapter 7, in which we present and evaluate our end-to-end automatic summarization system).

5.6.1 Evaluation Metrics

Because there is disagreement among researchers with respect to the proper evaluation measure for NP coreference resolution, we use three generally accepted metrics to evaluate our topic coreference system.

B^3. B-CUBED (B^3) is a commonly used NP coreference metric (Bagga and Baldwin, 1998). It calculates precision and recall for each item (in our case, each opinion) based on the number of correctly identified coreference links, and then

computes the average of the item scores in each document. Precision/recall for an item i is computed as the proportion of items in the intersection of the response (system-generated) and key (gold standard) clusters containing i divided by the number of items in the response/key cluster. The B^3 evaluation measure is described in more detail in Chapter 6.

CEAF. As a representative of another group of coreference measures that rely on mapping response clusters to key clusters, we selected Luo's (2005) CEAF score (short for Constrained Entity-Alignment F-Measure). Similar to the ACE (2006) score, CEAF operates by computing an optimal mapping of response clusters to key clusters and assessing the goodness of the match of each of the mapped clusters. CEAF score is also discussed in more detail in Chapter 6.

Krippendorff's α. Finally, we use Passonneau's (2004) generalization of Krippendorff's (1980) α — a standard metric employed for inter-annotator reliability studies. Krippendorff's α is based on a probabilistic interpretation of the agreement of coders as compared to agreement by chance. While Passonneau's innovation makes it possible to apply Krippendorff's α to coreference clusters, the probabilistic interpretation of the statistic is unfortunately lost.

Initially we intended to use a fourth metric – the MUC score (Vilain et al., 1995) (used in Chapter 4), but discovered that it is inappropriate for our problem. Topic coreference clusters tend to be much larger than NP coreference clusters, while the MUC score is not strict enough for responses that link too many clusters together (Bagga and Baldwin, 1998), leading to an extremely high MUC F-score (.920) for the simple baseline that groups all opinions in one cluster.

Table 5.1: Baseline results.

	α	B^3	CEAF
One cluster	-.1017	.3739	.2976
One per cluster	.2238	.2941	.2741
Same paragraph	.3123	.5542	.5090
Choi	.5399	.3734	.5370

5.6.2 Topic Coreference Baselines

We compare our topic coreference system to four baselines. The first two are the "default" baselines:

- **one topic** – assigns all opinions to the same cluster.
- **one opinion per cluster** – assigns each opinion to its own cluster.

The other two baselines attempt to perform topic segmentation (discussed in Section 5.2) and assign all opinions within the same segment to the same opinion topic:

- **same paragraph** – simple topic segmentation by splitting documents into segments at paragraph boundaries.
- **Choi 2000** – Choi's (2000) state-of-the-art approach to finding segment boundaries. We use the freely available C99 software described in Choi (2000), varying a parameter that allows us to control the average number of sentences per segment and reporting the best result on the test data.

Results for the four baselines are shown in Table 5.6.2. As expected, the two baselines performing topic segmentation show substantially better scores than the two "default" baselines.

Table 5.2: Inter-annotator agreement results.

	α	B^3	CEAF
All opinions	.5476	.6424	.6904
Sentiment-bearing opinions	.7285	.7180	.7967
Strong sentiment-bearing opinions	.7669	.7374	.8217

5.6.3 Inter-annotator Agreement

As mentioned previously, out of the 150 annotated documents, 20 were annotated by two annotators for the purpose of studying the agreement between coders. Results of the inter-annotator agreement study are shown in Table 5.6.3.

We compute agreement for three subsets of opinions: all available opinions, only the sentiment-bearing opinions and the subset of sentiment-bearing opinions judged to have polarity of medium or higher[3].

The results support our conjecture that topics of sentiment-bearing opinions are much easier to identify: inter-annotator agreement for opinions with non-neutral polarity (SENTIMENT-BEARING OPINIONS) improves by a large margin for all measures. As in other work in subjectivity annotation, we find that strong sentiment-bearing opinions are easier to annotate than sentiment-bearing opinions in general.

A problem with using coreference resolution scoring algorithms for our inter-annotator agreement studies is that it is hard to translate absolute scores to quality of agreement. Of the four metrics, only Krippendorff's α attempts to incorporate a probabilistic interpretation (Passonneau, 2004). It is generally agreed that an α score above 0.66 indicates reliable agreement. Our inter-

[3]These are identified using the manually annotated strength, i.e. intensity, values.

annotator agreement exhibits a score under that threshold when computed over all opinions (0.54) and a score above the threshold when computed over the sentiment-bearing opinions (0.71). However, as discussed above, in adapting α to the problem of coreference resolution, the score loses its probabilistic interpretation. For example, the α score requires that a pairwise distance function between clusters is specified. We used one sensible choice for such a function (we measured the distance between clusters A and B as $dist(A, B) = (2 * |A \cap B|)/(|A| + |B|)$), but other sensible choices for the distance lead to much higher scores. Furthermore, we observed that the behavior of the α score can be rather erratic — small changes in one of the clusterings can lead to big differences in the score.

Arguably, the numerical magnitudes of the inter-annotator agreement scores are insufficient to judge the quality of the annotation agreement. Perhaps a better indicator of the reliability of the coreference annotation is a comparison with the baselines, discussed in the previous section and shown in Table 5.6.2. As Table 5.6.2 shows, all baselines score significantly lower than the inter-annotator agreement with the exception of the Choi (2000) baseline when evaluated using the α score. Furthermore, the baseline that groups opinions by paragraph appears to agree much better with the annotator, which is to be expected given our understanding of the way that topics in general, and opinion topics in particular, are expressed in discourse. With one exception, the inter-annotator agreement scores are also higher than those for the learning-based approach (results shown in the Table 5.6.4), as would typically be expected. The exception is the classifier that uses the manual topic spans, but as we argued earlier these spans carry significant information about the decision of the annotator. This result leads us to believe that opinion topic annotation can be performed reliably.

Table 5.3: Results for the topic coreference algorithms.

	B^3	α	CEAF
Sentence	.5749	.4032	.5393
Rule-based	.5730	.4056	.5420
Modified manual	.6416	.5134	.6124
Manual	.7097	.6585	.6184

5.6.4 Learning methods

Results for the learning-based approaches are shown in Table 5.6.4. First, we see that each of the learning-based methods outperforms the baselines. This is the case even when sentences are employed as a coarse substitute for the true topic span. A Wilcoxon Signed-Rank test shows that differences from the baselines for the learning-based runs are statistically significant for the B^3 and α measures ($p < 0.01$); for CEAF, using sentences as topic spans for the learning algorithm outperforms the SAME PARAGRAPH baseline ($p < 0.05$), but the results are inconclusive when compared with the system of CHOI.

In addition, relying on manual topic span information (MANUAL and MODIFIED MANUAL) allows the learning-based approach to perform significantly better than the two runs that use automatically identified spans ($p < 0.01$, for all three measures). The improvement in the scores hints at the importance of improving automatic topic span extraction.

5.7 Chapter Summary

In this chapter we presented a new, operational definition of opinion topics in the context of fine-grained subjectivity analysis. Based on this definition, we in-

troduced an approach to opinion topic identification that relies on the identification of topic-coreferent opinions. We further employed the opinion topic definition for the manual annotation of opinion topics to create the $\text{MPQA}_{\text{TOPIC}}$ corpus. Inter-annotator agreement results show that opinion topic annotation can be performed reliably. Finally, we proposed an automatic approach for identifying topic-coreferent opinions that significantly outperforms all baselines across three coreference evaluation metrics.

CHAPTER 6

EVALUATION MEASURES

A scientific approach to opinion summarization requires evaluation metrics in order to quantitatively compare summaries produced by different systems or different versions of the same system. Unfortunately, as is often the case in NLP, there is no "natural" measure of the goodness of opinion summaries. In this chapter we address the issue of evaluation by proposing two novel performance metrics for opinion summaries.

The metrics that we propose are inspired by two evaluation measures used for coreference resolution and one used for an entity extraction task. We begin this chapter by discussing these three measures in Section 6.1. We then briefly discuss requirements for opinion summary evaluation metrics in Section 6.2. Finally, we present our novel metrics for opinion summary evaluation in Section 6.3.

6.1 Existing Evaluation Metrics

The algorithms for scoring opinion summaries that we propose are inspired by three existing evaluation metrics – the B^3 score, the ACE Cost-Based Evaluation Metric and CEAF. Two of these scores, the B^3 score and CEAF, are used for evaluation of coreference resolution output and have been used (and described) elsewhere in this thesis. Nevertheless, we begin this chapter with a slightly more detailed description of all three metrics on which our novel scoring algorithms are based.

6.1.1 B^3 Score.

The B^3 score (Bagga and Baldwin, 1998) is a coreference resolution score, which evaluates the quality of an automatically generated clustering of items (the system response) as compared to a gold-standard clustering of the same items (the key). The B^3 score is computed for each entity i based on the number of entities in common between i's response and key clusters. More precisely, the B^3 recall for entity i is computed as:

$$Recall_i = \frac{num\ of\ correct\ items\ in\ R_i}{num\ of\ items\ in\ S_i},$$

where R_i and S_i are the clusters that contain i in the response and the key, respectively. The recall for a document is computed as the average over all items. Precision is computed by switching the roles of the key and the response and the reported score is the harmonic average of precision and recall (the F score).

6.1.2 The ACE Cost-Based Evaluation Metric.

In a nutshell, the task covered in ACE (ACE, 2006) is concerned with extracting information about real-world entities (called tokens) that fall in specific semantic classes (e.g., people, locations). Each extracted entity is characterized by a set of attributes (e.g. name, gender) and its mentions in the text (e.g., spans of the text that refer to the entity).

The ACE score relies on a *Value* score that reflect how well individual items (tokens) in the key and the response are matched. Given a correspondence between items in the key and the response, the overall score is computed as the sum of the *Value* scores of all of the response's items as compared to their corresponding key item, divided by the sum of the *Value* of all of the gold-standard

(key) items compared to themselves (i.e., the maximum value is 1). The match between response and key items is based on a globally optimal assignment, which maximizes the overall score (subjected to one-to-one match between the two set of items). The *Value* of each individual item (token) is defined as the product of the score for how accurately the token's attributes are recognized and the token's mentions are detected.

$$Value(token) = ElementVal(token) \cdot MentionsVal(token),$$

where *ElementVal* scores how well the attributes match if the token is mapped (has a corresponding key token), weighted by the inherent value of the attributes and reduced for any attribute errors by a penalty depending on the attribute type (W_{err-a}). If the system item is unmapped, then the value of the item is set to a false alarm penalty. *MentionsVal* is a score of how well mentions of the token (item) are extracted and is computed as the sum of the mutual mention values (MMV) between the mentions of the response token and the key token if the mention is mapped. The MMV score is weighted by the mention type and reduced for any mention attribute errors. For unmapped mentions, the score is weighted by the product of a false alarm penalty factor, *WM-FA*, and a co-reference weighting factor, *WM-CR*, if the system mention happens to correspond to a different legitimate key mention. As before, pairing of response and key mentions is optimal, subject to the one-to-one mapping constraint.

The formulas for $ElementVal$ and MMV are shown in Figure 6.1.2.

Response mentions and key mentions can correspond only if their spans in the text have a certain preset minimum mutual overlap.

$$
ElementVal(sys) = \begin{cases} min\left(\dfrac{\prod_{a \in attributes} attrVal(a_{sys})}{\prod_{a \in attributes} attrVal(a_{ref})} \right) . \prod_{a \in attributes} W_{err-a}, & if\ sys\ mapped \\ \left(\prod_{a \in attributes} attrVal(a_{sys}) \right) . W_{FA}, & if\ not\ mapped \end{cases}
$$

$$
MMV(mention\_sys) = \begin{cases} min\left(\dfrac{MTypeVal(mention_{sys})}{MTypeVal(mention_{ref})} \right) . \prod_{a \in attributes} W_{Mention-err}, & if\ sys\ mapped \\ -\Big(MTypeVal(mention_{ref}) \Big) . (W_{M-FA} . W_{M-CR}), & if\ not\ mapped \end{cases}
$$

Figure 6.1: Formulas for computing $ElementVal$ and MMV.

