Using Natural Language Processing to Mine Multiple Perspectives from Social Media and Scientific Literature

by

Amjad Abu Jbara

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2013

Doctoral Committee:

Professor Dragomir Radkov Radev, Chair
Associate Professor Steven P. Abney
Assistant Professor Eytan Adar
Assistant Professor Qiaozhu Mei
Assistant Professor Emily Kaplan Mower Provost

To my mother, my wife, and my adorable daughters

ACKNOWLEDGEMENTS

First and foremost, all praise and thanks are due to God for giving me the power to believe in my passion and pursue my dreams. I could never have finished this dissertation without the faith I have in Him.

I am heartily thankful to my advisor, Professor Dragomir Radev for his splendid guidance and encouragement during the course of my studies. His support and insightful advice were crucial to my academic success and my development as a researcher.

I am also grateful to Steven Abney, Eytan Adar, Qiaozhu Mei, and Emily Provost for serving on my dissertation committee and for providing constructive comments that made this dissertation better.

I was fortunate to collaborate and discuss research with brilliant researchers at the Computational Linguistics and Information Retrieval (CLAIR) group at the university of Michigan. Thanks to all the former and current members of CLAIR, especially Ahmed Hassan, Arzucan Ozgur, Vahed Qazvinian, Pradeep Muthukrishnan, Rahul Jha, Ben King, and Wanchen Lu. I would also like to extend my gratitude to our collaborators on the SCIL and FUSE projects from Columbia university, especially Kathy McKeown, Mona Diab, Owen Rambow, Muhammad Abdul-Mageed, Weiwei Guo, and Pradeep Dasigi.

I am sincerely thankful to Hani Qaddomi Scholarship Foundation (HQSF) for funding my master's studies. Special thanks to Professor Karem Sakallah who encouraged me to apply to Michigan and whose advice and constant support were important to me during the years I spent at Michigan.

I am also grateful to my amazing friends who made my life in Ann Arbor enjoyable and memorable, especially Ahmed Hassan, Khaled Alashmouny, Samer Kadous, Ahmed Almuhtady, Ahmed Mady, Mahmoud El-Azzouny, Ahmed El-Sabbagh, Khaled Aljanaideh, Ahmad Aljanaideh, Abdelrahman Mayhoub, Amr Alaa, and Ihab Ismael.

Finally, I would like to express my greatest and deepest gratitude to my family. Thanks to my mother whose endless love and sincere prayers have always been with me. Thanks to my brother Ahmed and my sisters Rawand, Rafeef, and Rahada for the unconditional support and constant encouragement they always give to me. Special thanks to Basma, my wife and best friend, for all the tremendous love and support she gives to me. Thank you Basma for being a steadfast source of encouragement and inspiration to me. It is no exaggeration to say that without your sacrifice and infinite patience, this dissertation would never have existed. Last but not least, thanks to my daughters, Jana and Sana, for being so cute and filling my life with joy and hope.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

This thesis studies how Natural Language Processing techniques can be used to mine multiple perspectives from textual data. The first part of the thesis focuses on analyzing the text exchanged by people who participate in discussions on social media sites. We particularly focus on threaded discussions that discuss ideological and political topics. The goal is to identify the different viewpoints that the discussants have with respect to the discussion topic. We use subjectivity and sentiment analysis techniques to identify the attitudes that the participants carry toward one another and toward the different aspects of the discussion topic. This involves identifying opinion expressions and their polarities and identifying the targets of opinion. We use this information to represent discussions in one of two representations: *discussant attitude vectors* or *signed attitude networks*. We use data mining and network analysis techniques to analyze these representations to detect rifts in discussion groups and study how the discussants split into subgroups with contrasting opinions.

In the second part of the thesis, we use linguistic analysis to mine scholars' perspectives from scientific literature through the lens of citations. We analyze the text adjacent to reference anchors in scientific articles as a means to identify researchers' viewpoints toward previously published work. We propose methods for identifying, extracting, and cleaning citation text. We analyze this text to identify the purpose (author's intention) and polarity (author's sentiment) of citation. Finally, we present several applications that can benefit from this analysis such as generating

multi-perspective summaries of scientific articles and predicting future prominence of publications.

CHAPTER I

Introduction

> *"I show two versions of reality and each makes complete sense to the participant who sees it. I think that's how life works."*
>
> *- Bill Watterson*

A popular old Indian legend tells the story of six blind men who were asked to determine what an elephant looks like by feeling different parts of its body. Each man described the elephant from his own perspective and based on his personal experience. The blind man who feels a leg says the elephant is like a pillar; the one who feels the tail says the elephant is like a rope; the one who feels the trunk says the elephant is like a tree branch; etc. This story is not just a wisdom story. It is an eloquent, concise description of how life works. Reality is multi-faceted and different people may see it differently when they look from different perspectives. This is why there are different ideologies, different cultures, different religions, and different approaches to solving problems and answering research questions. This pluralism in views, beliefs, and opinions can be observed everywhere: in family and friends conversations, in professional and business meetings, in parliaments, in conferences, in publications, in online discussions and dialogs, and everywhere.

The wide spread of the internet and the revolutionary growth of the world wide web facilitated access to information and increased the interaction among people. Social media sites are nowadays among the most visited sites on the internet. These sites allow users to post and share content with others. Examples of such sites include discussion forums, blogs, social networks, and instant messaging applications. Today's technologies gave people more opportunities than ever before to communicate, interact with each others, and express their opinions towards everything. This resulted in a huge influx of opinion-rich text being available online. Here is where Natural Language Processing techniques come into play by providing tools to analyze and mine this text. This analysis allows for a better understanding of how people see things from different perspectives and how they behave and what language they use when they communicate with one another.

Scientific research is another domain where the fact that reality is multi-faceted manifests itself. The same research question may be approached in several different ways by different scholars. Different scholars may also address different aspects of one research problem where every scholar focuses on the aspect that looks more important or more interesting from his or her perspective. In addition, when researchers describe a piece of related prior work, they usually focus on different aspects of it. One researcher may describe the problem it addresses; another researcher may summarize its methodology; a third researcher may discuss its results or criticize its limitations, and so on.

Identifying and analyzing the different viewpoints is useful for many applications. For example, decision makers in governments and political entities need to know how the public opinion reacts to their decisions; funding agencies need to measure how successful their funded projects are; companies need to know what their customers think about their products; hiring committees at universities and research institutions need to quantitatively and qualitatively evaluate the impact of a researcher's work, and so on.

The main focus of this dissertation is on using linguistic analysis techniques to identify, analyze, and summarize multiple viewpoints from textual data. We apply our analysis to two different domains: social media and scientific literature. In the rest of this chapter we give some background about the problem and a brief overview of the thesis and its contributions.

1.1 Multi-Perspectivism and Language

> *It is our needs that interpret the world; our drives and their For and Against. Every drive is a kind of lust to rule; each one has its perspective that it would like to compel all the other drives to accept as a norm.*
>
> *Friedrich Nietzsche*

A perspective or a viewpoint, in the context of cognition, is a mental view of situations and facts, and judging their relative importance.[1] Friedrich Nietzsche, a German philosopher, developed the theory of Perspectivism which states that all idea generation processes take place from particular perspectives. This means that there are several possible conceptual schemes in which judgment of truth can be made. This in turn implies that no way of seeing the world can be taken as an absolute truth. Schacht [169] expanded the ideas of Nietzsche into a revised form of "objectivity" in relation to "subjectivity". These theories were the foundations for many studies about Multi-Perspectivism and subjectivity in the Philosophy and Psychology literatures [151, 82, 81].

And because language is the medium that humans use to communicate and ex-

[1] http://www.thefreedictionary.com/perspective

press their beliefs and opinions, many researchers studied Multi-Perspectivism and subjectivity from a linguistic point-of-view [50, 192, 59, 46, 171]. For example, Banfield [27] studied the sentences that reflect a character's psychological point of view (subjective sentences), in contrast to sentences that objectively narrate events or describe facts. Psycholinguistic is a field of science that studies the impact of cognitive and psychological factors on the use of language. Sociolinguistics is a field that studies the relationship between social and cultural norms, expectations, and contexts on the use of language. The research that has been done in these two areas show that the mental status and the social context affect the way the language is comprehended and used. This includes the choice of language units such as words and phrases, the composition of sentences and their grammaticality, the structure of arguments, etc.

In the Natural Language Processing (NLP) and the Information Retrieval (IR) fields, subjectivity and sentiment analysis and opinion mining studies are essentially based on the ideas of Multi-Perspectivism and the findings of the psycholinguistics and sociolinguistics research. For example, Wiebe [200] used the ideas of Banfield about *subjectivity* and the concept of *private states*, defined by Quirk et al. [158] as states that are not open to objective observation or verification, to track the psychological point of view in third-person fictional narrative text. She developed an algorithm that looks for narrative regularities in the ways that authors manipulate point of view. The algorithm tracks the viewpoints of the characters that appear in text based on the regularities found. Greene [66] studied text that reveals perspectives that are not necessarily expressed in overt expressions of opinion. He conducted psycholinguistic experiments to guide and support his work on identifying perspectives and implicit sentiment.

Multi-Perspectivity was also studied in the context of building question answering systems for opinion-based questions. In 2002, Wiebe and her colleagues held a two month long workshop about multi-perspective question answering. The participants in the workshop studied how opinions are expressed in language. They developed annotation instructions for identifying expressions of opinion in text. They used this annotation scheme to manually annotate a corpus of news articles for opinion, the MPQA corpus. This work was the basis for a lot of work in multi-perspective question answering and other sentiment analysis and opinion mining applications.

Several other studies focused on the language used by scholars to express their viewpoints toward prior and related work [120, 186, 216, 93, 32]. For example, Thompson and Ye Yiyun [186] examined academic papers to identify which kinds of reporting verbs are used in citations as a basis for developing material for teaching scientific writing skills to non-native-speakers. MacRoberts and MacRoberts [120] studied the language used in negational references (i.e. negative citations).

This thesis builds on top of these efforts by applying the concepts of Perspectivism and subjectivity to social media and scientific literature.

1.2 Multiple Perspectives in Social Media

The first part of the thesis focuses on developing NLP and network analysis techniques for mining multiple perspectives from social media. The term *Social Media* refers to “a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of user-generated content”[99]. The data published and shared on these sites is huge in volume, increases at very high rates, and is very diverse. Although the content of social media sites is in all kinds of formats: text, images, audio, video, animation, etc., text is still the most prevalent format and the most common way of communication and interaction among users. In this part of the thesis, we analyze the text that the users of social media sites exchange when they communicate online.

We are particularity interested in threaded discussions that discuss controversial topics. Our research attempts to answer several questions like: How do people use language to express their viewpoints? What language constructs do they use to show agreement or disagreement? What makes a group of discussants split into subgroups and how does this affect the language they use for communication? How can linguistic analysis be applied to detect or even predict rifts in groups? How do people influence each others' opinions? and so on. Answering these questions is useful for a broad spectrum of applications in marketing, economics, politics, and social studies.

We start by studying the characteristics of online discussions. We use a large corpus of labeled discussion threads to analyze the behavior of discussants and the language they use in online dialogs. We use the knowledge gained from this analysis to develop methods for detecting subjective text and identifying the explicit and implicit attitudes expressed in it. We represent the attitude expressed by the discussants in two formal representations: a signed attitude network and a space of attitude vectors (attitude profiles). These formal representations are then used for higher levels of analysis of the discussion such as detecting subgroups, identifying influential participants, measuring how disputed a topic is, etc.

1.3 Multiple Perspectives in Scientific Literature

In the scientific literature domain, we study the multiple viewpoints of scholars towards prior work through the lens of citations. We analyze the text that appears around citations in scientific articles. This work seeks to answer questions like: for what purposes do researchers cite previous work? How does the purpose of citing a publication changes over time? How do researchers express their viewpoints towards published research? Can we do some form of sentiment analysis of citation text to distinguish between positive, negative, and neutral citations?

Answering such questions is important for several applications such as summarizing a published article from the viewpoints of expert researchers, generating multi-perspective surveys of research topics, evaluating the impact of a researcher's work, identifying controversial scientific arguments, predicting which papers will have more impact and receive more citations than others, etc.

In this part of the thesis, we address the problem of identifying and analyzing citation text. This involves identifying text fragments that contain explicit references to other papers and the context and the scope of each reference. We use the term *citation context* to refer to the text in a scientific article that appear around an explicit reference and talk about it. We use the term *reference scope* to refer to the fragments of sentences that talk about a reference in citing sentences that cite multiple references. We analyze citation text to identify the author intention behind citing another paper and whether the citation is polarized (i.e. carries a non-neutral sentiment towards the cited work). We use this analysis of citation purpose and polarity to predict the future prominence of papers. We also show how this analysis can lead to more accurate and more informative bibliometric measures. We also present a method for producing citation-based summaries of scientific articles summarized from the viewpoints of the other scholars who read the paper and cited it. Although several methods have been advised for this problem, our method is uniquely characterized by focusing on the coherence and readability of summaries generated from citation text.

1.4 Summary of Contributions

The contributions of this thesis are:

1. An analysis of online discussions that focuses on the behavior of discussants and the language they use when they interact and communicate online. The analysis uses a dataset of discussion threads comprising 1 million posts in more than 16,000 disputed topics. The dataset was collected from a debating site called createdebate [2]. To our knowledge this is the first time this dataset is analyzed and used in research. The dataset is different from other similar datasets in being self-labeled (by the discussants themselves) for agreement, stances, and influence. This make it a reliable source for understanding the relation between opinion and language.

2. A framework for processing threaded discussions and mining them for multiple viewpoints. We propose a pipeline of linguistic analysis components for identifying attitudes, topics, and attitude targets and for building formal representations of discussant attitudes.

3. An approach for identifying the polarity of out-of-vocabulary (OOV) words. OOV words are common in the text used on social media sites. Our method is an extension of an existing algorithm that uses random walk on a semantic graph extracted from Wordnet to identify word polarity [76]. Our method uses co-occurrence statistics computed from a large corpus of social text to augment the semantic graph with OOV words. We show that restricting the computation of co-occurrence statistics to social text (e.g. tweets) leads to significantly better

[2] createdebate.com

performance than using general text corpora. We also show that combining corpus-based and random walk based methods lead to better performance in identifying word polarity than using corpus statistics alone.

4. A method for identifying the targets of attitude in ideological and political discussions and a method for encoding this information in a formal way. We propose two representations of discussants' attitudes: in the form of a *signed attitude network* and in the form of a space of attitude vectors (which we term *discussant attitude profiles*)

5. An approach for detecting opinion subgroups in discussions. The approach uses the aforementioned framework to mine attitudes from discussions and build formal representation of them. Data mining and network analysis methods are then used to study how the group of discussants split into subgroups with respect to the discussion topic.

6. An approach for extracting and cleaning citation text from academic articles. This includes identifying reference anchors and identifying the scope and the related context of each reference.

7. An approach for identifying the purpose and polarity of citation. Citation purpose refers to the author's intention behind citing a piece of previous work. For example, previous work might by cited in the context of comparing it to a new method or to declare that the new work is based on or inspired by that previous work, etc. Citation polarity refers to whether the citation indicates positive, negative, or neutral sentiment from the author towards the cited work.

8. A method for producing multi-perspective, citation-based summaries of scientific papers. This method is different from previous methods in that it focuses

on producing coherent and readable summaries.

1.5 Thesis Outline

The thesis falls in two parts. The first part studies the problem of mining multiple perspectives from multi-party discussions that take place on the internet. This part is based on the work published in [15, 8, 12, 16, 9]. This part is organized as follows: Chapter II is an introduction to the first part. The aim of this introduction is to give background about social media analysis and motivate why we are interested in studying multiple perspectives in discussions. The definitions of the terms used and the problems addressed in the first part of the thesis are also presented in this chapter. The introduction chapter also reviews the related work. Chapters III - V present our approach to analyzing threaded discussions and identifying multiple viewpoints. The approach involves several tasks such as identifying the opinion expressions [77, 74] (Chapter III), handling negation [12] (Chapter IV), identifying the candidate targets of opinion (Chapter V), and associating each opinion expression with its target [15]. In Chapter VI, we show how our framework for identifying opinions in discussions can be used to detect rifts in discussion groups which lead to their split into smaller subgroups with contrasting views [15].

The second part studies the problem of mining multiple viewpoints form scientific literature. This part is based on the work published in [10, 13, 159, 13]. Chapter VII presents the motivations and defines the research problems. Chapters VIII - IX present the approach. The approach involves several processing tasks such as reference tagging [10], reference scope identification [13], and citation purpose and polarity classification [7]. Chapters X -XI present some useful applications. These applications include summarizing multiple viewpoints and predicting the future prominence of papers [10, 159].

Chapter XII concludes the dissertation and summarizes the contributions. Chapter XIII suggests directions for future work.

Part I. Mining Multiple Perspectives from Social Media

CHAPTER II

Introduction

The revolutionary growth of the world wide web made the communication among people easier than ever before. Websites that allow people to communicate and talk about topics of shared interest are common. Social networking sites are nowadays the most visited and used sites on the internet. These sites allow users to communicate not only with their personal acquaintances or friends, but also with a vast pool of random people from different locations and different backgrounds. More and more people today are making their information and their opinions available publicly on the internet in the form of blog posts, status updates on social networking sites, posts on discussion forums, etc. The abundance of opinions and the ease of communication lead to lots of discussions among the members of social media sites. Members of these sites discuss all kinds of topics including politics, ideologies, social causes, religions, and scientific arguments. In such discussions it is quite natural to see instances of agreement and disagreement among discussants. For example, the following two snippets are taken from a discussion forum:

> *(1) The Global Warming theory is nonsense. What we've been experiencing over the past decade or so is part of a natural cycle that will be reversed in the near future.*
>
> *(2) I disagree with you. The theory is true and global warming is happening. According to an estimate, after 10 years from now, people will be using skin protectors and temperature controller devices in their homes.*

The two snippets discuss a controversial topic, Global Warming. The writer of (1) adopts the viewpoint that rejects the theory of global warming and sees it as a normal phase of the natural cycle. The writer of (2) adopts an opposing viewpoint and expresses an explicit disagreement with the writer of (1).

This part of the dissertation is interested in analyzing this kind of discussion where discussants may have different viewpoints with respect to the discussed topic. We particularly focus on threaded discussions in which user comments are grouped in a hierarchy originating from a root post.

In this chapter, we briefly review the previous work that has been done on this problem. A more detailed review of the previous work related to each subtask will be presented in the chapter that discusses each task. Next, we define the terms used and the problems addressed in this part of the thesis. As a case study, we present an analysis of one large corpus of discussion threads crawled from a discussion forum. We end this chapter with an example that illustrates the entire approach to identifying and representing discussants' viewpoints.

2.1 Related Work

This part of the thesis is related to a large body of research in the areas of sentiment analysis and opinion mining. Pang & Lee [145] and Liu and Zhang [116] wrote two recent comprehensive surveys about sentiment analysis and opinion mining techniques and applications. As we mentioned in Chapter I, all the work in these two areas is based on the idea of subjectivity which has been introduced first by philosophy scholars and then studied by psycholinguistic and sociolinguistic scholars. The first task in opinion mining is subjectivity analysis. The goal of subjectivity analysis is to distinguish between text that expresses opinion and objective text that presents factual information. Previous work proposed methods for identifying cue words that indicate subjectivity [197, 79, 26]. Other studies focused on context-aware subjectivity analysis where the subjectivity of the text is determined after taking the context into account [166, 211, 144, 149]. Wiebe et al. [198] list a number of applications of subjectivity analysis such as classifying emails and mining product and movie reviews.

The second task after identifying subjective text is to identify the polarity (or the semantic orientation) of subjective text. The polarity of a word indicates the direction the word deviates to from the norm for its semantic group or lexical field [111]. Polarized words can be either positive, to express a desired state; or negative, to express an undesired state. Several methods have been proposed for building polarity lexicons. Most of these methods start with a set of seed polarized words and use them to determine the polarity of other words in a bootstrapping fashion. There are two main categories of approaches to perform this bootstrapping, dictionary-based approaches [97, 129, 90, 56, 58, 102, 22, 132, 193, 98, 76] and corpus-based approaches [78, 190, 193, 98, 181]. Dictionary based methods start with a small set of seed labeled words and grow this set by searching a thesaurus for the synonyms and antonyms of the seed words. A synonym to a seed word is assigned the same polarity as the seed word and an antonym to a seed word is assigned the opposite polarity. The newly labeled words then get added to the seed set and the bootstrapping continues. Corpus-based methods use co-occurrence statistics in a large corpus of text (e.g. the web) to determine the relatedness of every unlabeled word to two sets of labeled words, one containing positive words and the other containing negative words. An unlabeled word is assigned the label of the set it is more related to.

The next task in opinion mining is to identify sentence level polarities. The objective here is to determine if a sentence is subjective or objective and if it is subjective determine whether it expresses positive, negative, or neutral opinion. Previous work addressed this problem as a supervised classification task in which a classification model is trained using lexical, syntactic, and dependency features extracted from a labeled dataset [80, 75, 165, 167].

All the aforementioned work focused on identifying opinion-bearing text. Other research efforts focused on identifying the holders and the targets of opinion [212, 150, 126, 29]. In most applications, the holder of opinion is often the author of the text such as the author of a blog post or the author of a product review. The target of opinion can be a product, a feature of a product, a service, a named entity, a topic, etc. Most of the methods proposed for opinion target identification start by finding named entities and frequent noun phrases and treating them as potential targets of opinion. The relation between a potential target and an opinion expression that appears close to it is determined based on the syntactic structures and the dependency relations that connect them.