6.1.3 CEAF Score.

Luo's (2005) CEAF score (for Constrained Entity-Alignment F-Measure) is a coreference resolution evaluation metric resembling the ACE score. Similar to ACE, CEAF relies on a measure of how well a response cluster matches a key cluster and computes an optimal mapping between key and response clusters. CEAF differs from ACE in that it computes recall by dividing the score of the optimal match by the score for mapping the key to itself (i.e. the maximum is 1) and precision by dividing by the score of matching the response to itself. The reported CEAF score is the harmonic average (F-score) of precision and recall.

Luo (2005) suggests several functions to score the goodness of the match of a key cluster A and response cluster B. We borrow one of these functions: $\phi(A, B) = (2 * |A \cap B|)/(|A| + |B|)$. In other words, the score for the match is the number of items the two clusters have in common proportional to the combined size of the two clusters.

6.2 Requirements for an Opinion Summary Evaluation Metric

In Section 1.3, we proposed two different types of opinion summary based on application needs. These two types of opinion summaries differ in what qualities make for a good automatically extracted opinion summary as compared to a gold standard summary. Next, we briefly remind the reader of the form of each of the two types of summary and discuss the requirements for summaries of the corresponding type to be considered correct when compared to their gold standard.

Opinion Set Summary. In an *opinion set* summary, multiple opinions from a source on a topic are simply collected into a single set (without necessarily analyzing them for the overall trend). An opinion set summary is correct if it groups together fine-grained opinions from the same source and on the same topic.

Aggregate Opinion Summary. In an *aggregate opinion* summary, multiple opinions from a source on a topic are merged into a single aggregate opinion that represents the cumulative opinion of the source on that topic considering the document as a whole

An aggregate opinion summary is similar in many ways to an extracted entity (i.e., the task for which the ACE score is used). For an aggregate summary to be correct, each of its aggregate opinions from a source on a topic has to be extracted correctly along with its attributes (for us, those are the name of the source, the polarity and the name of the topic). Optionally, an aggregate summary could be judged on how many of the individual fine-grained opinions (that make up each aggregate opinion in the summary) it identifies correctly.

6.3 Evaluation Metrics for Opinion Summaries

Finally, we propose two evaluation metrics for evaluation of opinion summaries. The first evaluation metric, Doubly-linked B^3 score, is suitable for evaluating opinion summaries of the opinion set form, while the second, Opinion Summary Evaluation Metric (OSEM), is a hybrid evaluation metric that can be used for both kind of summaries.

6.3.1 Doubly-linked B^3 score

Opinion set summaries are similar to the output of coreference resolution – both target grouping a set of items together. However, the two differ in an important way: opinion sets are doubly linked – two opinions are in the same set when they have the same source **and** the same topic. We address this difference by introducing a modified version of the B^3 algorithm – the Doubly Linked B^3 (DLB^3) score . DLB^3 computes the recall for each item (opinion) i as an average of the recall with respect to the source ($recall_i^{src}$) and the recall with respect to the topic ($recall_i^{topic}$). More precisely:

$$DLB^3\ recall_i = (recall_i^{src} + recall_i^{topic})/2$$
$$recall_i^{src} = \frac{num\ of\ correct\ items\ in\ R_i^{src}}{num\ of\ items\ in\ S_i^{src}},$$

where R_i^{src} and S_i^{src} are the sets of all opinions attributed to the source of opinion i in the response and the key, respectively. $recall_i^{topic}$ is computed similarly. As with the B^3 score, precision is computed by switching the key and the response and the DLB^3 score is reported as the harmonic average of precision and recall.

6.3.2 Opinion Summary Evaluation Metric

Finally, we propose a novel Opinion Summary Evaluation Metric (OSEM) that combines ideas from the ACE and the CEAF scores and can be used for both types of summaries.

The OSEM metric compares two opinion summaries – the key, K, and the response, R, containing a number of "summary opinions", each of which is comprised of one or more fine-grained opinions. (In aggregate opinion summaries, the fine-grained opinions are aggregated; in opinion set summaries, they are not.) Each summary opinion is characterized by three attributes (the source name, the polarity and the topic name) and by the set of fine-grained opinions that were joined to form the summary opinion. OSEM evaluates how well the key's summary opinions are extracted in the response by establishing a mapping $f : K \rightarrow R$ between the summary opinions in the key and the response. A value is associated with each mapping, defined as: $value_f(K, R) = \sum_{A \in K} match(A, f(A))$, where $match(A, B)$ is a measure of how well opinions A and B match (discussed below). Similarly to the ACE and CEAF score, OSEM relies on the globally optimal matching $f* = argmax_f(value_f(K, R))$ between the key and the response. OSEM takes CEAF's approach and compute precision as $value_{f*}(K, R)/value(R, R)$ and recall as $value_{F*}(K, R)/value(K, K)$. The final reported OSEM score is the harmonic average (F-score) of precision and recall. The optimal matching is computed efficiently using the Kuhn-Munkres algorithm.

The remaining details of the OSEM score are in the way $match(A, B)$, the score for a match between summary opinions A and B, is computed. We borrow from the ACE score and compute the match score as a combination of how well the attributes of the summary opinion are matched and how well the individual

opinion mentions (i.e., the fine-grained opinions in the text that form the aggregate opinion) are extracted. More precisely we define,

$$match(A, B) = attrMatch(A, B)^{\alpha} * mentOlp(A, B)^{(1-\alpha)},$$

where $attrMatch(A, B) \in [0, 1]$ is computed as an average of how well each of the three attributes (source name, topic name and polarity) of the two summary opinions match[1]. $mentOlp(A, B)$ is a measure of how well fine-grained opinions that make up the summary opinion are extracted. We borrow Luo's function (2005) and set $mentOlp(A, B) = (2*|A \cap B|)/(|A|+|B|)$. Lastly $\alpha \in [0, 1]$ is a parameter that controls how much weight is given to identifying correctly the attributes of summary opinions vs. extracting all fine-grained opinions.

The α parameter allows us to tailor the OSEM score toward either type of opinion summary. When $\alpha = 0$ (we will use $OSEM_0$ to refer to the OSEM score with $\alpha = 0$) the OSEM score reflects only how well the response groups together fine-grained opinions from the same source **and** on the same topic and makes no reference to the attributes of summary opinions. Thus, this value of α is suitable to evaluating opinion set summaries. Note that $OSEM_0$ bears similarity to the DLB^3 score. The difference is that DLB^3 looks in isolation at clusterings for source and topic and computes the average of the two while $OSEM_0$ looks only at complete clusters of fine-grained opinions on the same source **and** the same topic.

On the other hand, $OSEM_1$ ($\alpha = 1$) puts all weight on how well the attributes of each summary opinion are extracted, which is suitable for evaluating aggre-

[1]There are other ways of combining the scores for the three attributes, especially if the aggregate opinion is considered a relation between the opinion source and the topic. For instance, a combination based on multiplication instead of average would guarantee that a response opinion receives a non-zero score only when all attributes match partially the corresponding attributes in the key opinion. We selected the average because it is more lenient and because it is used in other information extraction tasks that include scoring filled templates.

gate opinion summaries. It should be noted, however, that OSEM_1 does not require summary opinions to be connected to any fine-grained opinions in the text. This can lead to inconsistent summaries getting undeserved credit. For instance, in the example of Figure 1.1 a system could recognize that the text mentions "Bush" and "American public" and infer that there is a neutral opinion from Bush toward the American public. OSEM_1 will give partial credit to such a summary opinion when compared to the negative opinion from Bush toward Al Qaeda, for example. At any other value ($\alpha < 1$) the *mentOlp* for such an opinion will be 0 giving no partial credit for opinions that are not grounded to a fine-grained opinion in the text.

The influence of the α parameter is studied empirically in the next chapter, which also gives an example of the computation of the OSEM score.

6.4 Chapter Summary

We devoted this chapter to discussion of evaluation metrics that can be used to quantitatively judge the quality of complete opinion summaries as compared to a gold-standard summary. We began with a discussion of three existing scoring algorithms for tasks that can be considered similar in different ways to opinion summarization. We then briefly discussed the requirements for what makes for a "good" summary for each of the two summary types that we propose. Finally, we presented two novel scoring algorithms: Doubly-linked B^3 (DLB^3) score, which is suitable for evaluating opinion summaries of the summary set form and Opinion Summary Evaluation Metric (OSEM), which is a hybrid evaluation metric that can be used for both kind of summaries.

CHAPTER 7

GENERATING AND EVALUATING COMPLETE OPINION SUMMARIES

In this chapter we put all components together to introduce OASIS (for Opinion Aggregation and SummarIzation System), the first system known to us that can produce rich non-extract-based domain-independent opinion summaries. The system relies on automatically extracted fine-grained opinion information and constructs fully automatic opinion summaries in the forms that we suggested in Chapter 1.

Unlike most extract-based summarization tasks, we are happily able to generate gold standard summaries for comparison with the automatic summaries. As a result, our evaluation requires no human intervention to judge overlap with a manually generated gold standard summary. Our results are encouraging — OASIS substantially outperforms a competitive baseline when creating document-level *aggregate summaries* (like the one in Figure 1.1) that compute the average polarity value across the multiple opinions identified for each source about each topic. We further show that as state-of-the-art performance on fine-grained opinion extraction improves, we can expect to see opinion summaries of very high quality — with F-scores of 54-77% using our OSEM evaluation measure.

We begin this chapter by describing the architecture of our system, OASIS, giving details for the different subsystems that we use and their accuracy. We then describe the results of an empirical evaluation of OASIS that we perform using the aforementioned $\text{MPQA}_{\text{TOPIC}}$ corpus.

7.1 OASIS

OASIS employs a pipelined architecture, which relies on four steps. Below we describe each step in more detail.

7.1.1 Fine-grained Opinion Extraction

As discussed in Chapter 2, a number of research efforts have addressed extracting fine-grained opinions and their attributes. OASIS builds on this work by using two previously developed fine-grained opinion extractors.

Our system starts with the predictions of Choi et al.'s (2006) opinion source and opinion trigger extractor. The extractor works by combining a source extraction identifier (described in Choi et al. (2005)) and an opinion trigger classifier from Breck et al. (2007) by explicitly considering the linking relation between sources and opinion triggers. In addition, Choi et al. employ semantic role labeling to arrive at a system that achieves F-measures of 79 and 69 for entity and relation extraction, respectively.

Predictions of the Choi et al.'s system can be described as a tuple [opinion trigger, source], where each of the two components represents a span of text in the original document signaling the expression of opinion and a reference to the opinion source, respectively. We enhance these fine-grained opinion predictions by using an opinion polarity classifier from Yessenalina and Cardie (2009), which adds polarity predictions as one of three possible values: *positive*, *negative* or *neutral*. This value is added to the opinion tuple to obtain [opinion trigger, source, polarity] triples. The fourth element of fine-grained opinions, the topic, is incorporated later during the topic coreference step.

7.1.2 Source Coreference Resolution

Given the fine-grained opinions, our system decides which opinions should be attributed to the same source, i.e., performs source coreference resolution. For this task, we use the partially supervised learning approach described in Chapter 4. As a result of this step, OASIS produces opinion triples grouped according to their sources.

7.1.3 Topic Extraction/Coreference Resolution

Next, our system has to label fine-grained opinions with their topic and decide which opinions are on the same topic. Here, we use the topic coreference resolution described in Chapter 5. As a result of this step, OASIS produces opinion four-tuples [opinion trigger, source, polarity, topic name] that are grouped both according to their source and their topic. This four-tuple constitutes an opinion set summary as described in Section 1.3.

7.1.4 Aggregating Multiple Opinions

After obtaining the opinion set summary for a document, OASIS can create an aggregate opinion summary like the one described in Section 1.3. This requires a means for combining the multiple (possibly conflicting) opinions from a source on the same topic that appear in the opinion set summary. There are several different sensible ways to combine multiple opinions from the same source that are about the same topic relevant to different application needs. Here we discuss several such methods, but for the purpose of evaluation in next section we use only one such method, **average opinion**, described below. Incorporating other ways to combine opinions in OASIS is a straightforward process, but in

the absence of practical application needs, we avoid adding the complexity of evaluation in the context of different aggregation methods. Preliminary experiments revealed that results using other aggregation methods are comparable.

The most straightforward way to merge opinions is **average opinion**: the polarity of the opinion set is an average of the polarity of all the opinions from the source on the topic. This method for computing the overall opinion is likely to be useful for applications that value capturing the overall trend. As noted above, **average opinion** is the default opinion aggregation method for OASIS and is used for evaluation.

Another way to compute overall opinions is **conflicting opinion**, which characterizes the set of opinions into one of four polarity classes: *positive, negative, neutral and mixed*. If a source expresses only positive and neutral opinions on a topic, then the overall polarity is *positive* (likewise for negative). If a source expresses both positive and negative opinions, then the polarity is *mixed*. If all opinions are neutral, then the overall polarity is *neutral*. The **conflicting opinion** method is likely to be useful for applications that need not only the overall trend, but need information on whether any conflicting opinions are expressed.

There are other ways to aggregate opinions such as only keeping mixed opinions, only showing negative opinions, or classifying opinions by their strength and keeping only the strongest opinions. Again, we expect that the method for computing overall opinions will be dictated by the application needs.

Performance of the different subcomponents of our system as it applies to

Table 7.1: Performance of components of the opinion summarization system.

Component	Measure	Score
Fine-grained opinion extractor	F1	59.7
Polarity classifier	Accuracy	65.3
Source coreference resolver	B^3	83.2
Topic coreference resolver	B^3	54.7

our data (see Section 7.2) are shown in Table 7.1. F1 refers to the harmonic average of precision and recall, while the B^3 evaluation metric for coreference resolution (Bagga and Baldwin, 1998) is described in Chapter 6. Our scores for fine-grained opinion extraction are lower than the published results (Choi et al., 2006) because we do not allow the system to extract speech events that do not signal expressions of opinions (i.e. the word "said" when used in objective context: "John said his car is blue.").

7.2 Experimental Evaluation

For evaluation we use the aforementioned MPQA (Wiebe et al., 2005b) and $\text{MPQA}_{\text{TOPIC}}$ (Stoyanov and Cardie, 2008b) corpora.[1] As a reminder, the 1.2 version of the MPQA corpus consists of 535 documents from the world press, manually annotated with phrase-level opinion information following the annotation scheme of Wiebe et al. (2005b). In particular, the corpus provides annotations for opinion expressions, their polarities, and sources as well as source coreference. The $\text{MPQA}_{\text{TOPIC}}$ corpus consists of 150 documents from the MPQA corpus, which are also manually annotated with opinion topic information, including topic spans, topic labels, and topic coreference. Since the MPQA corpus is a

[1]The MPQA corpus is available at `http://nrrc.mitre.org/NRRC/publications.htm`.

It is unlikely that the Vatican will establish diplomatic ties with mainland China any time soon, judging from their differences on religious issues, Ministry of Foreign Affairs (MOFA) spokeswoman [$_{\text{Source}}$ Chang Siao-yue] [$_{\text{neu}}$ *said*] Wednesday.
[$_{\text{Source}}$ Chang]'s [$_{\text{neu}}$ *remark*] came in response to a foreign wire [$_{\text{neu}}$ *report*] that mainland China and the Vatican are preparing to bridge their differences and may even pave the way for full diplomatic relations.
[$_{\text{Source}}$ Beijing authorities] are [$_{\text{neu}}$ *expected*] to take advantage of a large religious meeting slated for October 14 in Beijing to develop the possibility of setting up formal relations with the Vatican, [$_{\text{neu}}$ *according*] to the report.
...
[$_{\text{Source}}$ The MOFA spokeswoman] [$_{+}$ *affirmed*] that from the angle of Eastern and Western cultural exchanges, the sponsoring of similar conferences will be instrumental to [$_{\text{Source}}$ mainland Chinese people]'s [$_{+}$ *better understanding*] of Catholicism and its contributions to Chinese society.
As for the development of diplomatic relations between mainland China and the Vatican, [$_{\text{Source}}$ Chang] [$_{-}$ *noted*] that differences between the Beijing leadership and the Holy See on religious issues dates from long ago, so it is impossible for the Vatican to broach this issue with Beijing for the time being.
[$_{\text{Source}}$ Chang] also [$_{+}$ *reaffirmed*] the solid and cordial diplomatic links between the Republic of China and the Vatican.

KEY SUMMARY:

♯ source	opinion	topic
k1. Chang Siao-yue	**neutral**	diplomatic links
	said remark noted reaffirmed	
k2. foreign wire	**neutral**	diplomatic links
	report according to	
k3. Chinese people	**positive**	Catholicism
	better understanding	
k4. Chang Siao-yue	**positive**	conferences
	affirmed	
k5. author	**neutral**	Beijing authorities
	are expected	

RESPONSE SUMMARY:

♯ source	opinion	topic
r1. Chang Siao-yue	**positive**	pave bridge vatican
	said remark noted reaffirmed	
r2. MOFA spokeswoman	**positive**	sponsor conference
	affirmed	Catholicism
r3. Chinese people	**neutral**	sponsor conference
	better understanding	Catholicism
r4. Beijing authorities	**neutral**	Beijing authorities
	are expected	

Figure 7.1: An opinion summary produced by OASIS. The example shows the original article with gold-standard fine-grained opinion annotations above, the key opinion summary in the middle and the summary produced by OASIS below.

general, domain-independent corpus consisting of documents from the world press, we believe that the results that we obtain are representative of other similar domain-independent corpora including those that use different definitions of opinion as described in Section 2. As mentioned elsewhere in the thesis, summarization of product reviews have different characteristics and are amendable to different approaches and, thus, we do not expect results described in this thesis to be representative of such corpora.

Our gold-standard summaries are created automatically for each document in the $\text{MPQA}_{\text{TOPIC}}$ corpus by relying on the manually annotated fine-grained opinion and source- and topic-coreference information. This constitutes our test set for the experiments below. For our experiments, all components of OASIS are trained on the 407 documents in the MPQA corpus that are not part of the $\text{MPQA}_{\text{TOPIC}}$ corpus, with the exception of topic coreference, which is trained on the $\text{MPQA}_{\text{TOPIC}}$ corpus using 5-fold cross-validation.

7.2.1 Example

We begin our evaluation section by introducing an example of an output summary produced by OASIS. The top part of Figure 7.1 contains the text of a document from the $\text{MPQA}_{\text{TOPIC}}$ corpus, showing the fine-grained opinion annotations as they are marked in the MPQA corpus. The middle part of Figure 7.1 shows the gold-standard summary produced from the manual annotations. The summary is shown as a table with each box corresponding to an overall opinion. Each opinion box shows the source name on the left (each opinion is labeled with a unique string, e.g., $k1$ for the first opinion in the key) and the topic name on the right (string equivalence for the source and topic name indicate the same

source/topic for the purpose of the example). The middle column of the opinion box shows the opinion characterized by the computed overall opinion shown in the first row and all opinion mentions that were combined to produce the overall opinion shown in subsequent rows (for the purpose of presentation mentions are shown as strings, but in reality they are represented as spans in the original text by the summaries). Finally, the summary produced by OASIS is shown in the bottom part of Figure 7.1 following the same format.

OASIS performed relatively well on the example summary of Figure 7.1. This is partially due to the fact that most of the opinion mentions were identified correctly. Additionally, source coreference and topic coreference appear to be mostly accurate, but there are several mistakes in labeling the topic clusters as compared to the gold standard. Generally, manual review of the results of OASIS show that the overall results exhibit somewhat similar trends. Automatic identification of opinions is responsible for a fair number of mistakes that propagate through source and topic coreference resolution and affect aversely the quality of the summaries. Source coreference resolution and source name labeling is correct on many occasions, while topic coreference is often correct, but assignment of topic names is often wrong. The overall accuracy appears sufficient for many tasks that rely on opinion attributes.

Next, we use the example of Figure 7.1 to illustrate the computation of the OSEM score. The first step of computing the score is to calculate the scores for how well each response opinion matches each key opinion. The four-by-five matrix of scores for matching response opinions to key opinions is shown in Table 4.1. Scores in the table are computed for value of the α parameter set to .5. As discussed in the previous section, all values of $\alpha < 1$ require that key and

Table 7.2: $OSEM_{.5}$ score for each response opinion as matched to key opinions in the example summary of Figure 7.1.

	k1	k2	k3	k4	k5
r1	.58	0	0	0	0
r2	0	0	0	.81	0
r3	0	0	.71	0	0
r4	0	0	0	0	.81

Table 7.3: $OSEM_{1.0}$ score for each response opinion as matched to key opinions in the example summary of Figure 7.1.

	k1	k2	k3	k4	k5
r1	.33	0	.33	.67	0
r2	0	0	.33	.50	0
r3	.33	.33	.50	.16	.33
r4	.33	.33	0	0	.67

response opinions have at least one mention in common to receive a non-zero score. This is illustrated in Table 4.1, where only four of the 20 match scores are greater than 0.

Based on the scores in Table 7.2, the optimal match between key and response opinions is $r1 \rightarrow k1$, $r2 \rightarrow k4$, $r3 \rightarrow k3$, and $r4 \rightarrow k5$. The value of this score is 2.91, which translates to $OSEM_{.5}$ precision of .73 and recall of .58 for an overall $OSEM_{.5}$ F-score of .65.

Finally, to illustrate the different implications for the score when the α parameter is set to 1, we show the match scores for $OSEM_1$ in Table 7.3. Note that there are far fewer 0 scores in Table 7.3 as compared to Table 7.2. In the case of this particular summary, the optimal matching between key and response opinions is the same as for the setting of $\alpha = .5$, but this is not always the case. The $OSEM_1$ precision, recall and F-score for this summary are .50, 60 and .55,

Table 7.4: Scores for the summary system with varying levels of automatic information.

Fine-grained opinions	System	DLB^3	OSEM				
			$\alpha = 0$	$\alpha = .25$	$\alpha = .5$	$\alpha = .75$	$\alpha = 1$
Automatic	Baseline	29.20	50.78	37.32	27.90	21.12	25.47
	OASIS	31.24	49.75	41.71	35.82	31.52	41.50
Manual	Baseline	51.12	78.67	60.72	47.04	36.60	28.59
	OASIS	59.82	78.69	69.04	61.47	55.59	54.80
	OASIS + manual src coref	79.85	82.65	79.39	76.68	74.61	74.95
	OASIS + manual tpc coref	80.80	82.40	78.14	74.53	71.56	71.03

Table 7.5: OSEM precision, recall and F-score as a function of α.