This analysis of opinion expression, their holder, and their targets is useful for many applications. One example of such applications is identifying viewpoints and perspectives [67, 113, 114, 110]. For example, in [114], the authors experiment with several supervised and statistical models to capture how perspectives are expressed at the document and the sentence levels. Laver et al. [110] proposed a method for extracting perspectives from political texts. They used their method to estimate the policy positions of political parties in Britain and Ireland, on both economic and social policy dimensions. Other research efforts addressed the problem of summarizing viewpoints [214, 187, 179, 172, 148]. For example, Zhuang et al. [214] and Titov & McDonald [187] addressed the problem of summarizing product reviews. Paul et al. [148] proposed an extension of the LexRank summarization algorithm to make it capable of summarizing contrastive viewpoints in opinionated texts.

This was a brief overview of the related work. Each chapter will include a related work section that presents a more detailed and more specific review of the prior work done on the task discussed in the chapter.

2.2 Definitions

The beginning of wisdom is the definition of terms

–Socrates

In this section, we define the terms used in the first part of the thesis:

- Social media: the means of interactions among people in which they create, share, and exchange information and ideas in virtual communities and networks [19]. Examples include social networking sites such as Facebook, photo sharing sites such as Picasa, video sharing sites such as Youtube, etc. In this thesis we focus on social media sites that allow the users to engage in threaded discussions.

- Discussion: the consideration of a question in open and usually informal debate [1]. Our focus in this thesis is on discussions that take place on social media sites and discuss ideological and political topics. Such discussions usually start by a person asking a question or expressing an opinion on a discussion forum, or posting a blog post on this personal blog. Those who are interested in the topic comment on the post and on each others.

- Threaded discussion: a common way of structuring and presenting discussions in social media sites. A threaded discussion consists of a set of posts grouped in a tree-like structure. The thread usually originates from a root post posted by one of the participants. Other participants read that post and add comments if they are interested in the topic. Other members can also comment on others' comments. The result is a tree of posts rooted at the first post.

- Post: a piece of text (could also contain other media types) written by one of the discussants. A discussion thread consists of a hierarchy of posts. The first post in a discussion thread is termed the *root post*. We also refer to the post that is a response to a previous post as *comment*.

- Attitude: the way a person views something or tends to behave towards it, often in an evaluative way.[2] Participants in discussions express their attitudes explicitly or implicitly towards one another and towards aspects of the topic being discussed. For example, the sentence “I like your brilliant ideas” is expresses positive attitude from its writer to its recipient. Similarly, the sentence “I hate school uniforms” expresses negative attitude towards *school uniforms*.

- Opinion expression: a piece of text (a word, an acronym, a phrase, etc.) that expresses a subjective belief that is the result of emotions or interpretation of

[1] http://www.merriam-webster.com/dictionary/discussion

[2] http://dictionary.reference.com/browse/attitude

facts. Opinion expressions can be positive, indicating agreement, praise, or admiration (e.g. like, love, agree, good, etc.) or negative indicating disagreement, insult, or dislike (e.g. hate, disagree, bad, etc.).

- Out-of-Vocabulary word: is a word that cannot be found in the standard dictionaries of the language and is not a proper name. These words could be colloquial words, acronyms, or misspelled words. OOV words are common in online discussions and many of them are used in subjective contexts (e.g. using *gr8* to mean *great*). Handling OOV words is important for opinion mining and sentiment analysis. This thesis proposes a method for identifying the polarity of OOV words.

- Attitude/opinion target: the object that the attitude holder is expressing attitude towards. An attitude target can be another discussant, an entity mentioned in the discussion, or an aspect of the discussed topic. For example, in "illegal immigration is bad for the economy", *illegal immigration* is the target of the opinion expression *bad*. Similarity, in "I agree with you", the recipient of this sentence (referred to by the pronoun *you*) is the target of the opinion expression *agree*.

- Subgroup: a subdivision of a group of discussants. The members of a subgroup share similar opinion with respect to the discussion topic. For example, in a discussion about a topic like the *health care reform*, the group of discussants may split into two subgroups, one that argue in support of the proposed reform and an opposing one that argues against the proposed reform.

Number of threads	14,308
Number of posts	178,317
Number of sentences	983,800
Number of participants	9,743
Average number posts per thread	12.4
Average number of sentences per post	5.51
Average number of participants per thread	6.5
Average number of subgroups per thread	2.24
Percentage of “dispute” comments	35.5%
Percentage of “support” comments	18.8%
Percentage of “neutral” comments	45.7

Table 2.1: Statistics of the Createdebate corpus

2.3 The Discussion Genre

The goal of this section is to study how the participants of in online discussions interact and express their opinions from a linguistic and a behavioral points of view. We use a large corpus of discussions crawled from a discussion forum called Createdebate[3]. Table 2.1 shows some statistics of the data set. The data set we used contains all the discussion threads posted on the web site since it started till January of 2012. Createdebate is a web site that allows its users to start discussions (debates) about any topic. The discussions cover a broad spectrum of topics such as Abortion, Elections, Religions, Politics, Human Rights, Economy, etc.

Each discussion thread starts with a question or an argument posted by one of the site users. The initiator of the discussion suggests two or more viewpoints for the other users to select from and support. The users can add more viewpoints. Users who participate in the discussion must explicitly declare their viewpoints by selecting from the list of viewpoints or add a new item to the list if they have a new viewpoint. Participants can post in response to the initial post (the root of the thread) or comment on each others posts. When participants comment on posts written by other

[3] http://www.createdebate.com

participants, they can explicitly declare whether they are commenting to *dispute* or to *support* the post they are commenting on. Moreover, the discussants can rate each others posts for the strength and the degree of persuasiveness. These post ratings can be viewed as indicators of influence.

So, the data is self-labeled for attitude relations, for viewpoints, and for influence. We analyzed this data in terms of both the language used in the discussions and the behavior of the participants. We present our observations from the data in the rest of this section.

- Discussants tend to comment on others' posts when they disagree more than when they agree. The number of dispute (or negative) comments is almost double the number of negative comments. The percentage of the former is 35.5% versus 18.8% for the later. These numbers are also supported by the average number of positive and negative comments per thread which are 2.28 and 4.42, respectively.

- Posts that express disagreement are usually longer (in number of sentences) than posts that express agreement. The average number of sentences in *dispute* posts is 7.86 which is significantly greater than the average number of sentences in *support* posts: 3.68. Figure 2.1 shows the distribution of sentence count in both *support* and *dispute* posts. To verify that the difference in post length is significant, we use a two-tailed paired Student's T-test (with alpha set to 0.05). The test showed that the difference is significant ($p < 0.0001$).

- Out-of-Vocabulary (OOV) words are very common in online discussions. We observed that 27.7% of the sentences in the corpus contain at least one OOV word. We identify OOV words by looking up all the words in the corpus in two

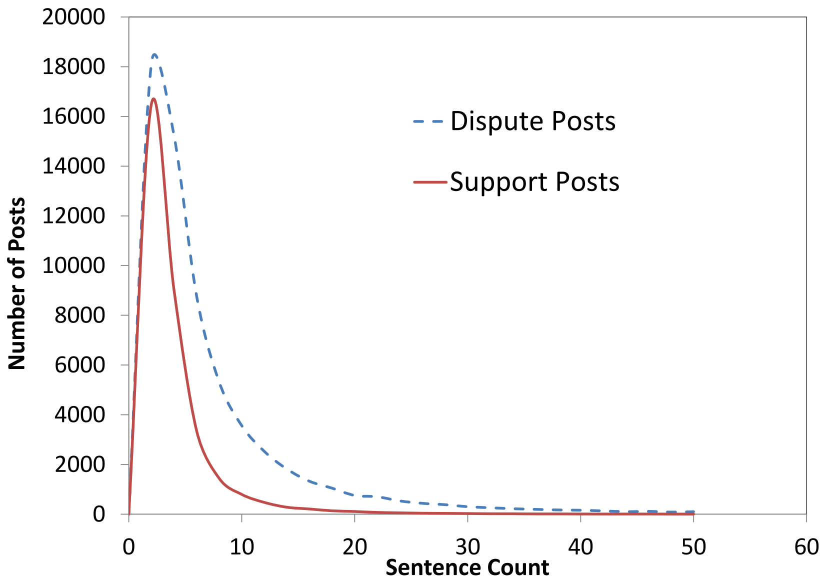


Figure 2.1: Distribution of sentence count support (solid red) and blue (dashed blue) posts. The average sentence count is 7.86 in support posts and 3.68 in dispute posts

standard English dictionaries. The word is labeled as OOV if it does not appear in any of the dictionaries. This indicates that any text processing applied to this corpus should take this issue into account. The consideration of this issue becomes more important in opinion mining systems because OOV words are likely to be polarized since people tend to use colloquial words and acronyms when they express opinion as a way to emphasize their opinions and impress the audience. In Chapter 3.4, we present a method for handling OOV words and identifying their polarities.

- Discussants tend to avoid expressing their attitude toward other participants by addressing them directly and explicitly. The ratio of comments that each has at least one sentence that contains a first person pronoun (e.g., I) and a mention (i.e. name or second person pronoun) of the recipient of the comment is 23.2% only. This means that the majority of comments do not address the recipient of the post when the agreement or disagreement is expressed. We also observed that direct expressions of attitude are significantly more common in *dispute* posts than in *support* posts. The ratios are 26.3% and 17.3%, respectively. However, we observed that in cases of *support* posts, explicit direct expressions of attitude appear more towards the beginning of the post. In about 53.5% of the support posts that contained explicit direct expression of attitude, the attitude was expressed in the first sentence. In the case of *dispute* posts, the ratio is 30.5%. This can be explained by the discussants trying to be polite when they express disagreement and not start their comment with an explicit disagreement expression. Figure 2.2 shows the ratio of comments (y-axis) in which a direct speech from the post writer to the post recipient appears in the N^{th} sentence (x-axis) for the first time in the post. The graph shows that direct explicit expressions appear more towards the beginning and less towards the end of the post in cases of *support*. The opposite happens in the case of *dispute* posts.

- Emoticons (facial expressions written using punctuation marks) are used significantly more in cases of agreement than in cases of disagreement. The percentage of *dispute* posts that contain at least one emoticon is 4.3%. In the case of *support* posts, the percentage is 15.6%. Surprisingly, the most frequent emoticons in both cases were smiling and laughing faces. The top three emoticons in both types are *:)*, *;)*, and *:D*. We use the Ark Twitter Tokenizer & Tagger [65] to identify emoticons. By examining the *dispute* sentences that contains smiley faces, we noticed that a significant portion of those sentences are sarcastic.

- Both positive and negative opinion expressions are used in both *dispute* and

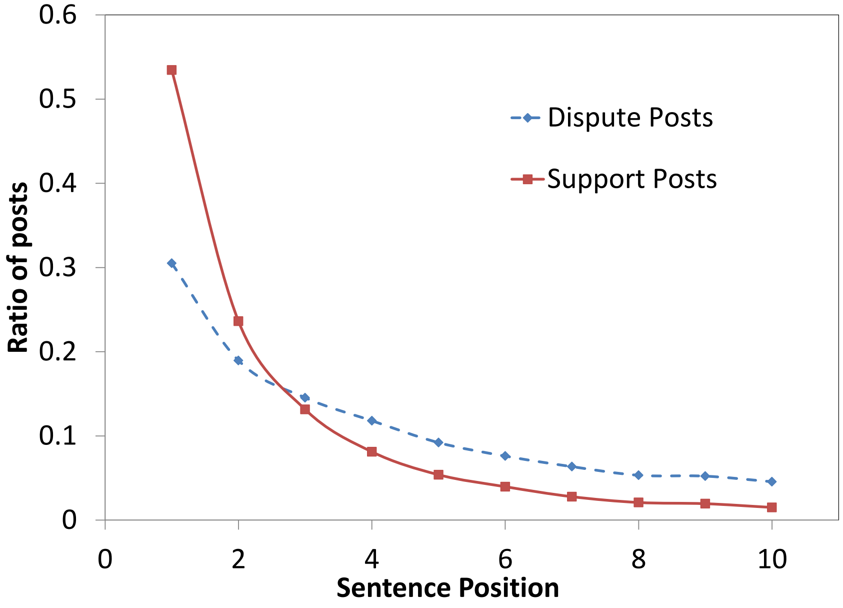


Figure 2.2: Distribution of sentence count support (solid red) and blue (dashed blue) posts. The average sentence count is 7.86 in support posts and 3.68 in dispute posts

support posts. Table 2.2 shows the percentages of positive and negative opinion expressions in *support* and *dispute* comments. These numbers can be explained by the fact that the participants in discussions usually switch back and forth between arguing for their opinions and arguing against the opposing opinions. These numbers show that relying on opinion expression identification *only* is not going to be very helpful for identifying the viewpoints of the participants, because each comment may contain both negative and positive expressions. To detect participant viewpoints we need not only to identify opinion expressions, but also their targets. The numbers in the table also show that if only the sentences that contain both a first person pronoun and a second person pronoun are considered, the percentage of *support* comments that contain positive expressions is higher than *dispute* comments that contain positive expressions. This can be explained by the fact that these sentences usually express direct

	Type	Positive	Negative
Any Sentence	Dispute	70.0%	68.0%
	Support	57.6%	46.6%
1st PP - 2nd PP Sentences	Dispute	55.0%	49.2%
	Support	57.0%	33.1%

Table 2.2: Percentage of posts that contain at least one opinion expression in *dispute* and *support* comments. The top two rows are the numbers when all the sentences are considered. The bottom two rows are when only sentences that contain a first person pronoun and a second person pronoun are considered.

attitude from the post author to the post recipient, hence, it is more likely that the polarity of the opinion expressions used in these sentences match the polarity of the whole comment. The list of opinion cues that we used for this analysis was taken from OpinionFinder [203].

This study of the createdebate data gave us a clearer picture of the discussion/debate genre and the problem that we are addressing in this part of the thesis. It also motivates the need for addressing problems such as handling Out-of-Vocabulary words and identifying opinion targets.

2.4 Overview of the Approach

In this section, we present an overview of the processing steps that we apply to discussions to identify the different perspectives of the discussants. Figure 2.3 shows a block diagram that illustrates the components of the proposed processing pipeline. The input is a discussion thread downloaded from an online discussion forum. The first component parses the thread to identify participants, posts, and the reply structure of the thread. When a discussion thread is downloaded from the internet, it gets stored on the disk in HTML format. We use an HTML parser to extract the needed information from the thread.

The next component analyzes the text of the posts to identify opinion expressions

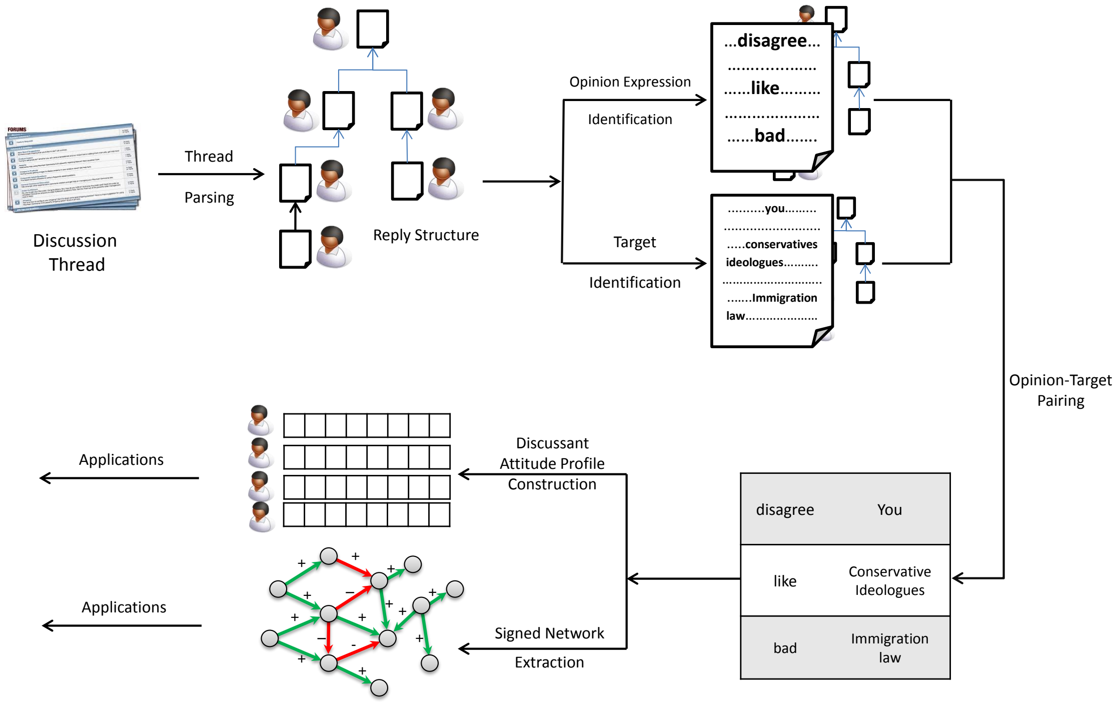


Figure 2.3: A block diagram summarizing the different tasks and applications presented in this part.

and their polarities. As we showed in the previous section, a significant number of the words used in discussions are out-of-vocabulary (OOV) words that can not be found in the existing polarity lexicons. We propose a method for identifying the polarity of OOV words. Our method is an extension to the random walk method used in [76]. We augment the semantic graph described in [76] by adding the OOV words to it and connecting them to words that are semantically related to them. The semantic relatedness is estimated based on co-occurrence statistics computed from a corpus of social text. The polarity of an opinion expression depends on the context in which it appears. Negation is probably the most important contextual feature that affects the polarity of an opinion expression. We propose a method for identifying negation cues and their scope. Our method uses two CRF models, one for identifying negation cues, and one for identifying the scope of each negation cue.

The following component in the pipeline analyzes the text of the posts to identify candidate targets of attitude. Attitude targets can be other participants, aspects of the discussion topic, or named entities mentioned in the discussion. We use noun phrase chunking and named entity recognition techniques to identify candidate targets. We use co-reference resolution to identify all the anaphoric mentions of the identified targets.

The next component in the pipeline takes the output of the last two components and pairs each opinion expression with its target(s). We propose two methods for addressing this problem, a supervised method and an unsupervised method. The unsupervised method uses a set of handcrafted dependency-based rules to determine the target of an opinion expression. The supervised method uses a set of lexical, syntactic, and dependency features to train a model for identifying attitude targets.

The last component in the pipeline uses the extracted targeted attitudes to build a formal representation of the discussion. We experiment with two representations. In the first representation, each participant has an attitude profile that includes an entry for every attitude target. In the second representation, each discussant is represented by a node in a network. The edges in this network represent attitude relations. The sign of the edge is positive if the attitude between the two connected participants is positive, otherwise the sign is negative.

Finally, we present a method for detecting opinion subgroups. The members of each subgroup share the same opinion with the respect to the discussion topics. We experiment with two methods for detecting subgroups. In the first methods, we use a partitioning algorithm to partition the signed network representation of the discussion. Each resulting partition forms a subgroup. In the second method, we use a clustering algorithm to cluster the vector space of all the discussant attitude

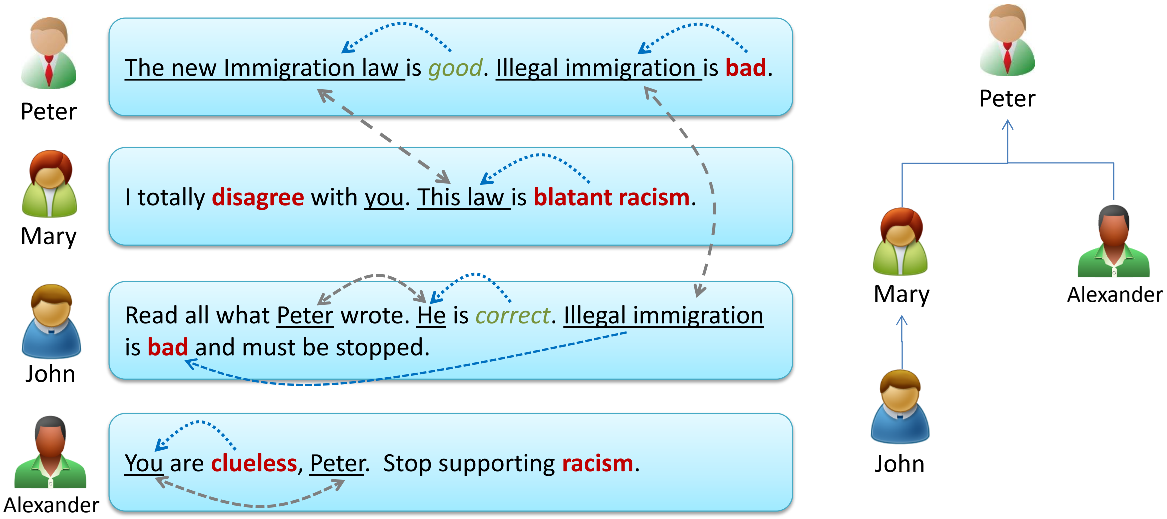

Peter
The new Immigration law is good. Illegal immigration is bad.
Mary
I totally disagree with you. This law is blatant racism.
John
Read all what Peter wrote. He is correct. Illegal immigration is bad and must be stopped.
Alexander
You are clueless, Peter. Stop supporting racism.
Peter
Mary
Alexander
John

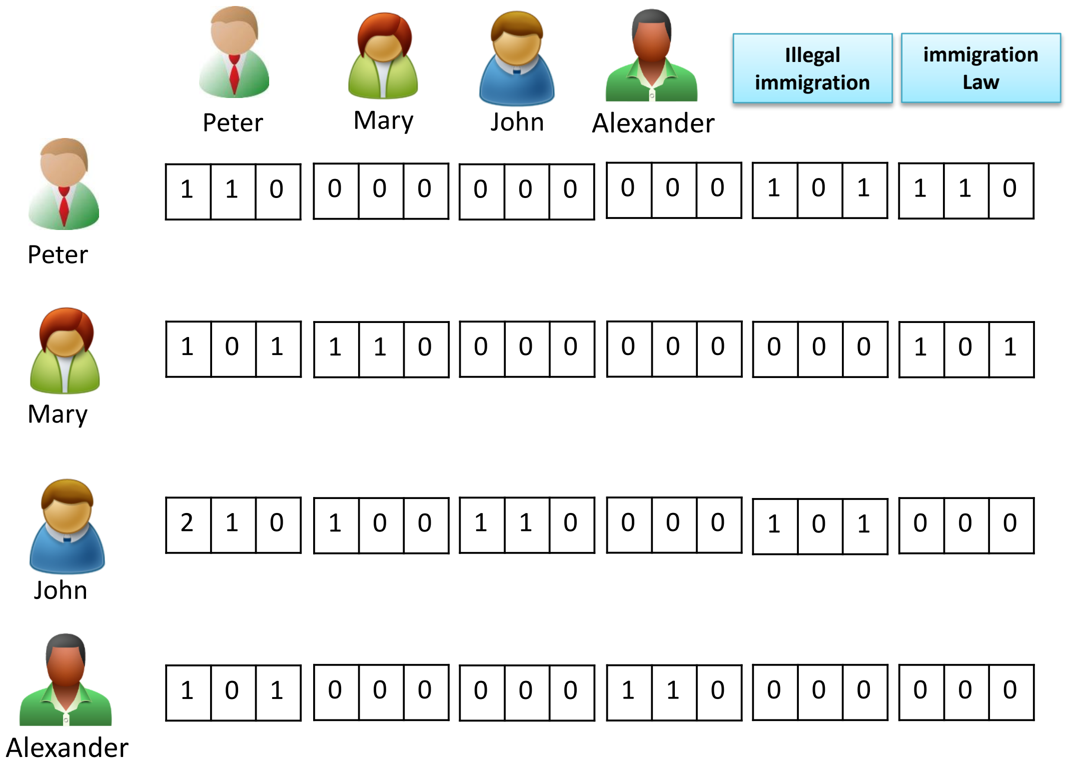
Figure 2.4: An example discussion thread. The figure on right shows the actual posts with opinion expressions and their candidate targets tagged (positive expressions colored in green; negative expressions colored in red; candidate opinion targets underlined. The figure to the left shows the reply structure of the thread.

profiles.