α	0.00	0.01	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	0.99	1.00
OSEM prec	51.5	50.9	47.8	44.6	41.8	39.3	37.1	35.2	33.5	32.0	30.7	29.6	42.8
OSEM recall	48.1	47.6	44.7	41.7	39.0	36.7	34.6	32.8	31.2	29.7	28.5	27.5	40.3
OSEM F1	49.8	49.2	46.2	43.1	40.4	38.0	35.8	33.9	32.3	30.8	29.5	28.5	41.5

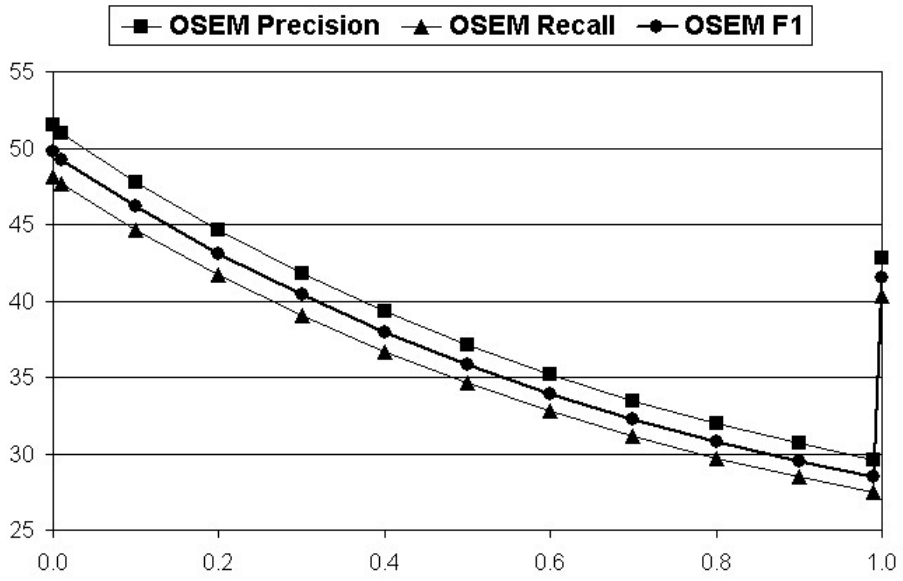

Figure 7.2: OSEM precision, recall and F-score (x-axis) vs. α (y-axis).

respectively.

Next, we discuss the empirical performance of OASIS on the $\text{MPQA}_{\text{TOPIC}}$ cor-

pus. We begin by presenting the baseline to which we compare our results.

7.2.2 Baseline

We compare our system to a baseline that creates one summary opinion for each fine-grained opinion. In other words, each source and topic mention is considered unique and each opinion is in its own cluster.

7.2.3 Results

Results are shown in Table 7.4. We compute DLB^3 score and OSEM score for 5 values of α chosen uniformly over the $[0, 1]$ interval. The top two rows of Table 7.4 contain results for using fully automatically extracted information.

Compared to the baseline, OASIS shows little improvement when considering opinion set summaries (DLB^3 improves from 29.20 to 31.20, while OSEM_0 worsens from 47.67 to 46.54). However, as α grows and more emphasis is put on correctly identifying attributes of summary opinions, OASIS substantially outperforms the baseline (OSEM_1 improves from 24.01 to 38.95).

Next, we try to tease apart the influence of different subsystems. The bottom four rows of Table 7.4 contain system runs using gold-standard information about fine-grained opinions (i.e., the [opinion trigger, source, polarity] triple). Results indicate that the quality of fine-grained opinion extractions has significant effect on overall system performance – scores for both the baseline and OASIS improve substantially. Additionally, OASIS appears to improve more compared to the baseline when using manual fine-grained opinion information. The last two rows of Table 7.4 show the performance of OASIS when using manual

information for source and topic coreference, respectively. Results indicate that the rest of the errors of OASIS can be attributed roughly equally to the source and topic coreference modules.

Lastly, the OSEM score is higher at the two extreme values for α (0 and 1) as compared to values in the middle (such as .5). To study this anomaly, we compute OSEM scores for 13 values of α. Results, shown in Table 7.5 and Figure 7.2, indicate that the OSEM score decreases as more weight is put on identifying attributes of summary opinions (i.e., α increases) with a discontinuity at $\alpha = 1$. We attribute this discontinuity to the fact that OSEM_1 does not require opinions to be grounded in text as discussed in Section 6.3.2. Note, however, that the $\alpha = 1$ setting is akin to the standard evaluation scenario for many information extraction tasks.

7.3 Chapter Summary

In this chapter we introduced OASIS, our end-to-end completely automatic system that produces opinion summaries such as the ones that we propose. OASIS is the first system known to us that can produce rich non-extract-based opinion summaries from general text. We began by describing the pipelined architecture of OASIS, discussing each step in detail. We then described the results of an empirical evaluation of OASIS that we performed using the $\text{MPQA}_{\text{TOPIC}}$ corpus. Results are promising – OASIS outperforms a competitive baseline by a large margin when we put more emphasis on computing an aggregate summary.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

In this dissertation, we have addressed the problem of summarizing fine-grained opinion information extracted from text. This chapter summarizes the contributions of our research and outlines directions for future work.

8.1 Summary of Contributions

The main contribution of this thesis is that it constitutes the first work that addresses the problem of creating non-extract-based opinion summaries for domain-independent fine-grained opinions. To this end, this dissertation contains the first in-depth discussion of the form of domain-independent opinion summaries and identifies the research problems involved in creating opinion summaries. More importantly, the thesis addresses the identified research problems concerning the creation of opinion summaries:

Usability Study. Like other work in the area of fine-grained sentiment analysis, our work is based on the hypothesis that fine-grained opinion information can be used successfully in NLP applications. While previous work has argued in favor of this hypothesis, this conjecture has been supported by little empirical evidence. This thesis includes one of the first experimental studies that shows empirically that fine-grained opinion information can be useful for an NLP application, Multi-Perspective Question Answering (MPQA).

More precisely, we presented the OpQA corpus of opinion questions and answers. Using the corpus, we compared and contrasted the properties of fact and opinion questions and answers. We found that text spans identified as

answers to opinion questions: (1) are approximately twice as long as those of fact questions, (2) are much more likely (37% vs. 9%) to represent partial answers rather than complete answers, (3) vary much more widely with respect to syntactic category – covering clauses, verb phrases, prepositional phrases, and noun phrases; in contrast, fact answers are overwhelming associated with noun phrases, and (4) are roughly half as likely to correspond to a single syntactic constituent type (16-38% vs. 31-53%).

Based on the disparate characteristics of opinion vs. fact answers, we argued that traditional fact-based QA approaches may have difficulty in an MPQA setting without modification. We proposed, instead, that MPQA systems should rely on fine-grained information about opinions. In experiments in opinion question answering using the OpQA corpus, we found that filtering potential answers using filters based on automatically identified fine-grained opinion information substantially improves the performance of an end-to-end MPQA system according to both a mean reciprocal rank (MRR) measure (0.59 vs. a baseline of 0.42) and a metric that determines the mean rank of the first correct answer (MRFA) (26.2 vs. a baseline of 61.3). Further, we found that requiring opinion answers to match the requested opinion source dramatically improved the performance of the MPQA system on the hardest questions in the corpus.

Source Coreference Resolution. One of the steps in opinion summarization includes linking together opinions that belong to the same source – *source coreference resolution*. This thesis includes the first approach to the problem of source coreference resolution. In particular, we defined and treated source coreference resolution as a partially supervised version of noun phrase coreference resolution. The partially supervised nature of the problem led us to approach it as the

more general, but also novel, problem of partially supervised clustering. We proposed an algorithm for partially supervised clustering that extends a rule learner with structure information and is generally applicable to problems that fit the partially supervised clustering definition. We applied the algorithm to the source coreference resolution task and evaluated its performance on the MPQA corpus (Wiebe et al., 2005b). We found that our algorithm outperforms highly competitive baselines by a considerable margin - B^3 score of 83.2 vs. 81.8 and 67.1 vs. 60.9 F1 score for the identification of positive source coreference links.

Topic Identification. Topic identification has received little research attention due to both the difficulty of the task and the lack of appropriately annotated resources. This thesis addresses the problem of topic identification for fine-grained opinion analysis of general text. We provided a new, operational definition of *opinion topic* in which the topic of an opinion depends on the context in which its associated opinion expression occurs. We also presented a novel method for both manual and automatic general-purpose opinion topic identification that, following our new definition, treats the problem as an exercise in topic coreference resolution. We created the $\text{MPQA}_{\text{TOPIC}}$ corpus by adding manual annotations that encode topic information to part of the MPQA corpus (Wiebe et al., 2005b) and used the $\text{MPQA}_{\text{TOPIC}}$ corpus for evaluation.

Our empirical evaluation showed that inter-annotator agreement results for the manual annotations are reasonably strong across a number of metrics and the results of experiments that evaluate our topic identification method in the context of fine-grained opinion analysis are promising: using either automatically or manually identified topic spans, we achieve topic coreference scores that statistically significantly outperform two topic segmentation baselines across

three coreference resolution evaluation measures (B^3, α and CEAF). For the B^3 metric, for example, the best baseline achieves a topic coreference score on the $\text{MPQA}_{\text{TOPIC}}$ corpus of 0.55 while our topic coreference algorithm scores 0.57 and 0.71 using automatically, and manually, identified topic spans, respectively.

Evaluation Measures. There are no "natural" evaluation metrics that quantitatively assess the quality of an automatically generated opinion summary as compared to a gold standard. Additionally, we are not aware of any previous work that has suggested evaluation metrics for structures such as those of the opinion summaries. To address these problems, we proposed two evaluation metrics for opinion summaries inspired by evaluations in information extraction and noun phrase coreference resolution. These evaluation metrics allow us to quantitatively compare the output of different systems to a gold-standard summary.

Generating and Evaluating Complete Opinion Summaries. To the best of our knowledge, this thesis contains the first published work that generates and evaluates rich domain-independent non-extract-based opinion summaries. We presented our system, OASIS. OASIS relies on a pipelined architecture and combines two fine-grained opinion extraction systems (Choi et al., 2006; Yessenalina and Cardie, 2009) and the methods for source and topic coreference resolution presented in this thesis plus an opinion aggregation step employed when generating aggregate summaries. We evaluated empirically the performance of OASIS and found out that it substantially outperforms a competitive baseline when creating document-level *aggregate summaries* (like the one in Figure 1.1) that compute the average polarity value across the multiple opinions identified for each source about each topic. We further showed that as state-of-the-art

performance on fine-grained opinion extraction improves, we can expect to see opinion summaries of very high quality – with F-scores of 54-77% using our OSEM evaluation measure.

8.2 Future Work

There are numerous avenues to extend our work in the future. Below, we briefly describe future work pertaining to two of the problems that we addressed in this thesis, source coreference resolution and topic identification, followed by a discussion of future work for the overall problem of opinion summarization.

Source Coreference Resolution. As previously noted, we approach source coreference resolution as the novel problem of partially supervised clustering. A limitation of our method for partially supervised clustering is that we do not directly optimize for the performance measure (e.g. B^3). Other efforts in the area of supervised clustering (e.g. Finley and Joachims (2005), Li and Roth (2005)) have suggested ways to learn distance measures that can optimize directly for a desired performance measure. We plan to investigate algorithms that can directly optimize for complex measures (such as B^3) for the problem of partially supervised clustering. Unfortunately, a measure as complex as B^3 makes extending existing approaches far from trivial due to the difficulty of establishing the connection between individual pairwise decisions (the distance metric) and the score of the clustering algorithm.

Topic Identification. As noted in Chapter 5 our approach to topic identification is the first known to us. Therefore, there are still many ways in which our work can be extended.

One of the conclusions of the empirical evaluation in Chapter 5 was that identifying precisely the topic spans is important for the overall performance of the topic identification system. Currently our system uses manual rule-based methods for topic span extraction. An obvious extension to this approach is to use a machine learning approach for extracting the topic span. Such a machine learning approach could benefit from using semantic role labeling (e.g. Gildea and Jurafsky (2001)). Often, opinions in the MPQA corpus and in text in general are expressed by the means of reporting verbs. In most of these cases, the topic (or target) span constitutes the argument occupying a specific semantic position in relation to the reporting verb. Using semantic role labeling, we could learn from a training corpus the positions of the topic span argument for different classes of verb and adjective predicates and use those for topic span identification.