In the following section, we present an example that illustrates the processing steps in our approach.

2.5 An Illustrative Example

Figure 2.4 shows a sample discussion thread with four participants. The discussion topic is about a new immigration law enforced by the government of one of the US states. The right part of the figure shows the reply structure of the thread (i.e. who replies to whom). The first step after parsing the thread, is to identify opinion expression. In Figure 2.4, the words in green (italic) are positive while the words in red (bold) are negative. For example *good* and *correct* are positive opinion words while *bad* and *clueless* are negative words. The next step is identifying the candidate targets of attitudes. The underlined words and phrases in the figure are the candidate



	Peter			Mary			John			Alexander			Illegal immigration			immigration Law		
Peter	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0
Mary	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1
John	2	1	0	1	0	0	1	1	0	0	0	0	1	0	1	0	0	0
Alexander	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Figure 2.5: The attitude vector representation of the example discussion shown in Figure 2.4

targets that our system identifies in this example. Our method identifies the explicit and the anaphoric mentions of each target. The grey (dashed) double-sided arrow connects the phrases/words that refer to the same entity or concept. Next, we associate each opinion expression with its target. The blue dotted arrows in the figure point from opinion expressions to their targets. Finally, we build a vector representation (Figure 2.5) and a signed network representation (Figure 2.6 of the discussion.

In the vector representation, each discussant has a vector. The vector contains three entries for every target in the discussion. The first entry is the total number of mentions of that target by that participant; the second entry is the number of positive mentions; the third is the number of the negative mentions.

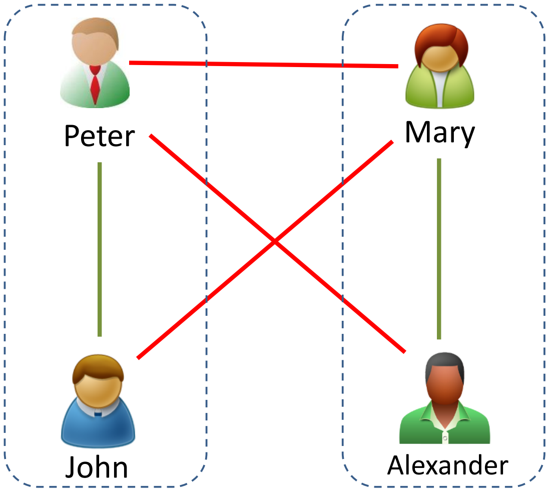


Figure 2.6: The signed network representation of the example discussion shown in Figure 2.4

In the signed network representation, each participants is represented by a node in the network. A positive edge connects two discussants if they carry positive attitude toward one another or if they have similar opinion towards the topics of shared interest. A negative edge connects two participants if they carry negative attitude towards each other or if they have different opinion towards the topics of shared interest. Partitioning the signed network splits the discussant into subgroups where the members of each subgroup have the same perspective with regard to the discussion topic. Figure 2.6 shows that Mary and Alexander, who are against the new law, were put in the same subgroup. Peter and John, who support the new law, were put in another subgroup.

2.6 Outline of Part I

In Chapter III, we describe our method for identifying opinion expressions. We present an extension of a random walk algorithm that allows it to handle OOV terms. In Chapter IV we explain how we handle negation. We propose a supervised method for identifying negation cues, the scope of negation, and the negated events.

In Chapter V, we present an approach for identifying attitude targets and pairing each opinion expression with its target. In chapter VI, we describe an application that uses attitude predictions to detect opinion subgroups in discussions.

CHAPTER III

Identifying Opinion Expressions

In this chapter, we study the problem of identifying the polarity of words; i.e. automatically classifying words as either positive, negative, or neutral. Lehrer [112] defines word polarity as the direction the word deviates to from the norm. For example, the word *beautiful* is positive, while the word *ugly* is negative. Identifying the polarity of individual words is essential for identifying the polarity of larger pieces of text. Identifying text polarity has been shown to be useful in many applications such as analyzing product reviews, building recommendation systems, and identifying attitude.

Automatic identification of word polarity is not a trivial task because of the following challenges. First, existing manually created polarity lexicons (e.g., Stone et al. [178, 177] and Tong [188]) are limited to thousands of words for a limited number of languages. Second, it is difficult, or even impossible, to build a polarity lexicon that covers all domains. The reason is that sentiment expressions vary from one domain to the other. A word can be subjective and positive in one domain but objective or negative in another domain [156]. For example, the word *unpredictable* often expresses positive sentiment in movie reviews (e.g. *unpredictable plot*), however, in car reviews, it is more likely to express negative sentiment (e.g. *unpredictable* *steering*). Third, the polarity of a word depends on its context. For example, the word *fine* is often a positive word when used as an adjective (e.g. *I am doing fine*), but it is a negative word when used as a noun (e.g. *I received a parking fine*). A positive word that appears in a negated context becomes negative and vice versa. Other contextual factors that affect text polarity include hedging, intensification, and neutralization. Forth, the language used in real life applications (especially in social media sites) is very dynamic and contains a significant amount of colloquial words and acronyms (e.g. using *gr8* to mean *great*). Most of these are out-of-vocabulary words that do not exist in polarity lexicons or dictionaries.

Most of these challenges have been addressed in previous work. Several unsupervised, supervised, and semi-supervised methods have been proposed to build polarity lexicons or extend existing ones. In our work, we focus on an understudied challenge: identifying the polarity of out-of-vocabulary (OOV) words in the social media domain.

Our analysis of the createdebate data set in Chapter II shows that OOV words are common in online discussions. Roughly 27.7% of the sentences in the corpus contain at least one OOV word each. Table III shows some OOV word examples. To show the importance of OOV word polarity identification, we calculated the proportion of OOV words in three corpora used for sentiment studies: a set of movie reviews, a set of online discussions from a political forum, and a set of randomly sampled tweets. For each word in the data, we look it up in two standard English dictionaries, together containing $160,000$ unique words. Table 3.1 shows the statistics.

OOV words have a high chance of being polarized because people tend to use informal language or special acronyms to emphasize their attitudes or impress the audience. Therefore, being able to automatically identify the polarity of OOV words

corpus	source	# of words	Percentage of OOV
Movie reviews	3,411 customer reviews from IMDB for the movie *The dark knight (2008)*	10.7 M (9.5 M)	5.3% (2.7%)
Political forum	23 K sentences from *www.politicalforum.com* on various topics	381 K (348 K)	8% (6%)
Tweets	0.6 M random English tweets from twitter.com. (We count a tweet as in English if at least half of the words are English dictionary words. Tags and symbols were removed.)	7.1 M (5.9 M)	30% (27%)

Table 3.1: OOV words proportion in some corpora used for real world applications. (Numbers in parenthesis exclude words whose first letters are capitalized since they are likely to refer to named entities.)

Positive		Negative	
Word	Meaning	Word	Meaning
beautimous	beautiful and fabulous	disastrophy	a catastrophy and a disaster
gr8	great	banjaxed	ruined
buffting	attractive	ijit	idiot

Table 3.2: Examples of positive and negative OOV words

will essentially benefit real-world applications.

We first review the previous work on word polarity identification. Then, we propose an extension of an existing semi-supervised approach that applies a Markov random walk model to a large semantic graph to determine the polarity of any given word. Our extension uses co-occurrence statistics computed from a large corpus of social text to add OOV words to the semantic graph. The random walk method is then used to label the OOV words. We present several variations of this method and compare them experimentally.

3.1 Related Work

Classifying the polarity of words has interested researchers for many years. Philip Stone [178] manually labeled 3,596 words with their polarity and included them in the General Inquirer system that he developed for content analysis of textual data. The full General Inquirer lexicon has 182 categories of word tags and 11,788 words. Words with multiple senses may have multiple entries in the lexicon. Hatzivassiloglou and McKeown (1997) proposed the first automatic method for determining adjective polarities. The method determines the polarity of adjectives by extracting pairs of adjectives conjoined by conjunctions such as *and*, *or* and *but* in a large corpus. Based on the conjunction operator used to connect the two adjectives, they classify the conjoined adjectives as same polarity (e.g. *easy and simple interface*) or opposite polarity (e.g. *clear but difficult question*). The weakness of this method is that it is limited to adjectives. It is normal to see nouns with opposite polarity conjoined with *and* (e.g. *peace and war*).

Wiebe [201] and Wiebe et al. [199] proposed a semi-supervised approach for classifying words into subjective and objective. They cluster words based on their distributional similarity. This method identifies distinguishes between subjective and objective words; it does not attempt to predict the polarity of subjective words.

Turney and Littman [191] proposed a method for inferring the semantic orientation of a word from its statistical association with a set of seed positive and negative paradigm words. They compute the pointwise mutual information (PMI) of each unlabeled word with each of the seed words. The PMI between two words can be seen as an indicator of their semantic association. This method requires access to large corpus of text to perform well. The size of the data set that achieved the best performance for them contained roughly one hundred billion words.

Kamps et al. [96] proposed a method that utilizes WordNet's synonymy relations to construct a network of synonymous words. The polarity of an adjective is decided by computing its shortest paths to two seed words: *good* and *bad*, which are chosen as representatives of positive and negative polarities. Hu and Liu [91] and Kim and Hovy [103] use WordNet synonyms and antonyms to predict the polarity of words. They start with a small number of labeled words. For each unlabeled word, they assign it the polarity of its synonym if this synonym is labeled or the opposite polarity of its antonym if this antonym is labeled. The bootstrapping continues until all possible words are labeled.

Esuli and Sebastiani [57] proposed a method that determines the polarity of words by classifying their gloss definitions. They extract gloss definitions from an online glossary. In another work that exploits glosses, Takamura et al. [182] proposed a method that regards word polarities as spins of electrons. They construct a lexical network by linking two words if one appears in the gloss of the other. They deal with each word in the network as an electron with a directed spin. As inspired by electrons energy theorems, neighboring spins tend to have the same polarity. They use the mean field method to greedily find the best solution. This method has two limitations. First, it assumes that every word has polarity (based on the analogy with electrons). Second, the greedy optimization step sometimes gets trapped in a local optimum.

Mohammad et al. [132] utilize the marking theory which states that overtly marked words such as *dishonest, unhappy* and *impure* tend to have negative semantic orientations whereas their unmarked counterparts: *honest*, *happy* and *pure* tend to have positive semantic orientation. They use a set of 11 antonym-generating affix patterns to generate overtly marked words and their counterparts from the *Macquarie Thesaurus*. After obtaining a set of 2,600 seeds by the affix patterns, they expand the sentiment lexicon using *Roget-like* Thesaurus. Their method does not require seed sentiment words or WordNet, but still needs a comprehensive thesaurus.

The idea of the marking theory is language dependent and cannot be applied from one language to another.

Rao and Ravichandran [162] proposed a method that treats polarity detection as a semi-supervised label propagation problem on a semantic network of words. They experimented with two different networks, one extracted from WordNet and one extracted from OpenOffice thesaurus. They applied their method to the word polarity identification problem in three languages: English, Hindi, and French. Hassan and Radev [77] used a Markov random walk model to estimate the polarity of words in WordNet. The model is capable of assigning a polarity sign and magnitude to each word. This work is described in more detail in the next section. In the rest of the chapter, we present an extension of this method to make it capable of labeling out-of-vocabulary words. In [73], Hassan et al. proposed an extension of their random walk model to other languages. They construct a multilingual network in which same-language words are connected based on synonym relations and co-occurrence statistics, and different-language words are connected using meaning relations extracted from a Foreign-English dictionary.

Velikovich et al. [193] investigated the viability of learning sentiment lexicons semi-automatically from the Web. They use label propagation on a word network extracted from the web. They conclude that label propagation is not suitable when the whole Web is used as a background corpus, since the constructed graph is very noisy and contains many dense subgraphs, unlike the lexical graphs constructed from WordNet. Kanayama and Nasukawa [98] use syntactic features and context coherency (i.e. the tendency for same polarities to appear successively) to detect polar clauses.

3.2 Identifying Word Polarity Using Random Walks

In this section, we summarize the work of Hassan and Radev [77] which we use as a basis for our work. The method starts by constructing a semantic network $G(W, E)$. Each node $w \in W$ represents a pair comprising a word and its part-of-speech tag (e.g *likes/VBZ*). Two nodes are connected if there is a semantic relation between them. Semantic relatedness is determined based on WordNet. Two words are considered semantically related if they appear in the same *synset* in WordNet. If more than one relation exists between any two words, the weight of the corresponding edge is adjusted accordingly.

A small number of nodes in the network are labeled for polarity and used as seed words. The polarity of unlabeled nodes is determined using a random walk model applied to the constructed network. To assign a label to an unlabeled node w_u, a random walk is started from w_u and continues until the random agent reaches a labeled node. This process is repeated many times and each time the number of steps the random agent took from w_u to a labeled node is recorded. This number is refereed to by the term *hitting time*. To determine the polarity of the unlabeled node, the average hitting time to positive nodes $P(w_n|+)$ and the average hitting time to negative nodes $P(w_n|-)$ are compared. If $P(w_n|+)$ is larger than $P(w_n|-)$, then w_n labeled *positive*, otherwise it is labeled *negative*. Radev and Hassan reported 93.1% cross-validation accuracy on the General Inquirer lexicon [178].

This method works only if the semantic relatedness between words is known and can be represented by edges in the network. Hassan and Radev used WordNet to get relatedness relations. In other studies, gloss definitions from dictionaries or co-occurrence statistics in large text corpora were used to infer relatedness [182, 57, 162].

The algorithm proposed by Hassan and Radev classifies each word as either positive or negative. They stated that their algorithm can be configured to abstain from classifying a word if the difference between the mean hitting time to positive seeds and the mean hitting time to negative seeds is very small. In the following section, we show how we modify their algorithm to to do 3-way classification: positive, negative, and neutral. We also run describe an experiment to evaluate the performance of this 3-way classifier. In section 3.4, we present our method for identifying the polarity of OOV words.

3.3 3-Ways Word Polarity Classification

Classifying a word as positive or negative assumes that the word is subjective and polarized. In real applications, the majority of words that appear in the text are neutral. Classifying all words as positive or negative. This motivates the need for a general algorithm that can handle the three polarity classes: positive, negative, and neutral. We modify the algorithm presented in Hassan and Radev [77] such that it classifies a word as neutral if the polarity magnitude is not big enough to classify the word as positive or negative with hight confidence. We compute the ratio between the mean hitting time to positive words and the mean hitting time to negative words. If this ratio is less than some threshold γ, the word is classified as neutral, otherwise it is classified as positive or negative. Algorithm 3.3 is a modified version of the algorithm presented in Hassan and Radev's paper.

3.4 Out-of-Vocabulary (OOV) Words Polarity Identification

Consider the graph $G(W, E)$ we described in Section 3.2. So far, the only resource we use to construct the graph is WordNet synsets. The first step in our approach to determining the polarity of an OOV word is to find the words in WordNet that are semantically related to it. We add the OOV word to G by creating a new node and

Algorithm 1 3-Class Word Polarity using Random Walks (parameter $\gamma : 0 < \gamma < 1$)

```
Require: A word relatedness graph $G$
Given a word $w$ in $V$
Define a random walk on the graph. the transition probability between any two nodes $i$, and $j$ is defined as: $P_{t+1|t}(j|i) = Wij / \sum_k W_{ik}$
Start $k$ independent random walks from $w$ with a maximum number of steps $m$
Stop when a positive word is reached
Let $h^*(w|S^+)$ be the estimated value for $h(w|S^+)$
Repeat for negative words computing $h^*(w|S^-)$
if $h^*(w|S^+) \leq \gamma h^*(w|S^-)$ then
    Classify $w$ as positive
else if $h^*(w|S^-) \leq \gamma h^*(w|S^+)$ then
    Classify $w$ as negative
else
    Classify $w$ as neutral
end if
```

linking it to the words that are semantically related to it. Once we have the extended network constructed, we use the random walk model described in Section 3.2 to predict the polarity of each OOV word.

3.4.1 Mining Word Relatedness from the Web

There are several methods that can be used to determine the relatedness of words. Agirre et al. [18] studied the strengths and weaknesses of the different approaches used for identifying term similarity and relatedness. They noticed that lexicographical methods such as the WordNet suffer from the limited coverage of the lexicon, which is the case here with OOV words. We use a web-based, distributional approach to find the set of words that are semantically related to an OOV word. We perform a web search using the OOV word as the search query and retrieve the top S search results. We extract the textual content of the retrieved results and tokenize it. After removing all the stop words, we compute the number of times each word co-occurs with the OOV word in the same document. We rank the words based on their co-occurrence frequency and return the top R words as the set of related words to the given OOV word.

We experimented with three different variants of this approach. In the first variant, the frequency values of related words are normalized by the lengths of the documents that contributed to the count of each related word. The intuition here is that longer documents contain more words and hence the probability that a word in the that document is related to the OOV word is lower than when the document is shorter.

In the second variant, we only consider the words that appear near the OOV word (i.e. within d words to the left and right of the OOV word) when we compute the co-occurrence frequency. The intuition here is that words that appear near the OOV word are more likely to be semantically related than the words that appear further away.

In the third variant, instead of searching the entire web, we limit the search to social text. In the experiments described below, we search for an OOV word in twitter[1]. The intuition here is that searching the entire web is likely to return results that do not necessarily contain sentimental text. Moreover, the meaning of many words depends on the context. The same word that is positive in one context may be neutral in another context. The text written in a social context is more likely to carry sentiment and express emotions. This helps us find better related words for our task.

3.4.2 Extending the Semantic Network With OOV Words

We start with the graph $G(W, E)$ constructed from Wordnet synsets. For each new OOV word that does not exist in G, we create a new node w. We set the part of speech tag of w to *unspecified.* Then we use the method described in the previous subsection to find a set of words that are most related to w. Finally, we create a link

[1] http://www.twitter.com

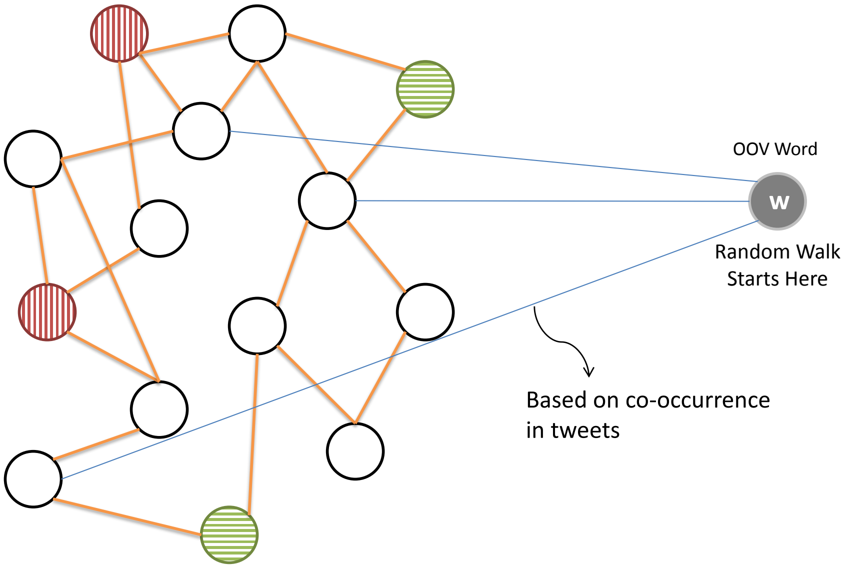


Figure 3.1: Illustration of the random walk algorithm and the extension we add for OOV words. Labeled nodes are filled with patterns; horizontal (green) lines for positive and vertical (red) lines for negative)

between each OOV word and each of its related words. To predict the polarity of an OOV word, we use the random walk model described above.

3.5 Evaluation

3.5.1 General Purpose Three-Way Classification

The experiments described so far all use the General Inquirer lexicon, which contains a well-established gold standard data set of positive and negative words. However, in realistic applications, a general purpose list of words will frequently have neutral words that don't express sentiment polarity. To evaluate the effectiveness of the random walk method in distinguishing polarized words from neutral words, we constructed a data set of 2000 words randomly picked from a standard English dictionary[2] and hand labeled them with three classes: positive, negative and neutral. Among the 2000 words, 494 were labeled positive, 491 negative, and 1015 neutral.

[2] Very infrequent words were filtered out by setting a threshold on IDF of the words in a corpus.

Class	Positive	Negative	Neutral	Overall
Accuracy	68.0	82.1	80.6	77.9

Table 3.3: Accuracy for 3 classes on a general purpose list of 2000 words

The distribution among different parts of speech is 532 adjectives, 335 verbs, 1051 nouns and 82 others.

We used the semi-supervised setting with General Inquirer lexicon polarized word list as training set. Since the 2000 test set has some portion of polarized words overlapping with the training set, we excluded the words that appear in test set from the training set. We ran Algorithm 2 in section 3.3 with parameters $\gamma = 0.8, m = 15, k = 1000$. The overall accuracy as well as the precision for each class is shown in table 3.5.1. We can see that the accuracy of the positive class is much lower than the negative class, due to the many positive words classified as neutral. This means that the average confidence of negative words is higher than positive words. One factor that could have caused this is the bias originated from the training set. Since there are more negative seeds than positive ones, the constructed graph has an overall bias towards negative class.

3.5.2 OOV Word Polarity

We created a labeled set of 300 positive and negative OOV words. We asked a native English speaker to examine a large number of threads posted on several online forums and identify OOV words and determine their polarities. Some examples of positive/negative OOV words are listed in Table III.

We used the approach described in Section 3.4 to automatically label the words. We used the words of the General Inquirer lexicon as labeled seeds. We sat the maximum number of steps $m = 15$ and the number of samples $k = 1000$. We

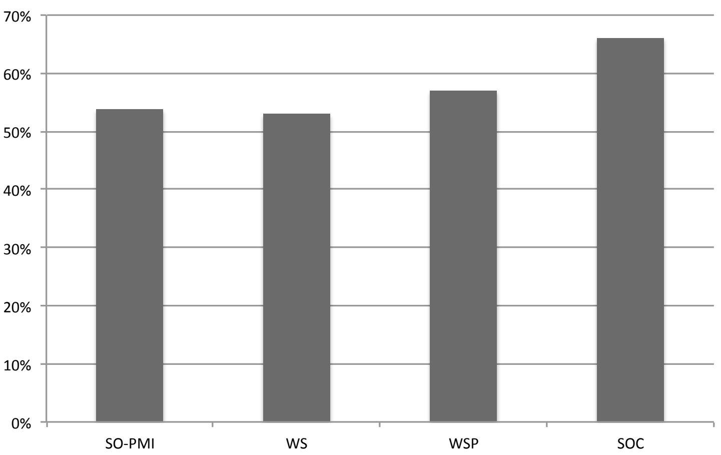


Figure 3.2: Comparison of the accuracy of OOV word polarity identification using three variations of our approach and SO-PMI

experimented with the three variants we proposed for extracting the related words as described in Section 3.4. We give the experimental setup for each variant here:

1. Search the entire web (WS): We used Yahoo search[3] to run the search queries. For each OOV word, we retrieve the top 500 results and use them to extract the related words.

2. Search the entire web and limit the extraction of related words to the proximity of the OOV word (WSP): We fix the proximity of a given OOV word to 15 words before and 15 words after (we experimented with different ranges but no significant changes were observed).

3. Limit the search to social content (SOC): We limit the search for OOV words to tweets posted on Twitter. We use the Twitter search API to send the search queries. For each OOV word, we retrieve the top 10,000 tweets. Each tweet is maximum 140 characters long.

We also compare these variations to the SO-PMI method with the same 14 seeds

[3] http://www.yahoo.com

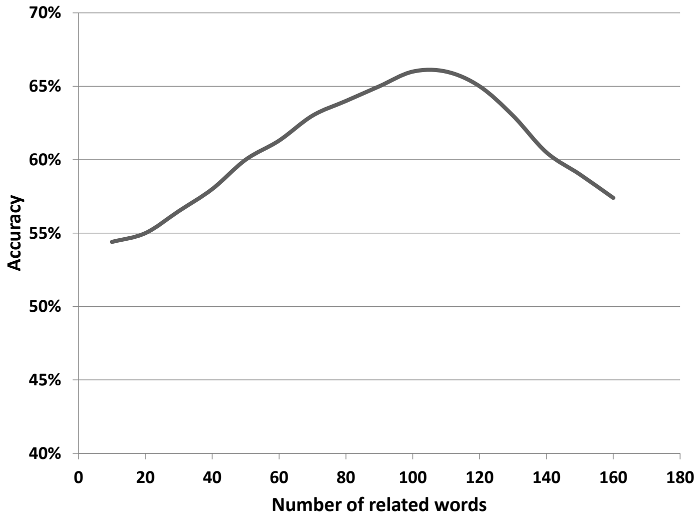


Figure 3.3: The effect of varying the number of extracted related words on accuracy

used in [190]. The SO-PMI value can be calculated as follows:

$$\text{SO-PMI}(w) = \log \frac{hits_{w,pos} \times hits_{neg}}{hits_{w,neg} \times hits_{pos}} \tag{3.1}$$

Figure 3.2 shows the results of the three variations. The results show that extracting related words from tweets gives the best accuracy. This corroborated our intuition that using social content is more likely to provide sentimental related words.

The accuracy changes by varying R, the number of related words extracted for each OOV word. The results shown in the figure correspond to $R = 90$.

To better understand the impact of varying this parameter, we ran the experiment that uses Twitter to extract related words several times using different values for R. Figure 3.3 shows how the accuracy of polarity prediction changes as R changes.

CHAPTER IV

Handling Negation

Negation is a pervasive and intricate linguistic phenomenon present in all languages [189, 89]. The automatic detection of negation and its scope is a problem encountered in a wide range of natural language processing applications including, but not limited to, data mining, relation extraction, question answering, and sentiment analysis. Detecting and handling negation is important in opinion mining and sentiment analysis systems. If a positive word appears in a negated context, it should actually be treated as negative and vice versa. Handling negation involves detecting negation cues and the scope of each cue in the sentence. A negation cue is a word, a phrase, a prefix, or a postfix that triggers negation. The scope of negation is the part of the meaning that is negated [92]. For example, in the sentence below ***never*** is the negation cue. The negation scope is enclosed in square brackets.

[John] ***never*** *[liked smart phones].*

This example also shows that the polarity of the verb *liked* becomes negative since it falls in a negation scope. Cues and scopes may be discontinuous. For example, the negation cue *neither ... nor* is discontinuous.

In this chapter, we present a system for automatically detecting negation cues and their scopes in English text. The system uses a conditional random field (CRF) model trained on labeled sentences extracted from two classical English novels. The CRF model is trained using lexical, structural, and syntactic features. This chapter is based on the work submitted in [12].

The rest of this chapter is organized as follows. Section 4.1 reviews previous work. Section 4.2 describes the data and the annotation process. Section 4.3 describes the CRFs models that we build. Section 4.4 presents the evaluation and the results.

4.1 Related Work

Most research on negation has been done in the biomedical domain [39, 140, 101, 136, 135, 17, 133, 163], mostly on clinical reports. The reason is that most NLP research in the biomedical domain is interested in automatically extracting factual relations and information from unstructured data. Negation detection is important here because information that falls in the scope of a negation cue cannot be dealt with as factual.

Chapman et al. [39] a rule-based algorithm called NegEx for determining whether a finding or disease mentioned within narrative medical reports is present or absent. The algorithm uses regular expression based rules. Mutalik et al. [140] developed another rule based system called Negfinder that recognizes negation patterns in medical text. It consists of two components: a lexical scanner, *lexer* that uses regular expression based rules to generate a finite state machine, and a parser. Morante [137] proposed a supervised approach for detecting negation cues and their scopes in biomedical text. Their system consists of two memory-based engines, one that decides if the tokens in a sentence are negation signals, and another that finds the full scope of these negation signals.

Negation has been recently studied in the context of sentiment analysis [204, 95, 45, 83, 88]. Wiegand et al. [202] surveyed the recent work on negation scope detection for sentiment analysis. Wilson et al. [204] studied the contextual features that affect text polarity. They used a machine learning approach in which negation is encoded in several features. One feature checks whether a negation expression occurs in a fixed window of four words preceding the polar expression. Another feature accounts for a polar predicate having a negated subject. They also have disambiguation features to handle negation words that do not function as negation cues in certain contexts, e.g. *not to mention* and *not just*.

Jia et al. [95] proposed a rule based method to determine the polarity of sentiments when one or more occurrences of a negation term such as not appear in a sentence. The hand-crafted rules are applied to syntactic and dependency parse tree representations of the sentence.

Hogenboom et al. [88] found that applying a simple rule that considers two words, following a negation keyword, to be negated by that keyword, to be effective in improving the accuracy of sentiment analysis in movie reviews. This simple method yields a significant increase in overall sentiment classification accuracy and macro-level F1 of 5.5% and 6.2%, respectively, compared to not accounting for negation.

4.2 Data

We use the data set distributed by *sem Shared Task 2012 on resolving the scope and focus on negation. This data set includes two stories by Conan Doyle, The Hound of the Baskervilles, The Adventures of Wisteria Lodge for training and development. All occurrences of negation are annotated accounting for negation expressed by nouns, pronouns, verbs, adverbs, determiners, conjunctions and prepositions. For each negation cue, the negation cue and scope are marked, as well as the negated event, if any. Cues and scopes may be discontinuous. The annotation guidelines follow the proposal of Morante et al. [138][1]. The data is split into three sets: a training set containing 3,644 sentences, a development set containing 787 sentences, and a testing set containing 1,089 sentences. The data are provided in CoNLL format. Each line corresponds to a token and each annotation is provided in a column; empty lines indicate end of sentence. The content of the columns given is:

- Column 1: chapter name
- Column 2: sentence number within chapter
- Column 3: token number within sentence
- Column 4: word
- Column 5: lemma
- Column 6: part-of-speech
- Column 7: syntax
- Columns 8 to last:
  - If the sentence has no negations, column 8 has a "***" value and there are no more columns.
  - If the sentence has negations, the annotation for each negation is provided in three columns. The first column contains the word or part of the word (e.g., morpheme "un"), that belongs to the negation cue. The second contains the word or part of the word that belongs to the scope of the negation cue. The third column contains the word or part of the word that is the negated event or property. It can be the case that no negated event or property are

[1] http://www.clips.ua.ac.be/sites/default/files/ctrs-n3.pdf

Token	Lemma	POS	Syntax	Cue 1	Scope 1	Event 1	Cue 2	Scope 2	Event 2
She	She	PRP	(S(NP*)	-	She	-	-	-	-
would	would	MD	(VP*	-	would	-	-	-	-
not	not	RB	*	not	-	-	-	-	-
have	have	VB	(VP*	-	have	-	-	-	-
said	say	VBD	(VP*	-	said	-	-	-	-
‘	‘	“	(SBAR(S(NP*	-	’	-	-	-	-
Godspeed	Godspeed	NNP	*	-	Godspeed	-	-	-	-
’	’	”	*)	-	’	-	-	-	-
had	have	VBD	(VP*	-	had	-	-	had	-
it	it	PRP	(ADVP*	-	it	-	-	it	-
not	not	RB	*)	-	not	-	not	-	-
been	be	VBN	(VP*	-	been	-	-	been	-
so	so	RB	(ADVP*))))))))	-	so	-	-	so	-
.	.	.	*)	-	-	-	-	-	-

Table 4.1: Example sentence annotated for negation following se* shared task 2012 format

marked as negated. For example, in Example 3 none of the negations has a negated event annotated because of the conditional construction.

Tokenization and information for columns 4 to 7 has been obtained by processing the corpus with the Shalmaneser semantic parser [54]. Shalmaneser produces the xml-format needed to annotate the corpus with negation with the Salto Tool [36]. The xml was converted into CoNLL format. Shalmaneser calls the Collins parser [44] to produce syntactic parse trees.

Table 4.1 shows an example of an annotated sentence that contains two negation cues.

4.3 Approach

The problem that we are trying to solve can be split into two tasks. The first task is to detect negation cues, words that trigger negation. The second task is to identify the scope of each detected negation cue. We use a machine learning approach to address both tasks. We train a Conditional Random Field (CRF) [107] model on lexical, structural, and syntactic features extracted from the training dataset. In following two subsection, we describe the CRF models we build for each of the two tasks.

4.3.1 Negation Cue Detection

Negation cues are lexical elements that indicate the existence of negation in a sentence. From lexical point of view, negation cues are four types: 1) Prefix (i.e. in-, un-, im-, il-, dis-). For example, **un-** in ***un****suitable*) is a prefix negation cue. 2) Postfix (i.e. -less). for example, **-less** in care**less**. 3) Multi-word negation cues such as *neither...nor*, *rather than*, *by no means*, etc. 4) Single word negation cues such as *not*, *no*, *none*, *nobody*, etc.

The goal of this task is to detect negation cues. We pose this problem as a sequence labeling task. The reason for this choice is that some negation cues may not indicate negation some contexts. For example, the negation cue *not* in the phrase *not to mention* does not indicate negation. Also, as we saw above, some negation cues consists of multiple words. We train a CRF model on the sentences included in the training set. The token level features that we train the model on are:

- *Token*: The word or the punctuation mark as it appears in the sentence.
- *Lemma*: The lemmatized version of the token.
- *Part-Of-Speech tag*: The part of speech tag of the token.
- *Part-Of-Speech tag category*: Part-of-speech tags reduced into 5 categories: Adjective (ADJ), Verb (VB), Noun (NN), Adverb (ADVB), Pronoun (PRO), and other (OTH).
- *Is punctuation mark*: This feature takes the value 1 if the token is a punctuation mark and 0 otherwise.

- *Starts with negation prefix*: This feature takes the value 1 if the token is a word that starts with un-, in-, im-, il-, or dis- and 0 otherwise.
- *Ends with negation postfix*: This feature takes the value 1 if the token is a word that ends with -less and 0 otherwise.

The CRF model that we use considers at each token the features of the current token, the two preceding tokens, and the two proceeding tokens. The model also uses token bigrams and trigrams, and part-of-speech tag bigrams and trigrams as features.

The labels are 5 types: ”O” for tokens that are not part of any negation cue; ”NEG” for single word negation cues; ”PRE-NEG” for prefix negation cue; ”POST-NEG” for postfix negation cue; and ”MULTI-NEG” for multi-word negation cues. Table 4.2 shows an example labeled sentence.

4.3.2 Negation Scope Detection

Scope of negation is the sequence of tokens (can be discontinuous) that express the meaning that is meant to be negated by a negation cue. A sentence may contain zero or more negation cues. Each negation cue has its own scope. It possible that the scope of two negation cues overlap. We use each negation instance (i.e. each negation cue and its scope) as one training example. Therefore, a sentence that contains two negation cues results in two training examples. We train a CRF model on features extracted from all negation instances in the training dataset. The features that we use are:

- *Token*: The word or the punctuation mark as it appears in the sentence.
- *Lemma*: The lemmatized version of the token.
- *Part-Of-Speech tag*: The part of speech tag of the token.

Token	Lemma	Is Punc.	POS Cat.	POS	Neg Label
Since	Since	0	OTH	IN	O
we	we	0	PRO	PRP	O
have	have	0	VB	VBP	O
been	be	0	VB	VBN	O
so	so	0	ADVB	RB	O
unfortunate	unfortunate	0	ADJ	JJ	PRE-NEG
as	as	0	ADVB	RB	O
to	to	0	OTH	TO	O
miss	miss	0	VB	VB	O
him	him	0	PRO	PRP	O
and	and	0	OTH	CC	O
have	have	0	VB	VBP	O
no	no	0	OTH	DT	NEG
notion	notion	0	NOUN	NN	O
of	of	0	OTH	IN	O
his	his	0	PRO	PRP$	O
errand	errand	0	NOUN	NN	O
,	,	1	OTH	,	O
this	this	0	OTH	DT	O
accidental	accidental	0	ADJ	JJ	O
souvenir	souvenir	0	NOUN	NN	O
becomes	become	0	VB	VBZ	O
of	of	0	OTH	IN	O
importance	importance	0	NOUN	NN	O
.	.	1	OTH	.	O

Table 4.2: Example sentence labeled for negation cue detection

- *Part-Of-Speech tag category*: Part-of-speech tags reduced into 5 categories: Adjective (ADJ), Verb (VB), Noun (NN), Adverb (ADVB), Pronoun (PRO), and other (OTH).

- *Is punctuation mark*: This feature takes the value 1 if the token is a punctuation mark and 0 otherwise.

- *Type of negation cue*: Possible types are: ”NEG” for single word negation cues; ”PRE-NEG” for prefix negation cue; ”POST-NEG” for postfix negation cue; and ”MULTI-NEG” for multi-word negation cues.

- *Relative position*: This feature take the value 1 if the token position in the sentence is before the position of the negation cue, 2 if the token position is after the position of the negation cue, and 3 if the token is the negation cue itself.

- *Distance*: The number of tokens between the current token and the negation cue.

- *Same segment*: This feature takes the value 1 if this token and the negation cue fall in the segment in the sentence. The sentence is segmented by punctuation marks.

- *Chunk*: This feature takes the value NP-B (VP-B) if this token is the first token of a noun (verb) phrase, NP-I (VP-I) if it is inside a noun (verb) phrase, NP-E (VP-E) if it is the last token of a noun (verb) phrase.

- *Same chunk*: This feature takes the value 1 if this token and the negation cue fall in the same chunk (noun phrase or verb phrase).

	gold	system	tp	fp	fn	precision	recall	F1
Cues	264	250	232	14	32	94.31	87.88	90.98
Scope (tokens)	1805	1716	1456	260	349	84.85	80.66	82.70
Scope (full)	249	227	126	14	123	90.00	50.60	64.78

Table 4.3: Results of negation cue negation scope detection

- *Is negation*: This feature takes the value 1 if this token is a negation cue, and 0 otherwise.

- *Syntactic distance*: The number of edges in the shortest path that connects the token and the negation in the syntactic parse tree.

- *Common ancestor node*: The type of the node in the syntactic parse tree that is the least common ancestor of this token and the negation cue token.

The CRF model considers the features of 4 tokens to the left and to the right at each position. It also uses bigram and trigram combinations of some of the features.

4.4 Evaluation

We use the testing set described in Section 4.2 to evaluate the system. The testing set contains 1089 sentences 235 of which contains at least one negation.

We use the standard precision, recall, and f-measure metrics to evaluate the system. We perform the evaluation on different levels:

- Cues: the metrics are computed only for cue detection.

- Scope (tokens): the metrics are calculated at token level counting as tokens the total number of scope tokens. If a sentence has 2 scopes, one with 5 tokens and another with 4, the total number of scope tokens is 9.

- Scope (full): the metrics are calculated at scope level. Both the negation cue and the whole scope should be correctly identified. If a sentence contains 2 negation cues, then 2 scopes are checked.

Table 4.3 shows the results of our experiments.

CHAPTER V

Identifying Opinion Targets

In this chapter, we study targets of attitude in discussions. A target of attitude may be another discussant, an entity mentioned in the discussion, or an aspect of the discussion topic. When the target of opinion is another discussant, either the discussant name is mentioned explicitly or a second person pronoun (i.e. *you, yours, yourself, yourselves*) is used to indicate that the opinion is targeting the recipient of the post. For example, in snippet (2) in the conversation below, the second person pronoun *you* indicates that the opinion word *disagree* is targeting *Discussant 1*, the recipient of the post.

(1) Discussant 1: The new immigration law is good. Illegal immigration is bad.

(2) Discussant 2: I totally disagree with you. This law is blatant racism, and quite unconstitutional.

The target of opinion can also be a subtopic or an entity mentioned in the discussion. For example, "*the new immigration law*", "*Illegal Immigration*", and "*This law*" are all targets of attitude, representing two subtopics: *a new immigration law* and *illegal immigration.* In this chapter, we describe how we identify candidate targets of opinion and how we pair opinion expressions identified using the method described in Chapter III with their targets. This work is based on the work published in [15]. We start by reviewing previous work on this problem followed by a description of our approach.

5.1 Related Work

Several methods have been proposed for identifing the target of an opinion expression. Most of the work has been done in the context of product reviews mining [91, 105, 127, 180]. In this context, opinion targets usually refer to product features (i.e. product components or attributes, as defined by Liu [115]).

In the work of Hu and Liu [91], they treat frequent nouns and noun phrases as product feature candidates. In our work, we extract as targets frequent noun phrases and named entities that are used by two ore more different discussants. Scaffidi et al. [168] propose a language model approach to product feature extraction. They assume that product features are mentioned more often in product reviews than they appear in general English text. However, such statistics may not be reliable when the corpus size is small. Somasundaran and Wiebe [175] present a rule-based method for pairing opinion expressions with the features of a product (e.g. screen, os, etc.). They mine the web to learn associations between products and their features.