Additionally, opinion topic identification could benefit from approaches to discourse modeling of the topic of text such as those building on theories of focus and centering (e.g. Reinhart (1982), Grosz et al. (1995), Traum et al. (1996), Singh et al. (2002)). These approaches can be incorporated within our method for topic coreference resolution as a way of tracking when the topic of discourse changes. Due to some differences in the way topics of opinions are expressed (e.g. they are not always sequentially coherent), discourse modeling approaches are unlikely to be successful on their own.

Finally, our approach to topic identification assumes that each opinion is about a single topic. An alternative approach is to consider opinions concerning multiple topics – i.e., an opinion can be considered to be about all entities or events that are mentioned directly or indirectly in the topic span or could be

otherwise inferred. We consider this *multiple topic* approach complementary to ours as it is likely to be relevant to different applications. We chose an approach based on the concept of a single topic for each opinion because of its relevance to our task and because it is easier to operationalize both in terms of manual annotation as well as automatic evaluation. In the future, we plan to explore the multiple topic approach.

Opinion Summarization. Above we discussed different ways in which the opinion summarization process can be improved through the use of different approaches for two of the underlying tasks. Here we discuss three extensions concerning the overall opinion summaries.

An important premise of our research is that the opinion summaries that we produce are a useful representation both for end-user consumption as well as for use in applications that incorporate information about opinions. In the future, we would like to validate this premise by: (1) creating graphical user interfaces (GUIs) that allow opinion summaries to be presented to a user in a way that is easy to browse and manipulate, and, (2) by incorporating the output of OASIS in applications such as multi-perspective question answering (MPQA) and empirically evaluating the influence of using summarized vs. raw fine-grained opinion information.

In terms of practical applications of our system, we argued in Chapter 2 that OASIS can benefit from a component that produces a textual summary from the graphical representation that we propose and utilize. This can be done by incorporating a natural language generation component (e.g. Reiter and Dale (2006)) that is aware of the structure of the graphical opinion summary that OASIS produces.

Finally, the summaries that we present and discuss are computed over a single document. Extending our work to allow the computation of summaries over a set of documents has a particular practical appeal. To achieve this extension we need to formulate approaches for cross-document source and topic coreference resolution.

APPENDIX A

INSTRUCTIONS FOR DEFINING "FACTOID" AND "OPINIONOID" QUESTIONS

This appendix contains the instructions that were used to create questions for the OpQA copus described in (Stoyanov et al., 2004; Stoyanov et al., 2004). The original title of the publication was INSTRUCTIONS FOR DEFINING AND IDENTIFYING "FACTOID" AND "OPINIONOID" QUESTIONS AND ANSWERS: SHORT-ANSWER QUESTIONS THAT ARE ANSWERED EXPLICITLY IN A TEXT. The instructions were created in November of 2002 by Janyce Wiebe, Diane Litman and Claire Cardie. Only the question creation part of the instructions was used for corpus creation. The rest of this appendix contains the directions verbatim.

A.1 Introduction

Today's search systems take a question and return a list of documents likely to contain an answer to the question. The user of the system must then read the documents to find the desired answer within them, if it's there. This can be a very tedious, time-consuming, and frustrating process. It would be better if the system returned smaller pieces of text - a few words or a sentence believed to contain the answer. Then the user would have less reading to do in order to see if any of the pieces contained an answer.

Such improved systems exist today in experimental versions, for certain types of questions ("factoids"). In order to know how good a job such improved search systems are doing we need to judge whether, given a question, these systems return pieces of text that are responsive to the question (i.e., you can

recognize the answer in the piece).

Your task will be to provide a set of questions and answers, that we will use to both evaluate how well current experimental systems can answer factoid questions, and to extend the state-of-the-art so that such systems can also handle a new type of question ("opinionoids").

A.2 Factiod questions and answers

A.2.1 Writing the questions

The "factoid" questions that we would like you to write are fact-based, short-answer questions such as "How many calories are there in a Big Mac?" Thus, factoid questions should have some definite answer (typically a noun phrase as opposed to a procedural answer). We also request that your questions do not require a compound answer (e.g., a list of items). Try not to have your question be an extremely contrived back-formulation of a statement in the document.

A.2.2 Identifying the answers

For each of the questions that you have defined above, you also need to provide a set of "answer strings". For our purposes, an answer string is a piece of text from a document that contains some words that answer the question. Each answer string MUST be wholly contained in a single sentence. In other words, the answer string should appear explicitly in the text, contained within a single sentence. Note that explicit means that the answer string need not contain the same words as used in the question, but that you should not need to bring in extra background knowledge to interpret the string as an answer. There should

be at least one document in your collection that contains an answer to your question.

To identify the answers for your questions, please execute the following procedure, for EACH question:

1. Read your question carefully.
2. Find all the answer strings by skimming through a subset of the documents that you have been given (we will contact you directly on this, to tell you which subset in particular), and identifying each piece of text ("the answer string") that contains a valid answer to the question. The answer string does not need to contain justification for the answer (although it optionally can). The answer string can be part of a single sentence; furthermore, it can be grammatically incorrect and might even contain word fragments. However, the answer string can NOT be longer than a whole sentence. In some cases, the context (i.e. sentence) in which a (proposed) answer string occurs interferes with recognizing the answer. In these cases, you should decide if the interference is severe enough to omit the string from consideration as an answer string.

You should construct your answer strings such that if the answer string were returned alone to a trustful user of a question-answering system, the user would be able to get the correct answer to the question. There should be no need for the system to provide justification in its answer string.

Some Special Cases to Note Note that if an answer string can only be inferred after pronoun resolution across sentences, then it technically does not count as

an answer using the above definitions. For example, imagine trying to answer the question ”What is the name of our national library?” given a document that contains the sentences ”But the Library of Congress was built for all the people. From the start, it was our national library.” Although the correct answer to the question is ”Library of Congress”, returning only one of the 2 sentences is insufficient by itself to answer the question. This is because the pronoun ”it” must be resolved across these sentences to determine the correct answer. For the purposes of this study, we will allow you to include some answer strings that contain pronouns, as long as you also include a significant number of answer strings that do not require this type of inference.

In contrast, when candidate answer strings to factoid questions appear in subjective contexts (e.g., opinions), then such strings should NOT be returned as answers. For example, consider the question ”What is the capital of New Jersey?” If the document only contains the sentence ”John thinks that Trenton is New Jersey’s capital”, this sentence does not answer the question. However, if the document instead said ”Trenton is the capital of New Jersey”, then this sentence, in particular ”Trenton”, would indeed be the correct answer string for the question.

Other Notes - For a single question, it is possible that there may be more than one answer string in your document collection.

- Construct your answers with respect to the context of each document. Thus, even if a document gives an answer that you believe is wrong, create your answer based on what the document says.

- You may decide that it is reasonable to provide a ”partial” answer, e.g.,

accepting a last name as an acceptable answer for a ”who” question.

See Appendices I and III for example answers (to the factoid questions in Appendices I and II) that have been used in previous evaluations of experimental question-answering systems.

A.3 Opinionoid questions and answers

In contrast to factoid questions, the answers to opinionoid questions involve opinions, evaluations, judgments, emotions, sentiments, or speculations (the general term is ”private state”). Since answering opinionoid questions automatically has never been attempted before, we do not want to be overly ambitious . We want to target the clearest cases first.

For example, consider the following sentence from Pravda.

”Vadim Orlav told Ulyanovsk journalists that the referendum was celebrated by the people in Iraq with festivals, concerts, shows, singing, and dancing”.

We could consider this sentence to be answers to questions such as,

”How do the Iraqi people feel about the referendum?”

”Who is positive toward the referendum?”

For question 1, the answer string could even be as small as ”the referendum was celebrated by the people in Iraq”, while for question 2 the answer string could be as small as ”the people in Iraq”.

Note that as with factoid questions, the answer string for an opinionoid question should appear explicitly in the text, contained within a single sentence. However, some answer strings to opinion-oriented questions might require some (limited) amount of inference to recognize them as an answer. For example, recognizing that "the referendum was celebrated by the people in Iraq" answers question 1 above requires a small amount of inference, e.g., recognizing that celebrated is a positive feeling.

The question need not specify whose private state is being presented, just as long as a human could determine that by looking at the sentence. For example, answers 1-3 are fine answers to the given question:

question: "Was the referendum conducted properly?"

answer 1: "The Major-General advised that the referendum was organized very well" (Opinion of the Major-General, according to the writer).

answer 2: "The referendum was organized very well" (Opinion of the writer)

answer 3: "The referendum was like a smooth-running machine," said the Major-General.

And again, as with factoids, it's fine if there are multiple different answers to a question (in a single text or across a set of texts).

How specific should the questions be? The automatic system will sometimes be tested on different documents than the ones given to you to develop the questions; however, the documents provided to the automatic system will be on the same general topic and from the same period of time. We want questions that are general enough to apply to more than one document on the topic, i.e.

the questions shouldn't be too specific, asking for details not likely to appear in other documents.

In general, we would like your question to be phrased as something that you might have asked, if you hadn't seen the document first. Here are some suggestions on how to make it easier to do this. Once you have a question, make sure that you can find answers to the same question in other documents on the same topic. If there are no answers in other documents, think about how your question could be rephrased to have multiple answers. Questions satisfying this constraint should be less likely to be "back-formulations" of specific sentences. Or conversely, try looking at at least two documents while you are developing your questions to begin with. Another general guideline is to phrase your questions in such a way that you could imagine someone else having asked such questions, without having seen the document/answers in advance.

One thing we are NOT targeting with the opinionoid questions are situations in which people have different factual beliefs. Suppose that Text A objectively states that there are 100,000 troops in Chechnya, and Text B objectively states that there are 50,000 troops in Chechnya, and you believe there are 10,000 troops in Chechnya. This does not make "How many troops are there in Chechnya?" an opinionoid question. If the only reason you think of something as an opinion is because it contrasts with a conflicting fact in another text (or in your own mind), then that is not the type of opinion we are targeting. (Note that "How many troops are there is Chechnya?" is a good factoid question, which in the above scenario has two different answers, 100,000 from Text A and 50,000 from Text B).

APPENDIX B

INSTRUCTIONS FOR ANNOTATING ANSWERS TO MULTI-PERSPECTIVE QUESTIONS

This appendix contains the instructions for the manual annotation of answers to Multi-Perspective Questions are presented in this chapter. These instructions were used for the creation of the OpQA corpus. The instructions are included here verbatim starting on the next page.

Instructions for Annotating Answers for Multi-Perspective Question Answering

Introduction

The research question of Multi-Perspective Question Answering (MPQA) is to discover efficient algorithms that can accomplish the task of answering questions about beliefs, opinions, and evaluations embedded in natural language texts. For the purpose of evaluating such systems we need a collection of sample documents together with a set of questions. In addition, for each question we need information about what parts of each document constitute an answer. Traditionally, question answering (QA) collections have used two kinds of information about what constitutes an answer to the questions in the collection: the textual form of the answer and/or the segments of documents in the collection that can constitute an answer. Information about what segments in the documents constitute (or can contribute to) an answer to each of the questions is largely used in evaluation of QA systems for at least two reasons: firstly, it can help assure that the system has found a real answer to the question as opposed to being lucky in picking out the correct answer in the wrong context; and, secondly, it can give credit to a system for finding the place in the collection where a question is answered although the system may not be able to convert the string to the exact string representation required for the answer (e.g. answers to yes/no questions).

For the task of MPQA, the latter reason for relying on information about the text segments that answer questions is even more important considering that it is much harder to convert an answer segment to an exact answer string even when the answering segments are found in the text. Therefore, we will augment the MPQA collection by adding annotations (marking up) to every text segment that can contribute to the answer for any question in the collection. This document explains what and how to annotate, as well as the idea behind the annotations.