Hassan et al. [75] present a method for identifying sentences that carry attitude from the text writer toward the text recipient. They define attitude as the mental position of one participant with regard to another participant. A detailed survey that covers techniques and approaches in sentiment analysis and opinion mining could be found in [146].

In another related work, Jakob and Gurevych [94] showed that resolving the anaphoric links in the text significantly improves opinion target extraction. In our work, we use anaphora resolution to improve opinion-target pairing as shown in Section 5.2 below.

5.2 Approach

In this section, we describe a component that takes as input a discussion thread split into posts and has the reply structure and the opinion expressions in each post identified, The output is a set of attitude targets and the opinion words associated with each of them.

5.2.1 Target Identification

The goal of this step is to identify the possible targets of opinion in a discussion. A target could be another discussant, a subtopic, or an entity mentioned in the discussion.

Identifying Discussant Targets

We use string matching to identify discussant targets. If a post B is a response to post A and post B contains a second person pronoun (*you, yours*, or *yourself*), we infer that the author of B is targeting the author of A. Likewise, if a post C contains the screen name of anther participant in the discussion, we infer that the author of C is targeting that participant. We consider a discussant as a candidate target only if the discussants mention appears in the same sentence with a polarized expression.

Identifying Subtopic Targets

The target of opinion can also be a subtopic. Subtopic targets often appear as noun groups (e.g. The new immigration law). We use shallow parsing to identify noun groups (NG). Dealing with all noun groups as subtopics results in a lot of noise. Therefore, we put two restrictions on noun groups that we consider as targets. First, the noun group should be mentioned by at least two different discussants. Second, the length of the noun group should be 2 or more words.

Named Entities	Subtopics
Barack Obama	The republican nominee
Middle East	The maverick economists
Bush	Conservative ideologues
Bob McDonell	The Nobel prize
Iraq	The federal government

Table 5.1: Some of the entities identified using NER and NP Chunking in a discussion thread about the US 2012 elections

Identify Entity Targets

The third type of targets is named entities that are mentioned in the discussion. The named entities that we consider are persons (e.g. Bill Gates), organizations (e.g. Microsoft Corporation), and locations (e.g. United States). We impose no length restriction on named entities that are considered as targets, but we require that an entity be mentioned by at least two different discussants.

Table 5.1 lists some examples of subtopics and entities identified in a discussion about the 2012 elections in the United States of America.

Co-reference Resolution

A challenge that always arises when performing text mining tasks at this level of granularity is that the same subtopic or named entity may be referred to using different phrases. For example, the noun group, "the new immigration law" in snippet (1) above and the noun group, "*this law*" in snippet (2) both refer to the same thing. It is also common to refer to topics and entities using anaphorical pronouns. For example, the following snippet contains an explicit mention of the entity *Obama* in the first sentence, and then uses a pronoun to refer to the same entity in the second sentence. The opinion word *unbeatable* appears in the second sentence and is associated with the pronoun *He*. In the next section, it will become clear why knowing which entity does the pronoun *He* refers to is essential for opinion-target pairing.

(3) It doesn't matter whether you vote for Obama. He is unbeatable.

Jakob and Gurevych [94] showed experimentally that resolving anaphoric links in text significantly improves opinion target extraction. We use the Stanford CoreNLP API.[1] for noun group identification, named entity recognition, and coreference resolution.

5.2.2 Opinion-Target Pairing

At this point, we have all the opinion words and the potential targets identified separately. The next step is to determine which opinion word is targeting which target. We propose three approaches for this problem: one unsupervised that uses dependency rules and two supervised that use classification and sequence labeling techniques. In the following subsections, we describe the different methods.

5.2.3 Rule-based Opinion-Target Pairing

In this section, we describe a rule based approach for opinion-target pairing. Our rules are based on the dependency relations that connect the words in a sentence. We use the Stanford Parser [104] to generate the dependency links of each sentence in the thread. An opinion word and a target are paired together if they stratify at least one of our dependency rules. Table 5.2 illustrates some of these rules. The rules basically examine the types of the dependencies on the shortest path that connect the opinion word and the head of the target in the dependency parse tree. It has been shown in previous work on relation extraction that the shortest dependency path between any two entities captures the information required to assert a relationship between them [35].

[1] http://nlp.stanford.edu/software/corenlp.shtml

ID	Rule	In Words	Example
R1	$OP \rightarrow nsubj \rightarrow TR$	The target *TR* is the nominal subject of the opinion word *OP*	TARGET$_{TR}$ is good$_{OP}$.
R2	$OP \rightarrow dobj \rightarrow TR$	The target *T* is a direct object of the opinion *OP*	I hate$_{OP}$ TARGET$_{TR}$
R3	$OP \rightarrow prep\_{*} \rightarrow TR$	The target *TR* is the object of a preposition that modifies the opinion word *OP*	I totally disagree$_{OP}$ with you$_{TR}$.
R4	$TR \rightarrow amod \rightarrow OP$	The opinion is an adjectival modifier of the target	The bad$_{OP}$ TARGET$_{TR}$ is spreading lies
R5	$OP \rightarrow nsubjpass \rightarrow TR$	The target *TR* is the nominal subject of the passive opinion word *OP*	TARGET$_{TR}$ is hated$_{OP}$ by everybody.
R6	$OP \rightarrow prep\_{*} \rightarrow poss \rightarrow TR$	The opinion word *OP* connected through a *prep_∗* relation as in *R*2 to something possessed by the target *TR*	The main flaw$_{OP}$ in your$_{TR}$ analysis is that it's based on wrong assumptions.
R7	$OP \rightarrow dobj \rightarrow poss \rightarrow TR$	The target *TR* possesses something that is the direct object of the opinion word *OP*	I like$_{OP}$ TARGET$_{TR}$'s brilliant ideas.
R8	$OP \rightarrow csubj \rightarrow nsubj \rightarrow TR$	The opinion word *OP* is a causal subject of a phrase that has the target *TR* as its nominal subject	What TARGET$_{TR}$ announced was misleading$_{OP}$.

Table 5.2: Examples of the dependency rules used for opinion-target pairing.

5.2.4 Classification-based Opinion-Target Pairing

In this approach, we pose the problem as a classification problem. For every possible opinion-target pair (OP_i, TR_j) in a sentence, we use a machine learning classifier to determine if TR_j is the target of OP_i. We train an SVM model on a set of sentences with labeled opinion-target pairs. We use a set of syntactic, structural, and dependency features that capture the relation between the opinion word and the candidate target. The features are as follows:

- Distance: the number of words between the position of the opinion word and the position of the head word of the candidate target. For example, in sentence (4) below, the distance between the opinion word *good* and the head of the target *law* is 8.

  (4) The new law, which will be enforced starting next month, is ***good*** *for our community.*

- Relative Position: whether the candidate target occurs in the sentence before or after the opinion word. In sentence (4) above the target occurs before the opinion expression.

- Dependency path: all the dependency relations on the shortest path that connects the opinion word to the head word of target. In sentence (4) above the opinion expression *good* is directly connected to the head *law* through the relation: $nsubj(good, law)$. In this case, this feature take the value $OP \overset{nsubj}{\to} TR$.

- Dependency path length: the length of the shortest dependency path that connects the opinion word and the head word. For example, in sentence (4) above, the length of the shortest dependency path between *good* and *law* is 1.

- Syntactic path: all the nodes on the shortest path in the syntactic parse tree that connects the opinion word and the head of the target. In sentence (4) above, the path is: $NN - NP - NP - S - VP - ADJP - JJ$.

- Syntactic path length: the length of the shortest syntactic path described in the previous feature.

- Opinion word part-of-speech: The part-of-speech tag of the opinion word. In the example above, the POS tag of *good* is JJ.

- Least common ancestor: the least common ancestor node of both the opinion word and the target head in the syntactic parse tree. In the example above, the common ancestor node for *good* and *law* is S.

We extract one feature vector for every possible opinion word and candidate target pair in the sentence. For example, if a sentence contains 2 opinion words and 3 candidate targets, we extract 6 feature vectors. The feature vector is labeled 1 if it corresponds to an actual opinion-target pair, and 0 otherwise.

5.2.5 CRF-based Opinion-Target Pairing

In this method, we address the problem as a sequence labeling task. We use Conditional Random Fields (CRF) as our sequence labeling algorithm. The features used for training the CRF model are similar to the features used in the classification approach. We extract the features for every token in the sentence. if the token is the head of the target, we label it 1, and 0 otherwise. If the sentence contains multiple opinion words, we produce a training example per opinion word. Table 5.3 shows the training example that corresponds to the sentence *"I agree with you"*.

Token	POS	Distance	Position	Dep. Path	Dep. Length	Syn. Path	Syn Length	Common Anc.
I	PRP	1	BF	nsubj	1	PRP-NP-S-VP-VBP	5	S
agree	VBP	0	SL	-	0	-	0	VBP
with	IN	1	AF	prep	1	IN-PP-VP-VBP	4	VP
you	PRP	2	AF	prep-pobj	2	PRP-NP-PP-VP-VBP	5	VP
.	.	3	AF	-	-1	.-S-VP-VBP	4	S

Table 5.3: One of the training examples used to train the opinion-target pairing CRF model

5.3 Representation

If a sentence S in a post written by participant P_i contains an opinion word OP_j and a target TR_k, and if the opinion-target pair satisfies one of our dependency rules, we say that P_i expresses an attitude towards TR_k. The polarity of the attitude is determined by the polarity of OP_j. We represent this as $P_i \overset{+}{\rightarrow} TR_k$ if OP_j is positive and $P_i \overset{-}{\rightarrow} TR_k$ if OP_j is negative. If the opinion word falls inside a negation scope as determined by the method described in Chapter IV, the polarity is reversed.

It is likely that the same participant P_i express sentiment toward the same target TR_k multiple times in different sentences in different posts. We keep track of the counts of all the instances of positive/negative attitude that P_i expresses toward TR_k. We represent this as $P_i \xrightarrow[n-]{m+} TR_k$ where m (n) is the number of times P_i expressed positive (negative) attitude toward TR_k.

5.4 Evaluation

5.4.1 Data

For training and testing, we use the J.D. Power and Associates (JDPA) sentiment corpus [100]. JDPA consists of 515 blog posts about automobiles and digital cameras. The data is manually annotated for opinion words and named, nominal, and pronominal mentions of entities. Opinion-target relations are also annotated in the data. The data contains 330,762 tokens which make up 19,322 sentences, 87,532 mentions, 15,637 sentiment expressions, and 22,662 relations between entities. We

	Precision	Recall	F-Measure
Rule-based	0.88	0.44	0.59
CRF	0.75	0.56	0.64
SVM	0.73	0.60	0.66

Table 5.4: Results of opinion-target pairing using the three proposed methods

train our models on data feature examples extracted from this data set.

5.4.2 Results

We compare the three methods: the rule based method, the classification method, and the sequence labeling method. We used Weka [72] and Libsvm [38] to run the classification experiments. We used a linear kernel with c=1. We used CRF++ [2] for the sequence labeling experiments. We set all the parameters of CRF++ to default. Table 5.4 shows the results of 10-fold cross validation on the data. The results show that the rule-based method achieves a higher precision but lower recall than the supervised methods. The performance of the CRF model is comparable to the performance of SVM model. CRF achieves higher precision but lower recall than SVM. Overall, SVM achieves better F-measure.

[2] http://crfpp.googlecode.com/svn/trunk/doc/index.html

CHAPTER VI

Application: Subgroup Detection

Online forums discussing ideological and political topics are common[1]. When people discuss a disputed topic they usually split into subgroups. The members of each subgroup carry the same opinion toward the discission topic. The member of a subgroup is more likely to show positive attitude to the members of the same subgroup, and negative attitude to the members of opposing subgroups.

For example, let us consider the following two snippets from a debate about school uniform.

(1) Discussant 1: I believe that the school uniform is a good idea because school uniform improve student attendance.

(2) Discussant 2: I disagree with you. School uniform is a bad idea because people cannot show their personality.

In (1), the writer is expressing positive attitude regarding school uniform. The writer of (2) is expressing negative attitude (disagreement) towards the writer of (1) and negative attitude regarding the idea of school uniform. It is clear from this short dialog that the writer of (1) and the writer of (2) are members of two opposing subgroups. Discussant 1 is supporting school uniform, while Discussant 2 is against it.

[1] www.politicalforum.com, www.createdebate.com, www.forandagainst.com, etc

In this chapter, we present two unsupervised approaches for determining the subgroup membership of each participant in a discussion. We use the techniques presented in previous chapters to identify the reply structure of of discussion and to identify opinion expressions and their targets.

In the first approach to subgroups detection, we construct a vector of attitude features for each participant in the discussion,. We call this vector the *discussant attitude profile.* The attitude profile of a discussant contains an entry for every other discussant and an entry for every subtopic or entity mentioned in the discission. We use clustering techniques to cluster the attitude vector space. We use the clustering results to determine the subgroup structure of the discussion group and the subgroup membership of each participant.

In the second approach, we use attitude predictions to construct a signed network representation of the discussion thread. In this network nodes represent discussants. Edges represent attitude relations among discussants. An edge carries a positive sign if the relation between the two discussants it connects is positive; otherwise, the sign of the edge is negative. This work is based on the work published in [15].

The rest of this chapter is organized as follows. Section 6.1 reviews previous work. Section 6.2 presents our approach. Experiments, results and analysis are presented in Section 6.4. We conclude in Section 6.9.

6.1 Related Work

Previous work has studied community mining in social media sites. Somasundaran and Wiebe [175] presents an unsupervised opinion analysis method for debate-side classification. They mine the web to learn associations that are indicative of opinion stances in debates and combine this knowledge with discourse information. Anand et al. [21] present a supervised method for stance classification. They use a number of linguistic and structural features such as unigrams, bigrams, cue words, repeated punctuation, and opinion dependencies to build a stance classification model. This work is limited to dual sided debates and defines the problem as a classification task where the two debate sides are know beforehand. Our work is characterized by handling multi-side debates and by regarding the problem as a clustering problem where the number of sides is not known by the algorithm. This work also utilizes only discussant-to-topic attitude predictions for debate-side classification. Out work utilizes both discussant-to-topic and discussant-to-discussant attitude predictions.

In another work, Kim and Hovy [130] predict the results of an election by analyzing discussion threads in online forums that discuss the elections. They use a supervised approach that uses unigrams, bigrams, and trigrams as features. In contrast, our work is unsupervised and uses different types information. Moreover, although this work is related to ours at the goal level, it does not involve any opinion analysis.

Another related work classifies the speakers side in a corpus of congressional floor debates, using the speakers final vote on the bill as a labeling for side [185, 28, 209]. This work infers agreement between speakers based on cases where one speaker mentions another by name, and a simple algorithm for determining the polarity of the sentence in which the mention occurs. This work shows that even with the resulting sparsely connected agreement structure, the MinCut algorithm can improve over stance classification based on textual information alone. This work also requires that the debate sides be known by the algorithm and it only identifies discussant-to-discussant attitude. In our experiments below we show that identifying both discussant-to-discussant and discussant-to-topic attitudes achieves better results.

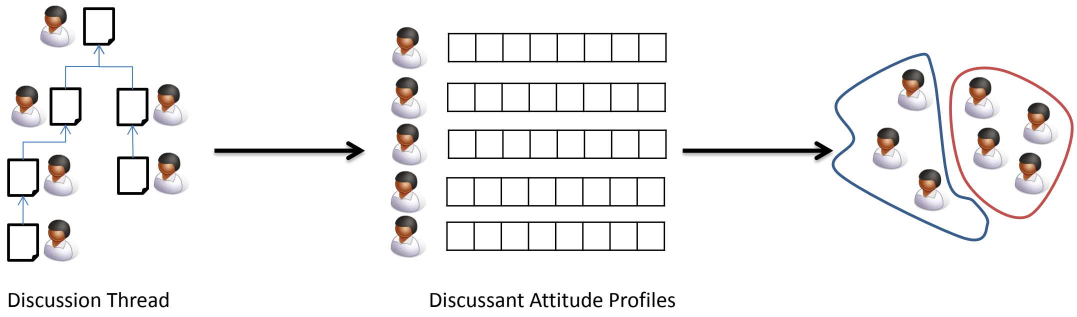


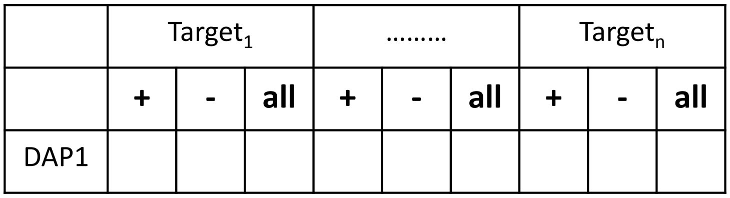
Figure 6.1: A diagram illustrating the subgroup detection approach that constructs vectors representing discussant attitudes and then cluster them

6.2 Approach

In this section, we describe a system that takes a discussion thread as input and outputs the subgroup membership of each discussant. The system starts by applying the methods described in the previous chapters to process the thread. First, it parses the discussion thread to identify posts, discussants, and the reply structure of the thread. Next, the system identifies opinion expressions and their targets. Once this processing is done, the system uses attitude predictions to determine the subgroup membership of each discussant. We propose two approaches for this task and present them in the following subsections.

6.2.1 Approach I: Clustering Attitude Vector Space

In this approach, a vector called Discussant Attitude Profile (DAP) is constructed. This vector represents the attitude the discussant has towards the different targets of opinion in the discussion. Subgroups are identified by clustering the vector space of all DAPs. Figure 6.1 illustrates the pipeline of this approach.



	Target_1			………			Target_n		
	+	-	all	+	-	all	+	-	all
DAP1									

Figure 6.2: Illustration of the discussant attitude profile

Discussant Attitude Profile

We propose a representation of discussantsáttitudes towards the identified targets in the discussion thread. As stated above, a target could be another discussant or an entity mentioned in the discussion. Our representation is a vector containing numerical values. The values correspond to the counts of positive/negative attitudes expressed by the discussant toward each of the targets. We call this vector the *discussant attitude profile (DAP)*. We construct a DAP for every discussant. Given a discussion thread with d discussants and e entity targets, each attitude profile vector has $n = (d + e) * 3$ dimensions (Figure 6.2). In other words, each target (discussant or entity) has three corresponding values in the DAP: 1) the number of times the discussant expressed positive attitude toward the target, 2) the number of times the discussant expressed a negative attitude towards the target, and 3) the number of times the the discussant interacted with or mentioned the target. It has to be noted that these values are not symmetric since the discussions explicitly denote the source and the target of each post.

Clustering

At this point, we have an attitude profile (or vector) constructed for each discussant. Our goal is to use these attitude profiles to determine the subgroup membership of each discussant. We can achieve this goal by noticing that the attitude profiles of discussants who share the same opinion are more likely to be similar to each other than to the attitude profiles of discussants with opposing opinions. This suggests that clustering the attitude vector space will achieve the goal and split the discussants into subgroups according to their opinion.

6.2.2 Approach II: Signed Network Partitioning

In this approach, we construct a signed network representation of the discussion thread. Figure 6.3 illustrates the pipeline of this approach.

Signed Network Construction

Each participant is represented by a node in the network. A signed edge is created between two participants A and B in one of these cases:

1. If A and B exchanged 2 or more posts. The edge is assigned a negative sign if at least one of the posts contained a mention of the post recipient that was a target of a negative opinion expression; otherwise, a positive sign is assigned to the edge.

2. If both A and B posted text that contains a mention of a subtopic or an entity and targeted it with an opinion expression. The sign of the edge is positive if their opinions have the same polarity. The sign of the edge is negative if their opinions have opposite polarities.

Signed Network Partitioning

To detect subgroups in a discussion thread, we partition the corresponding signed network. We experimented with two different partitioning methods. The first one is a greedy optimization algorithm that is based on the principals of the structural balance theory [51]. A criterion function for a local optimization partitioning pro-

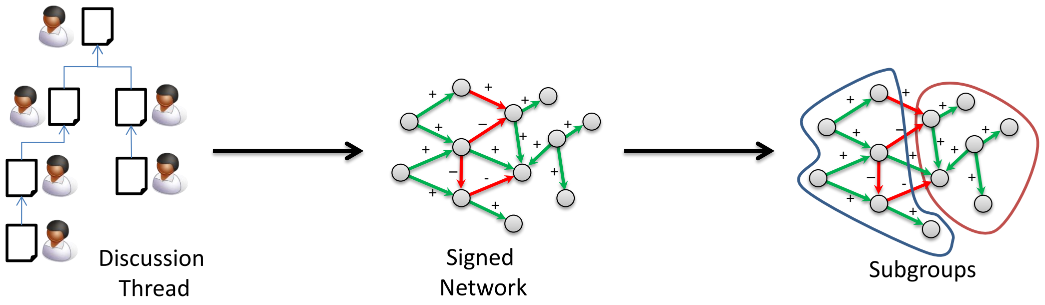


Figure 6.3: A diagram illustrating the subgroup detection approach that constructs a signed network representation of the discussion and then partitions it

cedure is constructed such that positive links are dense within groups and negative links are dense between groups.

For any potential partition C, we seek to optimize the following function:

$$P(C) = \alpha \sum_{n} + (1 - \alpha) \sum_{p} \tag{6.1}$$

where $\sum_n$ is the number of negative links between nodes in the same subgroup, $\sum_p$ is the number of positive links between nodes in different subgroups, and α is a trade factor that represents the importance of the two terms. We set α to 0.5 in all our experiments.