We will use an annotation of type ANSWER to designate text segments that constitute answers. An ANSWER annotation will be represented as an xml marking in the source document and will contain five attributes: *question topic, question number, confidence, confidence comment,* and *partial answer*. You will use the *question topic* and *question number* attributes to identify the question that the annotated text segment answers. These attributes will have to be set explicitly for all ANSWER annotations that you add, while all other annotations will have default values.

We will give an example of an ANSWER annotation. Assume question 3 of topic VENEZUELA is: "When was Chavez elected president of Venezuela?" and it is answered in the sentence "Chavez was elected president in 1991. The ANSWER annotation that we will add to the document from which the sentence came will look as follows:
"Chavez was elected president **<ANSWER: *topic*=Venezuela, *question #*= 3, *confidence*=5, *conf comment*="", *partial*=false** in 1991**>**."

In the next section we give some general instructions about the annotations, followed by a section that describes all attributes of an ANSWER annotation.

General Instructions

This section gives general directions of what text spans should be annotated as answers.

What constitutes an answer

The most important decision that has to be made during annotation is what constitutes an answer for the question. For the purpose of this QA collection, *we will annotate as an* ANSWER *every text segment that can contribute to the derivation or construction of an answer to the question*. We have identified the following difficulties that you are likely to encounter during annotation:

1) **Text segments can answer questions only indirectly.** For instance, the question "Did most Venezuelans support the 2002 coup?" can be answered by the following two text segments (among others), both of which should be annotated as answers: "Protesters … failed to gain the support of the army" and "… thousands of citizens rallied the streets in support of Chavez." The above two answers are indirect in two different ways. First, the subject in the above two sentences in not "most Venezuelans" as the question asks. Rather, we can infer from the fact that thousand of citizens were against the coup that it was not the case that most Venezuelans supported the coup (and in the case the inference is questionable since thousand of people may protest although most Venezuelans may still support the coup). Second, the answer is indirect because we must infer from the fact that thousand of people rallied in support of Chavez that they were supporting the President of Venezuela who was overthrown by the coup, and thus they were against the coup. That is, even if the segment was "… most Venezuelans rallied the streets in support of Chavez," we would still have to infer that most Venezuelans did not support the coup and use a certain amount of discourse knowledge in the inference.

   When annotating indirect answers use your best judgment. If you can determine the answer from a text segment using common sense, then the segment should be annotated as an answer. Use the *confidence* attribute to indicate how confident you are that an answer can be inferred from the text segment. In the above example, both segments should be annotated as ANSWER with the former segment having a lower confidence score (maybe a 2 on the scale 1-5) than the second segment (maybe a 3 or a 4).

2) **Answers to some questions require combining information from more than one segment in the text.** For instance, the question "Are the Japanese unanimous in their opinion of the Kyoto protocol?" might be answered by combining information from a segment in a document that states that some Japanese support the Kyoto protocol with information from a segment in the same document that states that some Japanese do not support the protocol. While neither of the two

segments would have been sufficient to answer the question, when the information from the two segments is combined, that is enough to give an answer.

In situations such as the one described above, annotate as ANSWER any text segment that contributes to an answer. For instance, in the case of "Are the Japanese unanimous in their opinion of the Kyoto protocol?" question, annotate any text segment that expresses an opinion of Japanese source, regardless of whether or not the document alone contains sufficient information to answer the question. Set the *partial* attribute of the ANSWER annotations to ***true*** to indicate that a given answer segment does not provide a sufficient answer in isolation.

In addition, some segments may answer a question partially, without any need for combining their information with that in different segments. For instance, a partial answer to the question "When was the Kyoto protocol ratified?" maybe the segment "the protocol will be ratified in the near future," which although not answering the question completely gives a lower bound on the date of the ratification. Such segments should be annotated as answers with the *partial* attribute set to ***true***.

3) **Sometimes it may be hard to know whether the sources in the document match the entities about which questions ask.** For example, a question mentioned in 1) asks about the opinion of "most Venezuelans." The two answers given in the example mention "the army" and "thousands of protesters." Using our background knowledge and our common sense we can conclude (and again this inference is questionable) that "most Venezuelans" did not support the coup.

   Similarly, the question given as example in 2), "Are the Japanese unanimous in their opinion of the Kyoto protocol?" asks about the opinion of "the Japanese." It is not clear what should count as the opinion of "the Japanese" – the opinion of Japanese government sources, the opinion of Japanese news sources, or the opinion of any person or organization from Japan.

   A general guideline in the situation described above is to use your common sense and background knowledge and annotate anything that you believe to match the source in the question. If you are unsure of whether a source in a document can be associated with the source of the question, use the *confidence* attribute to indicate that and add a comment to the *confidence comment* field explaining the issue.

4) **Sometimes answers may be given in future tense or conditional statements**, especially since the document collection includes documents over a certain time span. For instance, the question "When was the Kyoto protocol ratified?" might be answered by "… will sign the protocol next month," or the statement "will **likely** draw the ire of X" could answer a question about X's opinion. Again, future tense and conditional statements should be annotated as ANSWER and the *confidence* and *confidence comment* attributes should be used to indicate how confident are you that the segment answers a question and what is the reason for

the reduced confidence. The confidence score that you assign to the ANSWER annotation should indicate how sure you are that the event occurred in the future given the text span. For instance, the text segment, "the protocol will be signed next March" answering the question "When was the Kyoto protocol signed?" should receive only medium *confidence* score. Although we may believe the author of the document, the event described is in the future and a certain turn of events may have changed the actual date of signing. On the other hand, if the question was "Was there a solar eclipse in 1990?" and it was answered by a segment "the next solar eclipse will be in August 1999", the ANSWER annotation should receive a high *confidence* score since although the event that the segment describes is in the future, we know that with a high probability the solar eclipse occurred as predicted.

5) **Sometimes text segments may express opinions only indirectly through the style and choice of language.** In such situations, it may be hard to pinpoint the exact place in the text at which the opinion is expressed. However, you should be able in such text segments to attribute the expression of opinion to a specific word or phrase. Annotate all words and/or phrases that you think express the opinion for which the question asks. If you cannot pinpoint any particular word or expression, then do not annotate any part of the text as an answer. For example, a question asking about Bush's opinion of Saddam Hussein, could be answered by the segment "Saddam has been oppressing the country for far too long, Bush iterated." In the above segment "oppressing" and "far too long" are the phrases that indicate Bush's negative attitude towards Saddam and should be annotated as ANSWER.

For answers that have difficulties different from the ones described above use your best judgment. The general guideline is to use your common sense and best judgment. If you can infer that a given segment answers a question either fully or partially, then annotate it as an ANSWER and use *confidence* and *confidence comment* fields to indicate if you feel uncertain about the answer.

Minimal Spans

When annotating a text span as an answer to a question use the minimum span that answers the question. For instance, the question "What is the Kiko Network?" should be annotated as answered by the text segment "a Tokyo environment umbrella organization," which is the minimal segment answering the question as opposed to the longer segment "a Tokyo environment umbrella organization representing about 150 Japanese groups," which also constitutes a legitimate answer to the question but is not minimal. We believe that annotating only minimal answer segments will make our evaluation of the QA annotations easier.

Below we summarize all attributes of an ANSWER annotation and how they should be used.

Answer Annotation Attributes

Question Topic

Use to indicate the topic of the question that the segment answers. Should be one of *kyoto, mugabe, humanrights,* or *venezuela*.

Question Number

Use to indicate the number of the question that the segment answers.

Confidence

Use the confidence attribute to indicate how sure you feel that the segment answers the question, matches the source, or is a full vs. partial answer. Should be one of the following: 1, indicating low confidence, 2, indicating moderate confidence, 3, indicating medium confidence, 4, indicating high confidences, and 5, indicating very high confidence. DEFAULT VALUE: 5, VERY HIGH CONFIDENCE.

Confidence Comment

Use this attribute to indicate why you feel less confident that the segment answers the question. This attribute is free form and can be filled with any text that presents an explanation. The attribute is especially important to include for answer segments that present difficulties different from the ones described above. DEFAULT VALUE: "", EMPTY STRING.

Partial Answer

A Boolean attribute that should be set to *true* if the segment presents only a partial answer to the question and to *false* otherwise. DEFAULT VALUE: FALSE.

APPENDIX C

INSTRUCTIONS FOR ANNOTATING TOPICS OF OPINIONS

This appendix contains the instructions for the manual annotation of topics of opinion. These instructions were used for the creation of the $\text{MPQA}_{\text{TOPIC}}$ corpus. The instructions are included here verbatim starting on the next page.

Instructions for Annotating Topic of Opinions

Introduction

Our ultimate goal is to create algorithms and methods that can automatically extract opinions from text. For the purpose of the discussion, we use the term *opinion* to refer to opinions, beliefs, emotions, sentiment, and other private states expressed in text. Private state is a general term used to refer to mental and emotional states that cannot be directly observed or verified (Quirk et al. 1985).

In order to be able to automatically extract opinions from text we will rely on Machine Learning (ML) techniques. Both for the development and evaluation of these techniques, however, we need a corpus of documents manually annotated with information about opinions. Fortunately, such a corpus already exists – the MPQA corpus contains documents manually annotated with information about opinions.

Documents in the MPQA corpus are manually annotated by designating all expressions of opinions at the fine-grained level of individual expressions of opinions. The manual annotations of opinions include a number of attributes of the opinions such as the *source* of the opinion, the *opinion trigger* or the words that signal the expression of opinion, the *polarity* or favorability of the opinion, the *strength* of the opinion, as well as the *target* of the opinion. However, in the current version of the MPQA corpus, the target attribute of opinion is included for very few of the actual annotations. The absence of more comprehensive marking of the target attributes is due to the challenging nature of the target annotation task in its original definition.

The target attribute, however, is an integral part of the each expression of opinion. As such, it is desired that automatic extractors of opinions are able to extract the target of each opinion. To facilitate the creation and evaluation of automatic targets, the purpose of this annotation task is to add the target attribute to the opinion annotations in the MPQA corpus.

In order to avoid shortcomings of previous approaches to target annotations we chose a different definition (and name) for the target annotation task. For the reminder of the document, the target of an opinion will be referred to as the *topic*. As a term topic carries more or less the same meaning, but we prefer it to target because it is a more general and vague term. Target generally carries the connotation that it refers to a specific well-defined concrete entity.

We define *topic* of an opinion as the (physical or abstract) entity, action, event, artifact, ideology, matter, etc. that is targeted by an opinion. Here are several examples of topics of opinions.

(1) President Chen Shui-bian has on many occasions expressed goodwill to mainland China. (Topic: mainland China)

(2) The IHRC said in a statement that the international community has formulated numerous documents to honor human rights after 50 years of bitter experience and the heavy losses that the humanity has suffered. (Topic: human rights)

(3) The IHRC said in a statement that the international community has formulated numerous documents to honor human rights after 50 years of bitter experience and the heavy losses that the humanity has suffered. (Topic: period of force and violence in international relations. **Note that here the topic is clear only from the context.**)

In all examples in this document, opinion attributes are marked as follows:

- sources of opinions are underlined and in bold.
- opinion triggers (words that signal the presence of opinion) are also underlined and shown in either blue for positive opinions, red for negative opinions or gray for neutral or non-sentiment carrying opinions (more detail about the non-sentiment carrying opinions to come) .
- spans that signal the topic are highlighted in yellow.

The last example – example (3) -- hints at why topic annotation is difficult. To circumvent some of the difficulty of topic annotation, we introduce and use the notion of *topic coreference* of opinions. We say that two opinions are *topic coreferent* if they share the same general topic. For example, the opinion from (3) is coreferent with the following opinion in the same document:

(4) Tehran-based Islamic Human Rights Commission (IHRC) on Sunday expressed concern about return of the period in which force and weapon had the last say in international relations.