Clusters are selected such that:

$$C^* = \arg\min P(C) \tag{6.2}$$

A greedy optimization framework is used to minimize $P(C)$. Initially, nodes are randomly partitioned into t different clusters and the criterion function P is evaluated for that cluster. Every cluster has a set of neighbors in the cluster space. A neighbor cluster is obtained by moving one node from one cluster to another, or by exchanging two nodes in two different clusters. Neighbor partitions are evaluated, and if one with a lower value for the criterion function is found, it is set as the current partition. This greedy procedure is repeated with random restarts until a minimal solution is found. To determine the number of subgroups t, we select t that minimizes the optimization function P(C). In all experiments we used an upper limit of $t = 5$. This technique was able to identify the correct number of subgroups in 77% of the times. In the rest of the cases, the number was different from the correct number by at most 1 except for a single case where it was 2.

The second partitioning method uses a recursive agent-based algorithm [208]. This algorithm partitions a signed network into subnetworks in two phases. In the first phase, a random agent starts from an arbitrary node s and walks l random steps before it stops. At each step, the agent selects the next node based on the transition probability distribution that depends on the degree of connectivity. The process is repeated many times and at each iteration, the node at which the random walk stopped is recorded. This information is then used to compute the probability of reaching each node from the start node in l random steps. The best value for l was chosen using the development set. In the following experiments, l was set to 5 unless otherwise specified.

Intuitively, the probability of remaining in the same community after a number of transitions, is greater than that of going out to a different community. Based on this observation, the nodes in the network are ordered by the probability of reaching them from the start node. The next step is to pick a cut off point c so that all the nodes with probability higher than c are extracted and used to form the community that contains the start node s. The nodes with lower probabilities form an opposing community. The same process is then applied recursively to the two extracted communities until no more communities can be extracted. This happens when all the nodes have a probability above c. The cut off value c is chosen based on a criterion function that tries to maximize density of intra-community positive links and the maximize the density of the inter-community negative links. The details of how c is computed is described in [208]. FEC does not need to know the number of clusters in advance because it starts from the big group and recursively split it into subgroups until no more split is possible.

6.3 Data

In this section, we describe the data sets used in this paper. We use three different data sets. The first data set (*politicalforum*, henceforth) consists of 5,743 posts collected from a political forum[2]. All the posts are in English. The posts cover 12 disputed political and ideological topics. The discussants of each topic were asked to participate in a poll. The poll asked them to determine their stance on the discussion topic by choosing one item from a list of possible arguments. The list of participants who voted for each argument was published with the poll results. Each poll was accompanied by a discussion thread. The people who participated in the poll were allowed to post text to that thread to justify their choices and to argue with other participants. We collected the votes and the discussion thread of each poll. We used the votes to identify the subgroup membership of each participant.

The second data set (*createdebate*, henceforth) comes from an online debating site [3]. It consists of 30 debates containing a total of 2,712 posts. Each debate is about one topic. The description of each debate states two or more positions regarding the debate topic. When a new participant enters the discussion, she explicitly picks a position and posts text to support it, support a post written by another participant who took the same position, or to dispute a post written by another participant who

[2] http://www.politicalforum.com
[3] http://www.createdebate.com

took an opposing position. We collected the discussion thread and the participant positions for each debate.

The third data set (*wikipedia*, henceforth) comes from the Wikipedia[4] discussion section. When a topic on Wikipedia is disputed, the editors of that topic start a discussion about it. We collected 117 Wikipeida discussion threads. The threads contains a total of 1,867 posts.

The *politicalforum* and *createdebate* data sets are self labeled as described above. To annotate the Wikipedia data, we asked an expert annotator to read each of the Wikipedia discussion threads and determine whether the discussants split into subgroups in which case he was asked to determine the subgroup membership of each discussant.

Table 6.1 lists few example threads from our three data sets. Table 6.2 shows a portion of discussion thread between three participants about enforcing a new immigration law in Arizona. This thread appeared in the polictalforum data set. The text posted by the three participants indicates that A's position is with enforcing the law, that B agrees with A, and that C disagrees with both. This means that A and B belong to the same opinion subgroup, while belongs to an opposing subgroup.

We randomly selected 6 threads from our data sets (2 from *politicalforum*, 2 from *createdebate*, and 2 from *Wikipedia*) and used them as development set. This set was used to develop our approach.

6.4 Evaluation

In this section, we present several levels of evaluation of our system and the two approaches to subgroup detection. First, we compare our system to baseline systems. Second, we study how the choice of the clustering (portioning) algorithm impacts

[4] http://www.wikipedia.com

Source	**Topic**	**Question**	**#Sides**	**#Posts**	**#Participants**
Politicalforum	Arizona Immigration Law	Do you support Arizona in its decision to enact their Immigration Enforcement law?	2	738	59
	Airport Security	Should we pick muslims out of the line and give additional scrutiny/screening?	4	735	69
	Vote for Obama	Will you vote for Obama in the 2012 Presidential elections?	2	2599	197
Createdebate	Evolution	Has evolution been scientifically proved?	2	194	98
	Social networking sites	It is easier to maintain good relationships in social networking sites such as Facebook.	2	70	31
	Abortion	Should abortion be banned	3	477	70
Wikipedia	Ireland	Misleading description of Irland island partition	3	40	10
	South Africa Goverment	Was the current form of South African government born in May 1910?	3	23	5
	Oil Spill	Obama's response to gulf oil spill	3	30	12

Table 6.1: Example threads from our three datasets

Participant A posted:	I support Arizona because they have every right to do so. They are just upholding well-established federal law. All states should enact such a law.
Participant B commented on A's post:	I support the law because the federal government is either afraid or indifferent to the issue. Arizona has the right and the responsibility to protect the people of the State of Arizona. If this requires a possible slight inconvenience to any citizen so be it.
Participant C commented on B's post:	That is such a sad thing to say. You do realize that under the 14th Amendment, the very interaction of a police officer asking you to prove your citizenship is Unconstitutional? As soon as you start trading Constitutional rights for "security", then you've lost.

Table 6.2: Example posts from the Arizona Immigration Law thread

the results. Third, we study the impact of each component in our thread processing pipeline on the performance. All the results reported in this section that show difference in performance are statistically significant at the 0.05 level (as indicated by a 2-tailed paired t-test). Before describing the experiments and presenting the results, we first describe the evaluation metrics we use.

6.4.1 Evaluation Metrics

We use two evaluation metrics to evaluate subgroups detection accuracy: Purity and Entropy. To compute Purity [122], each cluster is assigned the class of the majority vote within the cluster, and then the accuracy of this assignment is measured by dividing the number of correctly assigned members by the total number of instances. It can be formally defined as:

$$purity(\Omega, C) = \frac{1}{N}\sum_{k}\max_{j}|\omega_k \cap c_j| \tag{6.3}$$

where $\Omega = \{\omega_1, \omega_2, ..., \omega_k\}$ is the set of clusters and $C = \{c_1, c_2, ..., c_J\}$ is the set of classes. ω_k is interpreted as the set of documents in ω_k and c_j as the set of documents in c_j. The purity increases as the quality of clustering improves.

The second metric is Entropy. The Entropy of a cluster reflects how the members of the k distinct subgroups are distributed within each resulting cluster; the global quality measure is computed by averaging the entropy of all clusters:

$$Entropy = -\sum^{j}\frac{n_j}{n}\sum^{i}P(i,j) \times log_2 P(i,j) \tag{6.4}$$

where $P(i,j)$ is the probability of finding an element from the category i in the cluster j, n_j is the number of items in cluster j, and n the total number of items

Method	Createdebate		Politicalforum		Wikipedia	
	P	E	P	E	P	E
DAPC - EM	0.63	0.71	0.61	0.82	0.63	0.61
DAPC - FF	0.63	0.70	0.60	0.83	0.64	0.59
DAPC - kmeans	0.64	0.68	0.61	0.80	0.66	0.55

Table 6.3: Comparison of different clustering algorithms

in the distribution. In contrast to purity, the entropy decreases as the quality of clustering improves.

6.4.2 Evaluation of The Attitude Vectors Clustering Approach

Choice of the clustering algorithm

We experimented with three different clustering algorithms: expectation maximization (EM), and k-means [119], and FarthestFirst (FF) [86, 47]. As we did in the previous subsection, we use Eculidean distance to measure the distance between vectors and we set the number of clusters to $\sqrt{d/2}$ where d is the number of discussants. All the system (DAP) components are included as describe in Section 8.3. The purity and entropy values using each algorithm are shown in Table 6.3. Although k-means seems to be performing slightly better than other algorithms, the differences in the results are not significant. This indicates that the choice of the clustering algorithm does not have a noticeable impact on the results. We also experimented with using Manhattan distance and cosine similarity instead of Euclidean distance to measure the distance between attitude vectors. We noticed that the choice of the distance does not have significant impact on the results as well.

Component Evaluation

In this subsection, we evaluate the impact of the different components in the pipeline on the system performance. We do that by removing each component from the pipeline and measuring the change in performance. We perform the following experiments: 1) We run the full system with all its components included (DAPC). 2) We run the system and include only discussant-to-discussant attitude features in the attitude vectors (DAPC-DD). 3) We include only discussant-to-entity attitude features in the attitude vectors (DAPC-DE). 4) We include only sentiment features in the attitude vector; i.e. we exclude the interaction count features (DAPC-SE). 5) We include only interaction count features to the attitude vector; i.e. we exclude sentiment features (DAPC-INT). 6) We skip the anaphora resolution step in the entity identification component (DAPC-NO_AR). 7) We only use named entity recognition to identify entity targets; i.e. we exclude the entities identified through noun phrasing chunking (DAPC-NER). 8) Finally, we only noun phrase chunking to identify entity targets (DAPC-NP). In all these experiments k-means is used for clustering and the number of clusters is set as explained in the previous subsection.

The results show that all the components in the system contribute to better performance of the system. We notice from the results that the performance of the system drops significantly if sentiment features are not included. This is result corroborates our hypothesis that interaction features are not sufficient factors for detecting rift in discussion groups. Including interaction features improve the performance (although not by a big difference) because they help differentiate between the case where participants A and B never interacted with each other and the case where they interact several time but never posted text that indicate difference in opinion between them. We also notice that the performance drops significantly in DAPC-DD and DAPC-DD which also supports our hypotheses that both the sentiment discussants show toward one another and the sentiment they show toward the aspects of the discussed topic are important for the task. Although using both named entity recognition (NER) and noun phrase chunking achieves better results, it can also be noted from

Method	Createdebate		Politicalforum		Wikipedia	
	P	E	P	E	P	E
DAPC	**0.64**	**0.68**	**0.61**	**0.80**	**0.66**	**0.55**
DAPC-DD	0.59	0.77	0.57	0.86	0.62	0.61
DAPC-DE	0.60	0.69	0.58	0.84	0.58	0.78
DAPC-SE	0.62	0.70	0.60	0.83	0.61	0.62
DAPC-INT	0.54	0.88	0.52	0.91	0.57	0.85
DAPC-NO_AR	0.62	0.72	0.60	0.84	0.64	0.60
DAPC-NER	0.61	0.71	0.58	0.86	0.63	0.59
DAPC-NP	0.63	0.75	0.59	0.84	0.65	0.62

Table 6.4: Impact of system components on the performance

	Precision	Recall	F1
positive	66.1%	96.2%	78.4%
negative	70.0%	60.1%	64.7%

Table 6.5: Performance of the signed network extraction component

the results that NER contributes more to the system performance. Finally, the results support Jakob and Gurevych [94] findings that anaphora resolution aids opinion mining systems.

6.4.3 Evaluation of The Signed Network Partitioning Approach

Signed Network Construction Evaluation

We used the *createdebate* dataset to evaluate extracting signed networks from discussions. Table 6.5 shows the average precision, recall, and F1 for predicting the edge signs in the networks extracted from the *createdebate* dataset.

Choice of Partitioning Algorithm

We compare the performance of the two signed network partitioning algorithms described above. We refer to the greedy optimization algorithm of [51] as *DM.* We refer to the random walk based algorithm of [208] as *FEC.* Table 6.6 shows the average purity P and entropy E values of the method based on signed networks and the baselines using different partitioning algorithms. The results show that

Method	Createdebate		Politicalforum		Wikipedia	
	P	E	P	E	P	E
Signed Network - DM	**0.61**	**0.74**	**0.58**	**0.80**	**0.65**	**0.54**
Signed Network - FEC	0.59	0.78	0.57	0.87	0.63	0.71

Table 6.6: Comparison of two partitioning algorithms

the greedy partitioning algorithm DM performs slightly better than the random walk based algorithm FEC. This could be explained by our observation that FEC performs better on large networks (> 150 nodes). Most of the discussions have less than 150 participants.

6.4.4 Comparison to Baseline Systems

We evaluate two baseline methods. The first baseline (GC) uses graph clustering to partition a network based on the interaction frequency between participants. We build a graph where each node represents a participant. Edges link participants if they exchange posts, and edge weights are based on the number of interactions. We tried two methods for clustering the resulting graph: spectral partitioning [118] and a hierarchical agglomeration algorithm which works by greedily optimizing the modularity for graphs [42].

The second baseline (TC) is based on the premise that the member of the same subgroup are more likely to use vocabulary drawn from the same language model. We collect all the text posted by each participant and create a tf-idf representations of the text in a high dimensional vector space. We then cluster the vector space to identify subgroups. We use k-means [119] as our clustering algorithm in this experiment (comparison of various clustering algorithms is presented later in the chapter). The distances between vectors are Euclidean distances. K-means needs the number of clusters as input. We follow the rule of thumb [123] of setting this

Method	Createdebate		Politicalforum		Wikipedia	
	P	E	P	E	P	E
GC - Spectral	0.50	0.85	0.50	0.88	0.49	0.89
GC - Hierarchical	0.48	0.86	0.47	0.89	0.49	0.87
TC - kmeans	0.51	0.84	0.49	0.88	0.52	0.85

Table 6.7: Comparison to baseline systems

number to $\sqrt{d/2}$ where d is the number of discussants. Table 6.7 shows the purity and entropy results achieved by the baseline systems.

The results show that our system (both approaches) performs significantly better the baselines on the three datasets in terms of both the purity (P) and the entropy (E) (notice that lower entropy values indicate better clustering). The values reported are the average results of the threads of each dataset. We believe that the baselines performed poorly because the interaction frequency and the text similarity are not key factors in identifying subgroup structures. Many people would respond to people they disagree with more, while others would mainly respond to people they agree with most of the time. Also, people in opposing subgroups tend to use very similar text when discussing the same topic and hence text clustering does not work as well.

6.5 Detecting Subgroups in Arabic Discussion

Unfortunately, not much work has been done on Arabic sentiment analysis and opinion mining. Abbasi et al. [2] applies sentiment analysis techniques to identify and classify document-level opinions in text crawled from English and Arabic web forums. Hassan et al. [73] proposed a method for identifying the polarity of non-English words using multilingual semantic graphs. They applied their method to Arabic and Hindi. Abdul-Mageed and Diab [3] annotated a corpus of Modern Standard Arabic (MSA) news text for subjectivity at the sentence level. In a later work [4], they expanded their corpus by labeling data from more genres using Amazon Mechanical Turk. Abdul-Mageed et al. [6] developed SAMAR, a system for subjectivity and Sentiment Analysis for Arabic social media genres. We use this system as a component in our proposed system.

In this section, we address the problem of identifying opinion subgroups in Arabic discussions. We propose a pipeline that consists of five components. The input to the pipeline is a discussion thread in Arabic crawled from a discussion forum. The output is the list of participants in the discussion and the subgroup membership of each discussant. We describe the components of our pipeline in the following subsections.

6.5.1 Preprocessing

The input to this component is a discussion thread in HTML format. We parse the HTML file to identify the posts, the discussants, and the thread structure. We transform the Arabic content of the posts and the discussant names that are written in Arabic to the Buckwalter encoding [34]. We use AMIRAN [49], a system for processing Arabic text, to tokenize the text and identify noun phrases.

6.5.2 Identifying Opinionated Text

To identify opinion-bearing text, we start from the word level. We identify polarized words that appear in text using a lexicon of Arabic polarized words. In our experiments, we use Sifat [5], a lexicon of 3982 Arabic adjectives labeled as positive, negative, or neutral.

The polarity of a word may be dependant on its context [205]. For example, a positive word that appears in a negated context should be treated as expressing negative opinion. To identify the polarity of a word given the sentence it appears in, we use SAMAR [6], a system for subjectivity and sentiment analysis for Arabic social media genres. SAMAR labels a sentence that contains an opinion expression as positive, negative, or neutral taking into account the context of the opinion expression. The reported accuracy for SAMAR on different data sets ranges between 84% and 95% for subjectivity classification and 65% and 81% for polarity classification.

6.5.3 Identifying Opinion Targets

In this step, we determine the targets that the opinion is expressed towards. We treat as an opinion target any noun phrase (NP) that appears in a sentence that SAMAR labeled as subjective (positive or negative). To avoid the noise that may result from including all noun phrases, we limit what we consider as an opinion target to the ones that appear in at least two posts written by two different participants. Since, the sentence may contain multiple possible targets for every opinion expression, we associate each opinion expression with the target that is closest to it in the sentence. For each discussant, we keep track of the targets mentioned in his/her posts and the number of times each target was mentioned in a positive/negative context.

6.5.4 Textual Similarity

If two participants share the same opinion, they tend to focus on similar aspects of the discussion topic and emphasize similar points that support their opinion. To capture this feature, we follow previous work [69, 48] and apply Latent Dirichelet Allocation (LDA) topic models to the text written by the different participants. We use an LDA model with 100 topics. Hence, we represent all the text written in the discussion by a participant as a vector of 100 dimensions.

6.5.5 Subgroup Detection

At this point, we have for every discussant the targets towards which he/she expressed explicit opinion and a 100-dimensions vector representing the LDA distribution of the text written by him/her. We use this information to represent the discussion in two representations. In the first representation, each discussant is represented by a vector. For every target identified in step 3 of the pipeline, we add three entries in the vector. The first entry holds the total number of times the target was mentioned by the discussant. The second entry holds the number of times the target was mentioned in a positive context. The third entry holds the number of target mentions in a negative context. We also add to this vector the 100 topic entries from the LDA vector of that discussant. So, if the number of targets identified in step 3 of the pipeline is t then the number of entries in the discussant vector is $3 * t + 100$.

To identify opinion subgroups, we cluster the vector space. We experiment with several clustering algorithms including K-means [119], FarthestFirst (FF) [86, 47], and Expectation Maximization (EM).

The second representation is a signed network representation. In this representation, each discussant is represented by a node in a graph. Two discussants are connected with an edge if they both mention at least one common target in their posts. If a discussant mentions a target multiple times in different contexts with different polarities, the majority polarity is assumed as the opinion of this discussant with respect to this target. A positive sign is assigned to the edge connecting two discussants if the number of targets that they have similar opinion towards is greater than the targets that they have opposing opinion towards, otherwise a negative sign is assigned to the edge. To identify subgroups, we use the DM algorithm to partition the network.

6.5.6 Data

We use data from an Arabic discussion forum, `www.Naqeshny.com`. Naqeshny is a platform for two-sided debates in various topics (ideological and political). The debate starts when a person asks a question (e.g. which political party do you support?) and gives two possible answers or positions. The members of the site who are interested in the topic participate in the debate by selecting a position and then posting text to support that position and dispute the opposing position. This means that the data set is self labeled for subgroup membership. Since the tools used in our system are trained on Modern Standard Arabic (MSA) text, we selected debates that are mostly MSA. The data set consists of 36 debates comprising a total of 711 posts written by 326 users. The average number of posts per discussion is 19.75 and the average number of participants per discussion is 13.08.

6.5.7 Experiments and Results

We use three metrics to evaluate the resulting subgroups: Purity [122], Entropy, and F-measure. We ran different variations of the system on the data set described in the previous section. In one variation, we use the signed network partitioning approach to detect subgroups. In the other variations, we use the vector space clustering approach. We experiment with different clustering algorithms. We also run two experiments to evaluate the contribution of both opinion-target counts and latent similarity features on the clustering accuracy. In one run, we use target-opinion counts only and in the other run we use latent similarity only. Table 6.8 shows the results. The results show that the clustering approach achieves better results than the signed network partitioning approach. This can be explained by the fact that the vector representation is a richer representation and encodes all opinion information

System	Purity	F-Measure	Entropy
Signed Network	0.71	0.67	0.68
Clustering - K-means	0.72	0.70	0.67
Clustering - EM	**0.77**	**0.76**	**0.50**
Clustering - FF	0.72	0.69	0.70
Opinion-Target Only	0.67	0.65	0.72
Text Similarity Only	0.64	0.65	0.74

Table 6.8: Comparison of the different variations of the proposed approach

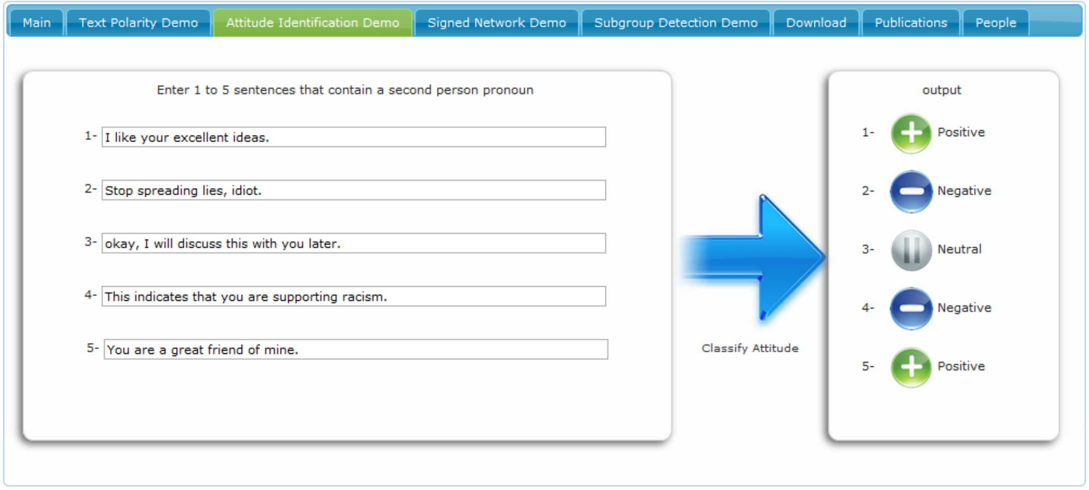
including counts which are not encoded explicitly in the signed network representation. The results also show that Expectation Maximization achieves significantly better results than the other clustering algorithms that we experimented with. It also shows that both text similarity and opinion-target features are important and contribute to the performance.