Armed with the notion of topic coreference, the goal of our annotation task is to group (cluster) together those opinions that concern the same topic (are topic coreferent) and label every group (cluster) with the topic of the cluster. Additionally, we would like to mark the text spans that signal the expression of the topic (corresponding to the yellow highlights in the examples).

In the next section we give a brief background of the existing opinion information, followed by sections containing general instructions of what opinions should be considered topic coreferent and how to form labels for clusters of topic coreferent opinions.

Background

As mentioned in the introduction, documents in the MPQA corpus are annotated with expressions of opinions. We will augment the existing opinion annotations with information about the topic of opinions. For this purpose we will use a special annotation tool to display and augment existing opinion annotations. In this section, we give a really brief overview of the important parts of the existing opinion annotations.

In general opinions in language can be expressed either directly – e.g. "John hates Mary" – or through the choice of style and words in the language used – e.g. "John whined about school all the way to the cafeteria" (the choice of the verb "whine" rather than the more neutral "complain" signals the author's negative opinion of John). Documents in the MPQA corpus are annotated with both types of opinions, to which we will refer as *direct opinion* and *expressive subjectivity* respectively.

As mentioned previously opinions have a number of attributes such as opinion trigger, source, polarity, and strength. In the examples in this document we show all of these attributes using underlining, highlighting, and color following a template that is similar to the one used by the topic annotation tool. It is worth noting, however, that sometimes not all of the opinion's attributes are present in the context. For example, for some opinions the source attribute is not explicitly mentioned, but rather inferred from the text (e.g. the source of the opinion "John whined about school all the way to the cafeteria" is the writer, who is not explicitly mentioned in the sentence). Additionally, for most of the opinions the polarity attribute is missing. For instance, in the sentence:

(5) John predicted that it will take a defensive mistake for one of the two teams to score in the game.

Clearly, John's opinion (or private state) is expressed. However, this sentence does not express any form of positive or negative sentiment, but rather his belief toward the subject matter. In that respect, our use of the term opinion can be somewhat misleading as it typically implies the expression of some form of sentiment. Remember, however, that we use the term *opinion* (arguably quite loosely) to refer to *private state* or a mental state that cannot be directly observed or verified. In this definition of opinion, the private state does not to have to express sentiment to be considered opinion.

The significance of the missing polarity should become clear in the next section, as we have found out that topics of opinions that express some form of sentiment are easier to judge as compared to opinions that express just any belief. In the next section, we look into more detail of how opinion annotations should be performed.

Annotation Instructions

Topic Coreference

Recall from the definition that we consider two opinions to be topic coreferent if the general topic of the opinions is the same. The topic might be a concrete person or object such as:

(6) I hate John.
(7) Sue is very found of John.

Or it could be an abstract concept such as event, idea, etc.:

(8) CNN has discussed the latest developments in the Israeli-Palestine dialogue.
(9) It is quite important that the governments of Israel and Palestine resume communication.

In the general case, topic of opinions will not be as clear-cut and easy to judge as the opinions above. There are at least a few issues, of which we are currently aware and which will make the process more difficult:

Multiple opinions in a sentence

In many cases, the sentence that you will be annotating contains more than one opinion. In the following sentence, for example:

(10) In her view, Tsai said, both sides have been endeavoring to prevent the results of Taiwan's recent elections from affecting the stability of cross-strait relations.

There are multiple opinions marked shown by their opinion triggers. Depending on which of the opinions in the sentence are being annotated, the topics of the opinions differ:

(10a) In her view, Tsai said, both sides have been endeavoring to prevent the results of Taiwan's recent elections from affecting the stability of cross-strait relations. (Topic: cross-straight relations)

(10b) In her view, Tsai said, both sides have been endeavoring to prevent the results of Taiwan's recent elections from affecting the stability of cross-strait relations. (Topic: both sides (Taiwan and China))

When annotating, you will need to carefully understand which of the multiple opinions are annotated and assign the appropriate topic.

Non-sentiment vs. sentiment opinions

As already mentioned and perhaps hinted in the last example, often topics of opinions that carry sentiment are easier to judge than general, non-sentimental opinions. This is due to the fact that sentiment is typically clearly stated and directed toward a specific entity or event. Non-sentiment opinions can be generally vague and concern multiple entities, events, or ideas (multiple topics of opinions are discussed in more detail in the next subsection). When you judge the topic of non-sentimental opinions, in many cases you will have to carefully read the span of the opinion and make a conjecture about the topic, possibly based on the surrounding context. For example, the following opinion can be judged only from the context:

(11) John has repeatedly stated that although the defense needs some upgrades, unless the coaching staff spends more resources on the offensive skill positions,

it will be another disappointing season. (Topic: the Philadelphia Eagles, but not mentioned anywhere in the sentence)

The significance of judging the problem in context is intimately connected with the problem of having more than one topic in a single opinion. For instance, the above example could be considered topic coreferent with opinions regarding offensive skill positions, if it was a part of say a paragraph on offensive skill positions around the NFL. For example if the next sentence was:

(12) Sue agrees that her beloved Broncos due most of their recent success on the depth that they possess on the offensive skill positions.

Then the two opinions would be considered coreferent in a cluster concerned with offensive skill positions. This problem of more than one possible topic is discussed in more detail in the next subsection.

The topic annotation software attempts to facilitate the annotation task by distinguishing the opinions that were judged to carry sentiment from the non-sentimental opinions. Opinions sentiments are color-coded in the software with gray highlighting for the opinion words signaling non-sentimental opinions. Additionally, the software distinguishes between direct opinions and expressive-subjectivity, since the latter more often than not carry a sentiment.

Multiple topics per opinion

Often one opinion can be considered to be concerning multiple topics. In many documents in the corpus, the exact topic of opinion may be hard to judge and depend on the context. Consider the following examples:

(13) Tsai Ing-wen said Tuesday she foresees the possibility of the two sides of the Taiwan Strait resume dialogue next year.

…

(14) "It all depends on how mainland China interprets President Chen's latest remarks on cross-strait relations and how the two sides cultivate an environment favorable for resumption of their long-stalled dialogue," Tsai explained.

The question is, are the opinions of (13) and (14) topic coreferent. The topic of the first sentence is expressed quite clearly. The opinion in the second sentence, however, concerns more than a single entity or event and can be judged only from the context. Tsai's statement in the sentence discusses the resumption of the dialogue (as does the opinion in the first sentence), but also talks about President Chen's remarks and the cultivation of environment of the two sides.

When annotating opinions with more than one potential topic, we consider the topic of the opinion to be the part of the opinion which is being emphasized. To make this judgment, we have to ask ourselves the question what is the purpose of the information

that is being conveyed in the opinion. For example, consider the original context of the sentence. It was after the following sentence:

(15) Tsai said that there should be opportunities for the two sides to resume talks.

In this context, the opinion concerns the resumption of dialogue and is coreferent with statement. In the context, the information that the statement gives is predominantly concerning the talks and Chen's statements and the environment can be considered secondary issues.

Let's assume that the preceding sentence was slightly different:

(16) Tsai concurred with Zhen on the importance of the remarks.

In this context, the topic of the opinion is Chen's remark as the statement serves the purpose to elaborate on the remark. In this case the opinions from and should not be considered coreferent.

Topic hierarchies

Yet another problem with the topic annotations is that opinions might be concerning different aspects or parts of the same topic. Remember the previous example:

(17) Tsai Ing-wen said Tuesday she foresees the possibility of the two sides of the Taiwan Strait resume dialogue next year.

The topic of the opinion here is the resumption of the dialogue by the two sides of the Taiwan Strait next year. Notice, however, that this topic can have different level of specificity:

- The resumption of the dialogue next year.
- The resumption of the dialogue.
- The dialogue.

The task of our annotation is to judge whether opinions are topic coreferent. So the question is, given another opinion, the topic of which may differ in its level of specificity, are the two opinions topic coreferent. Say we were given the sentence:

(18) In Tsai's opinion, the dialogue in the Taiwan Strait is very important.

We have to judge whether it is topic coreferent with example. For the purpose of the annotation, we will assume the following definition concerning the specificity of opinion topics: Two opinions are topic coreferent if they discuss different part or aspects of the same general topic; the label of the topic is the most general common topic.

Following this definition, the opinions from our examples should be considered coreferent, with the label for the topic being "the Taiwan Straight dialogue." Note that the topic label is "the least common divisor" of the topics. For example, if the two topics

were the resumption of the talks and the history of the talks the label of the opinion cluster would be "the talks" although it might not appear as a separate topic of an opinion.

Note that the above definition talks about different parts or aspects of the same topic. In the example we talk about the resumption and the history of the talks (or the dialogue). Note that these opinions differ only on the level of specificity, but not in the general topic. If, on the other hand, we had an opinion concerning Iraq and another one concerning the war in Iraq, the two should not be considered coreferent although they both discuss Iraq. The difference is that the war in Iraq is not a mere specification or attribute of Iraq, but rather a separate concept.

Topic Spans

In addition to the topic coreference annotations, our ML approaches can benefit from knowing the part of text which provides the evidence for the decision of the topic. That is, we would like to manually add the span of text which indicates the expression of the opinion topic (the text with yellow highlights in the examples). For this purpose, the task should be fairly straightforward – add to each annotation the part of the opinion sentence which indicated the topic of the opinion.

When performing this annotation, please select the minimal span of text which addresses the topic. Recall the example:

(17a) Tsai Ing-wen said Tuesday she foresees the possibility of the two sides of the Taiwan Strait resume dialogue next year. (Topic: dialogue in the Taiwan Strait)

Here the topic was the dialogue between the sides of the Taiwan Strait and this is exactly the part that we will annotate. If on the other hand, the topic was "the resumption of the dialogue in the Taiwan Strait next year," we would annotate the example as follows:

(17b) Tsai Ing-wen said Tuesday she foresees the possibility of the two sides of the Taiwan Strait resume dialogue next year. (Topic: resumption of the dialogue in the Taiwan Strait next year)

In addition, note that in some cases the expression of the topic might not be in the same sentence as the opinion. Recall example (11):

(11) John has repeatedly stated that although the defense needs some upgrades, unless the coaching staff spends more resources on the offensive skill positions, it will be another disappointing season. (Topic: the Philadelphia Eagles, but not mentioned anywhere in the sentence)

In this example, the topic is not explicitly mentioned in the opinion sentence. If you encounter such an example, please annotate the part of the document outside of the sentence which mentions the topic and on which you based your topic coreference decisions.

Topic labels

So far we have mentioned, but have not discussed the labels of topic clusters. Once opinions are separated into clusters, we face the task of assigning a label representing the opinions in the cluster. In the previous section we mention that the label should be the "least common denominator" of the opinion topics in the cluster; In addition, we would like the label of the opinion to be a text segment that occurs somewhere in the text[1]. Furthermore, we would like for the text segment to come from one of the parts that you have annotated as spans expressing the opinions (the text with yellow highlights).

Thus, the task of labeling can be viewed as the task of finding the most general and representative span of text that describes the topic for the cluster from the text spans which express the topics of opinions. Going back to the previous examples:

(19) Tsai Ing-wen said Tuesday she foresees the possibility of the two sides of the Taiwan Strait resume dialogue next year.

(20) In Tsai's opinion, the dialogue in the Taiwan Strait is very important.

A good label for this cluster would be "dialogue in the Taiwan Strait", coming from the part of the second sentence highlighted in yellow. If for example, the title of the document from which these opinions came was "Taiwan Strait talks", this could also be a good characterization of the topic of the cluster, but we would not desire this label since it did not come from a span expressing the topic of any of the opinions in the cluster.

In some cases, the label of the opinion cluster may not be expressed in any of the opinion sentences. In this case it is acceptable to create a topic label that is not found in any of the documents text.

[1] The need for labels to be actual text segments is motivated by the fact that we will attempt to recover topics and labels using machine learning techniques. For machine learning techniques it is generally harder to infer text that did not occur in the same form anywhere in the document.

REFERENCES

Steven Abney. 1996. Partial parsing via finite-state cascades. In *Proceedings of the ESSLLI '96 Robust Parsing Workshop*.

ACE. 2006. Ace 2005 evaluation, November.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. Addison Wesley, May.

A. Bagga and B. Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *In Proceedings of COLING/ACL*.

A. Balahur, E. Lloret, O. Ferrandez, A. Montoyo, M. Palomar, and R. Munoz. 2008. The DLSIUAES Team's Participation in the TAC 2008 Tracks. In *Notebook Papers and Results, Text Analysis Conference (TAC-2008)*, Gaithersburg, Maryland (USA), November.

S. Basu. 2005. *Semi-supervised Clustering: Probabilistic Models, Algorithms and Experiments*. Ph.D. thesis, Department of Computer Sciences, UT at Austin.

S. Bethard, H. Yu, A. Thornton, V. Hativassiloglou, and D. Jurafsky. 2004. Automatic extraction of opinion propositions and their holders. In *2004 AAAI Spring Symposium on Exploring Attitude and Affect in Text*.

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *Proceedings of IJCAI*.

Claire Cardie, Janyce Wiebe, Theresa Wilson, and Diane Litman. 2003. Combining low-level and summary representations of opinions for multi-perspective question answering. In *2003 AAAI Spring Symposium on New Directions in Question Answering*.

Giuseppe Carenini, Raymond T. Ng, and Adam Pauls. 2006. Multi-document summarization of evaluative text. In *Proceedings of EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics*.

Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of EMNLP*.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proceedings of EMNLP*.

F. Choi. 2000. Advances in domain independent linear text segmentation. *Proceedings of NAACL*.

W. Cohen. 1995. Fast effective rule induction. In *Proceedings of ICML*.

Jim Cowie and Wendy Lehnert. 1996. Information extraction. *Communications of the ACM*, 39(1):80–91.

Douglass R. Cutting, Jan O. Pedersen, David Karger, and John W. Tukey. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329.

H.T. Dang. 2008. Overview of the TAC 2008 Opinion Question Answering and Summarization Tasks. In *Notebook Papers and Results, Text Analysis Conference (TAC-2008)*, Gaithersburg, Maryland (USA), November.

S. Das and M. Chen. 2001. Yahoo for amazon: Extracting market sentiment from stock message boards. In *Proceedings of APFAAC*.

K. Dave, S. Lawrence, and D. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of IWWWC*.

I. Davidson and S. Ravi. 2005. Clustering with constraints: Feasibility issues and the k-means algorithm. In *Proceedings of SDM*.

A. Demiriz, K. P. Bennett, and M. J. Embrechts. 1999. Semi-supervised clustering using genetic algorithms. In *Proceeding of ANNIE*.

T. Finley and T. Joachims. 2005. Supervised clustering with support vector machines. In *Proceedings of ICML*.

Y. Freund and R. Schapire. 1998. Large margin classification using the perceptron algorithm. In *Proceedings of Computational Learing Theory*.

Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. 2005. Pulse: Mining customer opinions from free text. In *Proceedings of the 6th International Symposium on Intelligent Data Analysis*, pages 121–132.

Daniel Gildea and Daniel Jurafsky. 2001. Automatic labeling of semantic roles. *Computational Linguistics*, 28:245–288.

B. Grosz, A. Joshi, and S. Weinstein. 1995. Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21(2):203–226.

T. Hasegawa, S. Sekine, and R. Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of ACL*.

Minqing Hu and Bing Liu. 2004. Mining opinion features in customer reviews. In *AAAI*, pages 755–760.

R. Iida, K. Inui, H. Takamura, and Y. Matsumoto. 2003. Incorporating contextual cues in trainable models for coreference resolution. In *Proceedings of the EACL Workshop on The Computational Treatment of Anaphora*.

Abraham Ittycheriah, Martin Franz, and Salim Roukos. 2001. IBM's statistical question answering system - TREC-10. In *Text REtrieval Conference*.

T. Joachims. 1998. Making large-scale support vector machine learning practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

Min Y. Kan, Kathleen R. Mckeown, and Judith L. Klavans. 2001. Applying natural language generation to indicative summarization. In *Proceedings of the 8th. European Workshop on Natural Language Generation*, Toulouse, France.

Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, July.

Hiroshi Kanayama, Tetsuya Nasukawa, and Hideo Watanabe. 2004. Deeper sentiment analysis using machine translation technology. In *Proceedings of COLING 2004*.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of COLING*.

S. Kim and E. Hovy. 2005. Identifying opinion holders for question answering in opinion texts. In *Proceedings of AAAI Workshop on Question Answering in Restricted Domains*.

Soo-Min Kim and Eduard Hovy. 2006a. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of ACL/COLING Workshop on Sentiment and Subjectivity in Text*.

Soo-Min Kim and Eduard Hovy. 2006b. Identifying and analyzing judgment opinions. In *Proceedings of HLT/NAACL*.

Nozomi Kobayashi, Kentaro Inui, Yuji Matsumoto, Kenji Tateishi, and Toshikazu Fukushima. 2004. Collecting evaluative expressions for opinion extraction. In *IJCNLP*.

Nozomi Kobayashi, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2005. Opinion extraction using a learning-based anaphora resolution technique. In *IJCNLP*.

K. Krippendorff. 1980. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, Beverly Hills, CA.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. 2009. Sentiment summarization: Evaluating and learning user preferences. In *Proceedings of EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics*.

X. Li and D. Roth. 2005. Discriminative training of clustering functions: Theory and experiments with entity identification. In *Proceedings of CoNLL*.

Wenjie Li, You Ouyang, Yi Hu, and Furu Wei. 2008. PolyU at TAC 2008. In *Notebook Papers and Results, Text Analysis Conference (TAC-2008)*, Gaithersburg, Maryland (USA), November.

X. Luo. 2005. On coreference resolution performance metrics. In *Proceedings of EMNLP*.

C. Macdonald, I. Ounis, and I. Soboroff. 2008. Overview of TREC-2007 Blog track. In *Proceedings of TREC 2007*.

I. Malioutov and R. Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of ACL/COLING*.

Inderjeet Mani. 2001. *Automatic Summarization*. John Benjamins Publishing Co, June.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, July.

A. McCallum and B. Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proceedings of the IJCAI Workshop on Information Integration on the Web*.

D. Moldovan, S. Harabagiu, R. Girju, P. Morarescu, F. Lacatusu, A. Novischi, A. Badulescu, and O. Bolohan. 2002. LCC tools for question answering. In *Proceedings of TREC 2002*.

T. Morton. 2000. Coreference for NLP applications. In *Proceedings of ACL*.

Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: capturing favorability using natural language processing. In *Proceedings of K-CAP*.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing*, 4(2).

V. Ng and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. In *In Proceedings of ACL*.

I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, and I. Soboroff. 2007. Overview of TREC-2006 Blog track. In *Proceedings of TREC 2006*.

I. Ounis, C. Macdonald, and I. Soboroff. 2009. Overview of TREC-2008 Blog track. In *Proceedings of TREC 2008*.

B. Pang and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*.

Bo Pang and Lillian Lee. 2008. *Opinion Mining and Sentiment Analysis*. Now Publishers Inc, July.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*.

R. Passonneau. 2004. Computing reliability for coreference annotation. In *Proceedings of LREC*.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *HLT/EMNLP*.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A comprehensive grammar of the English language*. Longman, New York.

M. Razmara and L. Kosseim. 2008. Concordia university at the tac-2008 qa track. In *Notebook Papers and Results, Text Analysis Conference (TAC-2008)*, pages 125–131, Gaithersburg, Maryland (USA), November.

T Reinhart. 1982. Pragmatics and linguistics. an analysis of sentence topics. *Philosophica*, 38(27):53–9.

Ehud Reiter and Robert Dale. 2006. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press, New York.

E. Riloff and J. Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceesings of EMNLP*.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. *Proceedings of AAAI*.

B. Rosenfeld and R. Feldman. 2007. Clustering for unsupervised relation identification. In *Proceedings of CIKM*.

Y. Seki. 2008. Summarization focusing on polarity or opinion fragments in blogs. In *Notebook Papers and Results, Text Analysis Conference (TAC-2008)*, Gaithersburg, Maryland (USA), November.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, 16:105–133.

Benjamin Snyder and Regina Barzilay. 2007. Multiple aspect ranking using the good grief algorithm. In *HLT-NAACL*.

W. Soon, H. Ng, and D. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4).

V. Stoyanov and C. Cardie. 2006a. Partially supervised coreference resolution for opinion summarization through structured rule learning. In *Proceedings of EMNLP*.

V. Stoyanov and C. Cardie. 2006b. Toward opinion summarization: Linking the sources. In *Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text*.

V. Stoyanov and C. Cardie. 2008a. Annotating topics of opinions. In *Proceedings of LREC*.

V. Stoyanov and C. Cardie. 2008b. Topic identification for fine-grained opinion analysis. In *Proceedings of COLING*.

V. Stoyanov, C. Cardie, J. Wiebe, and D. Litman. 2004. Evaluating an opinion annotation scheme using a new Multi-Perspective Question and Answer corpus. In *2004 AAAI Spring Symposium on Exploring Attitude and Affect in Text*.

V. Stoyanov, C. Cardie, and J. Wiebe. 2005. Multi-Perspective question answering using the OpQA corpus. In *Proceedings of EMNLP*.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, June.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2006. Latent variable models for semantic orientations of phrases. In *Proceedings of EACL*.

Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, June.

Richard Tong. 1999. An operational system for detecting and tracking opinions in on-line discussions. In *Working Notes of the SIGIR Workshop on Operational Text Classification*.

David R. Traum, Lenhart K. Schubert, Massimo Poesio, Nathaniel G. Martin, Marc N. Light, Chung Hee Hwang, Peter A. Heeman, George M. Ferguson, and James F. Allen. 1996. Knowledge representation in the trains-93 conversation system. *International Journal of Expert Systems*, 9(1):173–223.

P. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL*.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*.

E. Voorhees and L. Buckland. 2003. Overview of the TREC 2003 Question Answering Track. In *Proceedings of TREC 12*.

Ellen Voorhees. 2001. Overview of the TREC 2001 Question Answering Track. In *Proceedings of TREC 10*.

Ellen Voorhees. 2002. Overview of the 2002 Question Answering Track. In *Proceedings of TREC 11*.

K. Wagstaff and C. Cardie. 2000. Clustering with instance-level constraints. In *Proceedings of the 17-th National Conference on Artificial Intelligence and 12-th Conference on Innovative Applications of Artificial Intelligence*.

J. Wiebe and E. Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of CICLing*.

J. Wiebe, T. Wilson, and C. Cardie. 2005a. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 1(2).

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005b. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 1(2).

J. Wiebe. 2005. Personal communication.

T. Wilson, J. Wiebe, and R. Hwa. 2004. Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of AAAI*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT/EMNLP*.

T. Wilson. 2005. Personal communication.

I.H. Witten and E. Frank. 2000. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco.

Ainur Yessenalina and Claire Cardie. 2009. A system for classifying polarity of opinion expressions. Personal communication.

Jeonghee Yi, Tetsuya Nasukawa, Razvan C. Bunescu, and Wayne Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of ICDM*.

Oren Zamir, Oren Etzioni, Omid Madani, and Richard M. Karp. 1997. Fast and intuitive clustering of web documents. In *Knowledge Discovery and Data Mining*, pages 287–290.

Li Zhuang, Feng Jing, and Xiaoyan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management*, pages 43–50.