6.6 Systems

In this section, we describe two systems that we developed as an implementation of the methods proposed in this chapter. *AttitudeMiner* implements the signed network approach for detecting subgroups, while *SubgroupDetector* implements the clustering approach. In the following subsections, we give a description of the two systems.

6.7 AttitudeMiner

AttitudeMiner is an implementation of the signed network approach for subgroup detection. The system is implemented in Perl. Some of the components in the processing pipeline use external tools that are implemented in either Perl, Java, or Python. All the external tools come bundled with the system. The system is compatible with all the major platforms including windows, Mac OS, and all Linux distributions. The installation process is very straightforward. There is a single installation script that will install the system, install all the dependencies, and do all the required configurations. The installation requires that Java JRE, Perl, and



Enter 1 to 5 sentences that contain a second person pronoun

1- I like your excellent ideas.

2- Stop spreading lies, idiot.

3- okay, I will discuss this with you later.

4- This indicates that you are supporting racism.

5- You are a great friend of mine.

output

1- Positive

2- Negative

3- Neutral

4- Negative

5- Positive

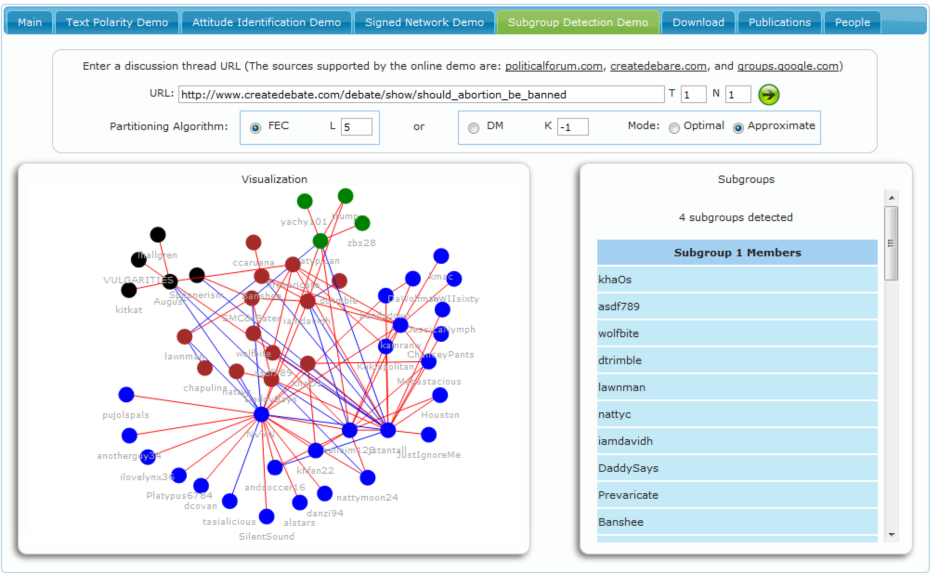
Figure 6.4: The web interface for identifying attitudinal sentences and their polarity

Python be installed on the machine.

The system has a command-line interface that provides full access to the system functionality. The command-line interface can be used to run the whole pipeline or any portion of it. It can also be used to access any component directly. Each component has a corresponding script that can be run separately. The input and output specifications of each component are described in the accompanying documentation. All the parameters that control the performance of the system can also be passed through the command-line interface.

The system can process any discussion thread that is input to it in a specific XML format. The final output of the system is also in XML format. The XML schema of the input/output is described in the documentation. It is the user responsibility to write a parser that converts an online discussion thread to the expected XML format. The system package comes with three such parsers for three different discussion sites: www.politicalforum.com, groups.google.com, and www.createdebate.com.

The distribution also comes with three datasets (from three different sources) comprising a total of 300 discussion threads. The datasets are annotated with the



Enter a discussion thread URL (The sources supported by the online demo are: politicalforum.com, createdebare.com, and groups.google.com)

URL: http://www.createdebate.com/debate/show/should_abortion_be_banned T 1 N 1

Partitioning Algorithm: ● FEC L 5 or ○ DM K -1 Mode: ○ Optimal ● Approximate

Subgroups

4 subgroups detected

Subgroup 1 Members
khaOs
asdf789
wolfbite
dtrimble
lawnman
nattyc
iamdavidh
DaddySays
Prevaricate
Banshee

Figure 6.5: The web interface for detecting subgroups in discussions

subgroup labels of discussants. Included in the distribution as well, a script for generating a visualization of the extracted signed network and the identified subgroups.

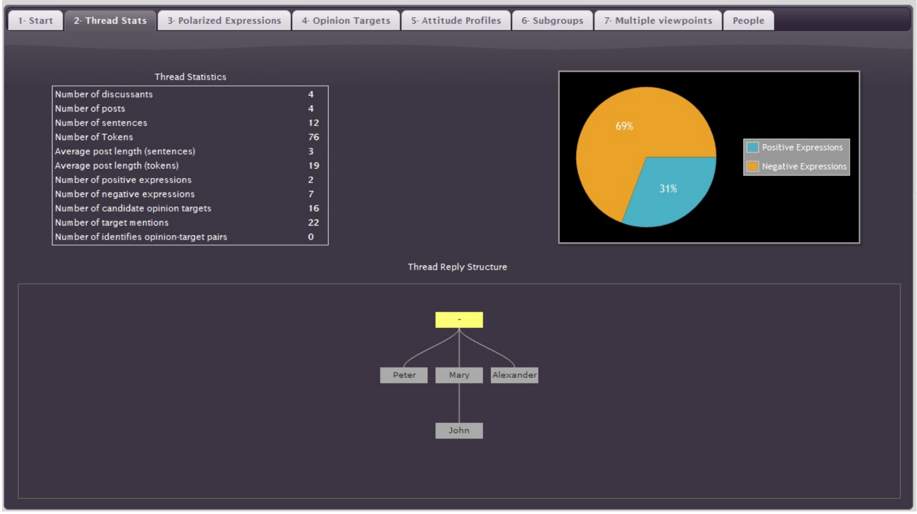
AttitudeMiner also has a web interface that demonstrates most of its functionality. The web interface is intended for demonstration purposes only. No webservice is provided. Figure 6.5 and Figrue 6.4 show two screenshots for the web interface.

6.8 SubgroupDetector

SubgroupDetector is an implementation of the clustering approach for subgroup detection. The system is fully implemented in Java. Part-of-speech tagging, noun group identification, named entity recognition, co-reference resolution, and dependency parsing are all computed using the Stanford Core NLP API.[5] The clustering component uses the JavaML library[6] which provides implementations to several clustering algorithms such as k-means, EM, FarthestFirst, and OPTICS.

The system requires no installation. It, however, requires that the Java Runtime

[5] http://nlp.stanford.edu/software/corenlp.shtml
[6] http://java-ml.sourceforge.net/



Thread Statistics

Number of discussants	4
Number of posts	4
Number of sentences	12
Number of Tokens	76
Average post length (sentences)	3
Average post length (tokens)	19
Number of positive expressions	2
Number of negative expressions	7
Number of candidate opinion targets	16
Number of target mentions	22
Number of identifies opinion-target pairs	0

Figure 6.6: A screenshot of the online demo

Environment (JRE) be installed. All the dependencies of the system come bundled with the system in the same package. The system works on all the standard platforms.

The system has a command-line interface that provides full access to the system functionality. It can be used to run the whole pipeline to detect subgroups or any portion of the pipeline. For example, it can be used to tag an input text with polarity or to identify candidate targets of opinion in a given input. The system behavior can be controlled by passing arguments through the command line interface. For example, the user can specify which clustering algorithm should be used.

To facilitate using the system for research purposes, the system comes with a clustering evaluation component that uses the ClusterEvaluator package.[7]. If the input to the system contains subgroup labels, it can be run in the evaluation mode in which case the system will output the scores of several different clustering evaluation metrics such as purity, entropy, f-measure, Jaccard, and RandIndex. The system also has a Java API that can be used by researchers to develop other systems using

[7] http://eniac.cs.qc.cuny.edu/andrew/v-measure/javadoc/index.html

our code.

The system can process any discussion thread that is input to it in a specific format. The format of the input and output is described in the accompanying documentation. It is the user responsibility to write a parser that converts an online discussion thread to the expected format. However, the system package comes with two such parsers for two different discussion sites: www.politicalforum.com and www.createdebate.com.

The distribution also comes with three datasets (from three different sources) comprising a total of 300 discussion threads. The datasets are annotated with the subgroup labels of discussants.

Finally, we created a web interface to demonstrate the system functionality. The web interface is intended for demonstration purposes only. No webservice is provided. Figure 6.6 shows a screenshots of the web interface. The online demo can be accessed at http://clair.eecs.umich.edu/SubgroupDetector/

6.9 Conclusions

In this chapter, we presented an approach for subgroup detection in ideological discussions. Our system uses Natural Language Processing techniques to identify the attitude the participants of online discussions carry toward each other and toward the aspects of the discussion topic. Attitude prediction as well as interaction frequency to construct an attitude vector for each participant. The attitude vectors of discussants are then clustered to form subgroups. Our experiments showed that our system outperforms text clustering and interaction graph clustering. We also studied the contribution of each component in our system to the overall performance.

Part II. Mining Multiple Perspectives from Scientific Literature

CHAPTER VII

Introduction

Scientific research is a cumulative activity. The work of downstream researchers depends on access to upstream discoveries. The footnotes, end notes, or reference lists within research articles make this accumulation possible. Each time a reference appears in a scientific paper, it is accompanied by a span of text that describes the work being cited. We use the term *citation text* to refer to this text. This text usually summarizes the contribution of the cited paper from the perspective of the citer. It may also be used to declare the relation between the citing work and the cited work. For example, the citing paper may be using an algorithm, a tool, or a corpus described in the cited paper.

In this part of the thesis, we study multiple viewpoints in scientific literature through the lens of citations. Citation text can be seen as a summary of how the scholar sees the cited work from his point of view. Citation text usually highlights the most important aspects of the cited paper such as the research problem the paper addresses, the method it proposes, the good results it reports, or even its drawbacks and limitations.

Bibliometrics (or Scientometrics) is the science that develops quantitative methods and metrics for evaluating the impact of a research field, the impact of a group of researchers, or the impact of a paper. Citation analysis and content analysis are the two main techniques used in bibliometrics. In this work, we study how analyzing *citation text* can be used to develop more accurate bibliometric measures that evaluate the impact of research both quantitatively and qualitatively.

7.1 Related Work

Studying citation patterns and referencing practices has interested researchers for many years [87, 61]. White [196] provides a good survey of the different research directions that study or use citations. Several research efforts have focused on studying the different purposes for citing a paper [62, 194, 139, 1, 32]. Bonzi [32] studied the characteristics of citing and cited works that may aid in determining the relatedness between them. Garfield [62] enumerated several reasons why authors cite other publications, including “alerting researchers to forthcoming work”, paying homage to the leading scholars in the area, and citations which provide pointers to background readings. Weinstock [194] adopted the same scheme that Garfield proposed in her study of citations.

The text surrounding citations has been studied and used in previous work. Nanba and Okumura [143] used the term *citing area* to refer to citation text. They define the citing area as the succession of sentences that appear around the location of a given reference in a scientific paper and have connection to it. They proposed a rule-based algorithm to identify the *citing area* of a given reference. Several other methods have been proposed for identifying citation text [143, 153, 14, 25].

Citation text has been used in several applications. Nakov et al. [141] use the term *citances* to refer to sentences that contain explicit reference to other papers. They explored several different uses of *citances* including the creation of training and testing data for semantic analysis, synonym set creation, database curation, summarization, and information retrieval.

Other examples of applications in which citation text has been used include: scientific paper summarization [53, 152, 128, 155, 153, 10], automatic survey generation [142, 131], citation function classification [142, 184, 173, 183], and paraphrase recognition [170].

7.2 Outline of Part II

In this part of the thesis, we study citation text and present some useful applications in which it can be used. We first address the problem of identifying citation text. This involves identifying text fragments that contain explicit references to other papers and the context and the scope of each reference. We use the term *citation context* to refer to the sentences that appear around an explicit reference and talk about it. We use the term *reference scope* to refer to the fragments of sentences that talk about a reference in citing sentences that cite multiple references. We analyze citation text to identify the author intention behind citing a reference and whether the citation is polarized (i.e. carries a non-neutral sentiment towards the cited work). We use this analysis of citation purpose and polarity to predict the future prominence of a paper. We also show how this analysis can be used to build more accurate and more informative bibliometric measures. We also present a method for producing citation-based summaries of scientific articles from the viewpoints of the other scholars, those who read the paper and cited it. Although several methods have been advised for this problem, our method is uniquely characterized by focusing on the coherence and the readability of summaries generated from citation text.

Chapter VIII describes our approach for identifying citation text. This includes identifying reference anchors, sentences that contain references (*citing sentences*), the adjacent sentences that talk about the cited work (the *citation context*), and the scope of each reference. Chapter IX presents supervised approaches for identifying the purpose (author intention) and the polarity (author sentiment) of citation. Chapter X presents an application that utilizes citing sentences to produce citation-based summaries of scientific articles. Chapter XI presents a number of other applications in which citation text may be useful. We present the initial work we have done on those applications and give directions to future work.

CHAPTER VIII

Identifying Citation Scope

Citation plays an important role in science. It makes the accumulation of knowledge possible. When a reference appears in a scientific article, it is usually accompanied by a span of text that highlights the important contributions of the cited article. We call a sentence that contains an explicit reference to previous work a *citing sentence*. For example, sentence (1) below is a citation sentence that cites a paper by Philip Resnik and describes the problem Resnik addressed in his paper.

(1) ***Resnik (1999)*** *addressed the issue of language identification for finding Web pages in the languages of interest.*

Previous work has studied and used citation sentences in various applications such as: scientific paper summarization [53, 152, 128, 155, 153, 10], automatic survey generation [142, 131], citation function classification [142, 184, 173, 183], and paraphrase recognition [170].

Sentence (1) above contains one reference, and the whole sentence is talking about that reference. This is not always the case in scientific writing. Sentences that contain references to multiple papers are very common. For example, sentence (2) below contains three references.

(2) Grefenstette and Nioche (2000) and Jones and Ghani (2000) use the web to generate corpora for languages where electronic resources are scarce, while ***Resnik (1999)*** *describes a method for mining the web for bilingual texts.*

The first fragment describes the contribution of Grefenstette and Nioche (2000) and Jones and Ghani (2000). The second fragment describes the contribution of Resnik (1999).

This observation should be taken into consideration when using citation sentences in the aforementioned applications. For example, in citation-based summarization of scientific papers, a subset of citation sentences that cite a given target paper is selected and used to form a summary of that paper. It is very likely that one or more of the selected sentences cite multiple papers besides the target. This means that some of the text included in the summary might be irrelevant to the summarized paper. Including irrelevant text in the summary introduces several problems. First, the summarization task aims at summarizing the contributions of the target paper using minimal text. Extraneous text takes space in the summary while being irrelevant and less important. Second, including irrelevant text in the summary breaks the context and confuses the reader. Therefore, if sentence (2) above is to be added to a citation-based summary of Resnikś (1999) paper, only the underlined fragment should be added to the summary and the rest of the sentence should be excluded.

For another example, consider the task of citation function classification. The goal of this task is to determine the reason for citing paper B by paper A based on linguistic and structural features extracted from citation sentences that appear in A and cite B. If a citation sentence in A cites multiple papers besides B, classification features should be extracted only from the fragments of the sentence that are relevant to B. Sentence (3) below shows an examples of this case.

(3) Cohn and Lapata (2008) used the GHKM extraction method (Galley et al., 2004), which is limited to constituent phrases and thus produces a reasonably small set of syntactic rules.

If the target reference is Cohn and Lapata (2008), only the underlined segment should be used for feature extraction. The limitation stated in the second segment of sentence is referring to Galley et al., (2004).

In this paper, we address the problem of identifying the fragments of a citation sentence that are related to a given target reference. Henceforth, we use the term *Reference Scope* to refer to those fragments. We present and compare three different approaches to this problem.

In the first approach, we define the problem as a word classification task. We classify each word in the sentence as *inside* or *outside* the scope of the target reference.

In the second approach, we define the problem as a sequence labeling problem. This is different from the first approach in that the label assigned to each word is dependent on the labels of nearby words. In the third approach, instead of classifying individual words, we split the sentence into segments and classify each segment as *inside* or *outside* the scope of the target reference.

Applying any of the three approaches is preceded by a preprocessing stage. In this stage, citation sentences are analyzed to tag references, identify groups of references, and distinguish between syntactic and non-syntactic references. This work is based on the work published in [13].

The rest of this chapter is organized as follows. Section 8.1 examines the related work. We define the problem in Section8.2. Section 8.3 presents our approaches. Experiments, results and analysis are presented in Section 8.4. We conclude and provide directions to future work in Section 8.5.

8.1 Related Work

Our work is related to a large body of research on citations [87, 61]. The interest in studying citations stems from the fact that bibliometric measures are commonly used to estimate the impact of a researcher's work [33, 117]. White [196] provides a good recent survey of the different research lines that use citations. In this section we review the research lines that are relevant to our work and show how our work is different.

One line of research that is related to our work has to do with identifying what Nanba and Okumura [143] call the *citing area* They define the citing area as the succession of sentences that appear around the location of a given reference in a scientific paper and has connection to it. Their algorithm starts by adding the sentence that contains the target reference as the first member sentence in the citing area. Then, they use a set of cue words and hand-crafted rules to determine whether the surrounding sentences should be added to the citing area or not. In [142] they use their citing area identification algorithm to improve citation type classification and automatic survey generation.

Qazvinian and Radev [153] addressed a similar problem. They proposed a method based on probabilistic inference to extract non-explicit citing sentences; i.e., sentences that appear around the sentence that contains the target reference and are related to it. They showed experimentally that citation-based survey generation produces better results when using both explicit and non-explicit citing sentences rather than using the explicit ones alone.

Although this work shares the same general goal with ours (i.e identifying the pieces of text that are relevant to a given target reference), our work is different in two ways. First, previous work mostly ignored the fact that the citation sentence itself might be citing multiple references. Second, it defined the *citing area* [143] or the *citation context* [153] as a set of whole contiguous sentences. In our work, we address the case where one citation sentence cites multiple papers, and define what we call the *reference scope* to be the fragments (not necessarily contiguous) of the citation sentence that are related to the target reference.

In a recent work on citation-based summarization by Abu-Jbara and Radev [10], the authors noticed the issue of having multiple references in one sentence. They raised this issue when they discussed the factors that impede the coherence and the readability of citation-based summaries. They suggested that removing the fragments of a citation sentence that are not relevant to the summarized paper will significantly improve the quality of the produced summaries. In their work, they defined the scope of a given reference as the shortest fragment of the citation sentence that contains the reference and could form a grammatical sentence if the rest of the sentence was removed. They identify the scope by generating the syntactic parse tree of the sentence and then finding the text that corresponds to the smallest subtree rooted at an S node and contains the target reference node as one of its leaf nodes. They admitted that their method was very basic and works only when the scope forms one grammatical fragment, which is not true in many cases.

Athar [23] noticed the same issue with citation sentences that cite multiple references, but this time in the context of sentiment analysis in citations. He showed experimentally that identifying what he termed the *scope of citation influence* improves sentiment classification accuracy. He adapted the same basic method proposed by Abu-Jbara and Radev [10]. We use this method as a baseline in our evaluation below.

In addition to this related work, there is a large body of research that used citation sentences in different applications. For example, citation sentences has been used to summarize the contributions of a scientific paper [152, 155, 153, 10]. They have been also used to generate surveys of scientific paradigms [143, 131]. Several other papers analyzed citation sentences to recognize the citation function; i.e., the author's reason for citing a given paper [142, 184, 183]. Schwartz et al. [170] proposed a method for aligning the words within citation sentences that cite the same paper. The goal of his work was to aid named entity recognition and paraphrase identification in scientific papers.

We believe that all the these applications will benefit from the output of our work.

8.2 Problem Definition

The problem that we are trying to solve is to identify which fragments of a given citation sentence that cites multiple references are semantically related to a given target reference. As stated above, we call these fragments the *reference scope*. Formally, given a citation sentence $S = \{w1, w2, ..., w_n\}$ where $w1, w2, ..., w_n$ are the words of the sentence (when tokenized using a standard English tokenizer); and given that S contains a set of two or more references R, we want to assign the label 1 to the word w_i if it falls in the scope of a given target reference $r \in R$, and 0 otherwise.

For example, sentences (4) and (5) below are labeled for the target references Tetreault and Chodorow (2008), and Cutting et al.(1992) respectively. The underlined words are labeled 1 (i.e., inside the target reference scope), while all others are labeled 0.

(4) For example, ***Tetreault and Chodorow (2008)*** *use a maximum entropy classifier to build a model of correct preposition usage, with 7 million instances in their training set, and Lee and Knutsson (2008) use memory-based learning, with 10 million sentences in their training set.*

*(5) There are many POS taggers developed using different techniques for many major languages such as transformation-based error-driven learning (Brill, 1995), decision trees (Black et al., 1992), Markov model (****Cutting et al., 1992****), maximum entropy methods (Ratnaparkhi, 1996) etc for English.*

8.3 Approach

In this section, we present our approach for addressing the problem defined in the previous section. Our approach involves two stages: 1) preprocessing and 2) reference scope identification. We present three alternative methods for the second stage. The following two subsections describe the two stages.

8.3.1 Stage 1: Preprocessing

The goal of the preprocessing stage is to clean and prepare the citation sentence for the next processing steps. The second stage involves higher level tasks such as part-of-speech tagging, syntactic parsing, and dependency parsing. The available tools for these tasks are not trained on citation sentences which contain references written in a special format. For example, it is very common in scientific writing to have references (usually written between parentheses) that are not a syntactic part of the sentence. It is also common to cite a group of references who share the same contribution by listing them between parentheses separated by a comma or a semi-colon. We address these issues to improve the accuracy of the processing done in the second stage. The preprocessing stage involves three tasks:

Reference Tagging

The first preprocessing task is to find and tag all the references that appear in the citation sentence. Authors of scientific articles use standard patterns to include references in text. We apply a regular expression to find all the references that appear in a sentence. We replace each reference with a placeholder. The target reference is replaced by TREF. Each other reference is replaced by REF. We keep track of the original text of each replaced reference. Sentence (6) below shows an example of a citation sentence with the references replaced.

(6) These constraints can be lexicalized (REF.1; REF.2), unlexicalized (REF.3; ***TREF.4****) or automatically learned (REF.5; REF.6).*

Reference Grouping

It is common in scientific writing to attribute one contribution to a group of references. Sentence (6) above contains three groups of references. Each group constitutes one entity. Therefore, we replace each group with a placeholder. We use GTREF to replace a group of references that contains the target reference, and GREF to replace a group of references that does *not* contain the target reference. Sentence (7) below is the same as sentence (6) but with the three groups of references replaced.

*(7) These constraints can be lexicalized (GREF.1), unlexicalized (****GTREF.2****) or automatically learned (GREF.3).*

Non-syntactic Reference Removal

A reference (REF or TREF) or a group of references (GREF or GTREF) could either be a syntactic constituent and has a semantic role in the sentence (e.g. GTREF.1 in sentence (8) below) or not (e.g. REF.2 in sentence (8)).

(8) (GTREF.1) apply fuzzy techniques for integrating source syntax into hierarchical phrase-based systems (REF.2).

The task in this step is to determine whether a reference is a syntactic component in the sentence or not. If yes, we keep it as is. If not, we remove it from the sentence and keep track of its position. Accordingly, after this step, REF.2 in sentence (8) will be removed. We use a rule-based algorithm to determine whether a reference should be removed from the sentence or kept. Our algorithm (Algorithm 2) uses stylistic and linguistic features such as the style of the reference, the position of the reference, and the surrounding words to make the decision.

When a reference is removed, we pick a word from the sentence to represent it. This is needed for feature extraction in the next stage. We use as a representative the head of the closest noun phrase (NP) that comes before the position of the removed reference. For example, in sentence (8) above, the closest NP before REF.2 is *hierarchical phrase-based systems* and the head is the noun *systems*.

8.3.2 Stage 2: Reference Scope Identification

In this section we present four different methods for identifying the scope of a given reference within a citation sentence. We compare the performance of four methods in Section 8.4. The following three subsections describe the methods.

Algorithm 2 Remove Non-syntactic References

```
Require: A citation sentence S
for all Reference R (REF, TREF, GREF, or GTREF) in S do
    if R style matches "Authors (year)" then
        Keep R // syntactic
    else if R is the first word in the sentence or in a clause then
        Keep R // syntactic
    else if R is preceded by a preposition (in, of, by, etc.) then
        Keep R // syntactic
    else
        Remove R // non-syntactic
    end if
end for
```

Unsupervised Approach

Many of the citation sentences contain references to multiple papers besides the target. In other words, some fragments of such sentences describe work other than the work of the target paper. These fragments are usually irrelevant, or at least mush less important. Including these fragments in the summary causes several problems. First, the aim of the summarization task is to summarize the contribution of the target paper using minimal text. These fragments take space in the summary, while being irrelevant and less important. Second, including these fragments in the summary breaks the context and hence degrades the fluency and confuses the reader. Third, the performance of later processing steps in our approach is negatively affected by these fragments.

Therefore, it is important to identify the scope of the target reference; i.e. the fragment of the citation sentence that corresponds to the target paper. We define the scope of a reference as the shortest fragment of the citation sentence that contains the reference and could form a grammatical sentence if the rest of the whole sentence was removed.

To find such a fragment, we use a simple but adequate heuristic. We start by parsing the sentence using the link grammar parser [174]. Since the parser is not trained on citation sentences, we replace the references with placeholders before passing the sentence to the parser. Figure 8.1 shows the a portion of the parse tree for Sentence (2) above.

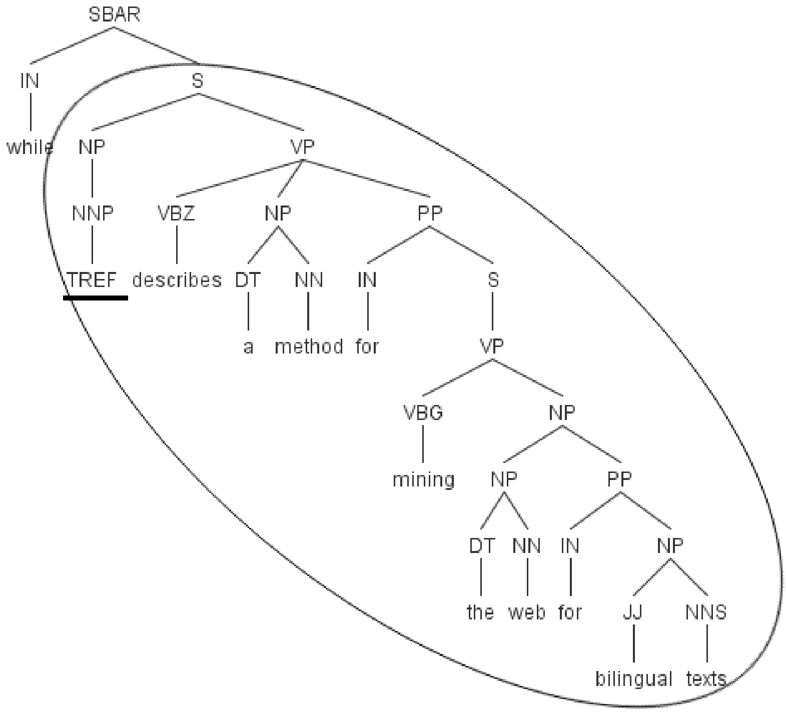


Figure 8.1: An example showing how to extract the scope of the target reference

We extract the scope of the reference from the parse tree as follows. We find the smallest subtree rooted at an S node (sentence clause node) and contains the target reference node. we extract the text that corresponds to this subtree if it is grammatical. Otherwise, we find the second smallest subtree rooted at an S node and so on. For example, the parse tree shown in the figure 8.1 suggests that the scope of the reference is:

Resnik (1999) *describes a method for mining the web for bilingual texts.*

Feature	**Description**
Distance	The distance (in words) between the word and the target reference.
Position	This feature takes the value 1 if the word comes before the target reference, and 0 otherwise.
Segment	After splitting the sentence into segments by punctuation and coordination conjunctions, this feature takes the value 1 if the word occurs in the same segment with the target reference, and 0 otherwise.
Part of speech tag	The part of speech tag of the word, the word before, and the word after.
Dependency Distance	Length of the shortest dependency path (in the dependency parse tree) that connects the word to the target reference or its representative. It has been shown in previous work on relation extraction that the shortest path between any two entities captures the information required to assert a relationship between them [35]
Dependency Relations	This item includes a set of features. Each features corresponds to a dependency relation type. If the relation appears in the dependency path that connects the word to the target reference or its representative, its corresponding feature takes the value 1, and 0 otherwise.
Common Ancestor Node	The type of the node in the syntactic parse tree that is the least common ancestor of the word and the target reference.
Syntactic Distance	The number of edges in the shortest path that connects the word and the target reference in the syntactic parse tree.

Table 8.1: The features used for word classification and sequence labeling

Word Classification

In this method we define reference scope identification as a classification task of the individual words of the citation sentence. Each word is classified as *inside* or *outside* the scope of a given target reference. We use a number of linguistic and structural features to train a classification model on a set of labeled sentences. The trained model is then used to label new sentences. The features that we use to train the model are listed in Table 8.1. We use the Stanford parser [104] for syntactic and dependency parsing. We experiment with two classification algorithms: Support Vector Machines (SVM) and logistic regression.

Sequence Labeling

In the method described in Section 8.3.2 above, we classify each word independently from the labels of the nearby words. The nature of our task, however, suggests that the accuracy of word classification can be improved by considering the labels of the words surrounding the word being classified. It is very likely that the word takes the same label as the word before and after it if they all belong to the same clause in the sentence. In this method we define the problem as a sequence labeling task. Now, instead of looking for the best label for each word individually, we look for the globally best sequence of labels for all the words in the sentence at once.

We use Conditional Random Fields (CRF) as our sequence labeling algorithm. In particular, we use first-order chain-structured CRF. The chain consists of two sets of nodes: a set of hidden nodes $\mathbf{Y}$ which represent the scope labels (0 or 1) in our case, and a set of observed nodes $\mathbf{X}$ which represent the observed features. The task is to estimate the probability of a sequence of labels Y given the sequence of observed features X: $P(\mathbf{Y}|\mathbf{X})$

Lafferty et al. [107] define the this probability to be a normalized product of potential functions ψ:

$$P(\mathbf{y}|\mathbf{x}) = \prod_t \psi_k(y_t, y_{t-1}, \mathbf{x}) \tag{8.1}$$

Where $\psi_k(y_t, y_{t-1}, \mathbf{x})$ is defined as

$$\psi_k(y_t, y_{t-1}, \mathbf{x}) = exp(\sum_k \lambda_k f(y_t, y_{t-1}, \mathbf{x})) \tag{8.2}$$

where $f(y_t, y_{t-1}, \mathbf{x})$ is a transition feature function of the label at positions $i-1$ and i and the observation sequence $\mathbf{x}$; and λ_j is parameter to be estimated from training data. We use, as the observations at each position, the same features that we used in Section 8.3.2 above (Table 8.1).

Segment Classification

We noticed that the scope of a given reference often consists of units of higher granularity than words. Therefore, in this method, we split the sentence into segments of contiguous words and, instead of labeling individual words, we label the whole segment as *inside* or *outside* the scope of the target reference. We experimented with two different segmentation methods. In the first method (method-1), we segment the sentence at punctuation marks, coordination conjunctions, and a set of special expressions such as ”for example”, ”for instance”, ”including”, ”includes”, ”such as”, ”like”, etc. Sentence (8) below shows an example of this segmentation method (Segments are enclosed in square brackets).

(8) [Rerankers have been successfully applied to numerous NLP tasks such as] [parse selection (GTREF)], [parse reranking (GREF)], [question-answering (REF)].

In the second segmentation method (method-2), we split the sentence into segments of finer granularity. We use a chunking tool to identify noun groups, verb groups, preposition groups, adjective groups, and adverb groups. Each such group (or chunk) forms a segment. If a word does not belong to any chunk, it forms a singleton segment by itself. Sentence (9) below shows an example of this segmentation method (Segments are enclosed in square brackets).

(9) [To] [score] [the output] [of] [the coreference models], [we] [employ] [the commonly-used MUC scoring program (REF)] [and] [the recently-developed CEAF scoring pro-

gram (TREF)].

We assign a label to each segment in two steps. In the first step, we use the sequence labeling method described in Section 8.3.2 to assign labels to all the individual words in the sentence. In the second step, we aggregate the labels of all the words contained in a segment to assign a label to the whole segment. We experimented with three different label aggregation rules: 1) rule-1: assign to the segment the majority label of the words it contains, and 2) rule-2: assign to the segment the label 1 (i.e., *inside*) if at least one of the words contained in the segment is labeled 1, and assign the label 0 to the segment otherwise, and 3) rule-3: assign the label 0 to the segment if at least of the words it contains is labeled 0, and assign 1 otherwise.

8.4 Evaluation

8.4.1 Data

We use the ACL Anthology Network corpus (AAN) [160] in our evaluation. AAN is a publicly available collection of more than 19,000 NLP papers. AAN provides a manually curated citation network of its papers and the citation sentence(s) associated with each edge. The current release of AAN contains about 76,000 unique citation sentences. From this set, we randomly selected 3500 citation sentences, each containing at least two references (3.75 references on average with a standard deviation of 2.5). The total number of references in this set of sentences is 19,591. [1]

We split the data set into two random subsets: a development set (200 sentences) and a training/testing set (3300 sentences). We used the development set to study the data and develop our strategies of addressing the problem. The second set was used to train and test the system in a cross-validation mode.

[1] The data and the code will be made public at the time of publication

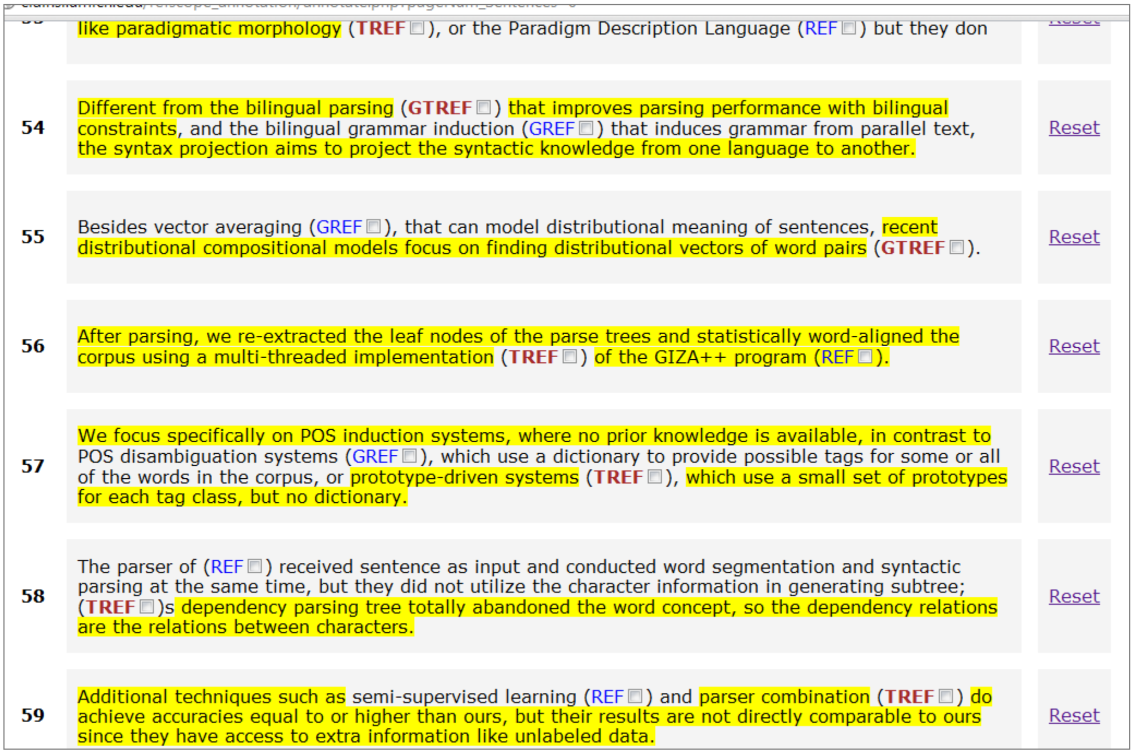
8.4.2 Annotation

We asked graduate students with good background in NLP (the area of the annotated sentences) to provide three annotations for each sentence in the data set described above. First, we asked them to determine whether each of the references in the sentence was correctly tagged or not. Second, we asked them to determine for each reference whether it is a syntactic constituent in the sentence or not. Third, we asked them to determine and label the scope of one reference in each sentence which was marked as a target reference (TREF). We designed a user-friendly tool to collect the annotations from the students (figure 8.2).

To estimate the inter-annotator agreement, we picked 500 random sentences from our data set and assigned them to two different annotators. The inter-annotator agreement was perfect on both the reference tagging annotation and the reference syntacticality annotation. This is expected since both are objective, clear, and easy tasks. To measure the inter-annotator agreement on the scope annotation task, we deal with it as a word classification task. This allows us to use the popular classification agreement measure, the Kappa coefficient [106]. The Kappa coefficient is defined as follows:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{8.3}$$

where P(A) is the relative observed agreement among raters and P(E) is the hypothetical probability of chance agreement. The agreement between the two annotators on the scope identification task was $K = 0.61$. On Landis and Kochs [109] scale, this value indicates substantial agreement.



	like paradigmatic morphology (TREF □), or the Paradigm Description Language (REF □) but they don	
54	Different from the bilingual parsing (GTREF □) that improves parsing performance with bilingual constraints, and the bilingual grammar induction (GREF □) that induces grammar from parallel text, the syntax projection aims to project the syntactic knowledge from one language to another.	Reset
55	Besides vector averaging (GREF □), that can model distributional meaning of sentences, recent distributional compositional models focus on finding distributional vectors of word pairs (GTREF □).	Reset
56	After parsing, we re-extracted the leaf nodes of the parse trees and statistically word-aligned the corpus using a multi-threaded implementation (TREF □) of the GIZA++ program (REF □).	Reset
57	We focus specifically on POS induction systems, where no prior knowledge is available, in contrast to POS disambiguation systems (GREF □), which use a dictionary to provide possible tags for some or all of the words in the corpus, or prototype-driven systems (TREF □), which use a small set of prototypes for each tag class, but no dictionary.	Reset
58	The parser of (REF □) received sentence as input and conducted word segmentation and syntactic parsing at the same time, but they did not utilize the character information in generating subtree; (TREF □)s dependency parsing tree totally abandoned the word concept, so the dependency relations are the relations between characters.	Reset
59	Additional techniques such as semi-supervised learning (REF □) and parser combination (TREF □) do achieve accuracies equal to or higher than ours, but their results are not directly comparable to ours since they have access to extra information like unlabeled data.	Reset

Figure 8.2: A screen shot of the tool used for annotation

8.4.3 Experimental Setup

We use the Edinburgh Language Technology Text Tokenization Toolkit (LT-TTT) [68] for text tokenization, part-of-speech tagging, chunking, and noun phrase head identification. We use the Stanford parser [104] for syntactic and dependency parsing. We use LibSVM [38] for Support Vector Machines (SVM) classification. Our SVM model uses a linear kernel with all the parameters set to LIBSVM default configurations. We use Weka [72] for logistic regression classification. We use the Machine Learning for Language Toolkit (MALLET) [125] for CRF-based sequence labeling. In all the scope identification experiments and results below, we use 10-fold cross validation for training/testing.

8.4.4 Preprocessing Component Evaluation

We ran our three rule-based preprocessing modules on the testing data set and compared the output to the human annotations. The test set was not used in the tuning of the system but was done using the development data set as described above. We report the results for each of the preprocessing modules. Our reference tagging module achieved 98.3% precision and 93.1% recall. Most of the errors were due to issues with text extraction from PDF or due to bad references practices by some authors (i.e., not following scientific referencing standards). Our reference grouping module achieved perfect accuracy for all the correctly tagged references. This was expected since this is a straightforward task. The non-syntactic reference removal module achieved 90.08% precision and 90.1% recall. Again, most of the errors were the result of bad referencing practices by the authors.

8.4.5 Reference Scope Identification Experiments

We conducted several experiments to compare the methods proposed in Section 8.3 and their variants. We ran all the experiments on the training/testing set (the 3300 sentences) described in Section 8.4.1. The experiments that we ran are as follows: 1) word classification using a SVM classifier (WC-SVM); 2) word classification using a logistic regression classifier(WC-LR); 3) CRF-based sequence labeling (SL-CRF); 4) segment classification using segmentation method-1 and label aggregation rule-1 (SC-S1-R1); 5,6,7,8,9) same as (4) but using different combinations of segmentation methods 1 and 2, and label aggregation rules 1,2 and 3: SC-S1-R2, SC-S1-R3, SC-S2-R1, SC-S2-R2, SC-S2-R3 (where Sx refers to segmentation method x and Ry refers to label aggregation rule y all as explained in Section 8.3.2); and finally, 10) the unsupervised approach using parse tree splitting (PT).

	Method	Output
Example 1	Word Classification (WC-SVM)	A wide range of contextual information, such as surrounding words (GREF), dependency or case structure (GTREF), and dependency path (GREF), has been utilized for similarity calculation, and achieved considerable success.
	Sequence Labeling (SL-CRF)	A wide range of contextual information, such as surrounding words (GREF), dependency or case structure (GTREF), and dependency path (GREF), has been utilized for similarity calculation, and achieved considerable success.
	Segment Classification (SC-S2-R1)	A wide range of contextual information, such as surrounding words (GREF), dependency or case structure (GTREF), and dependency path (GREF), has been utilized for similarity calculation, and achieved considerable success.
Example 2	Word Classification (WC-SVM)	Some approaches have used WordNet for the generalization step (GTREF), others EM-based clustering (REF).
	Sequence Labeling (SL-CRF)	Some approaches have used WordNet for the generalization step (GTREF), others EM-based clustering (REF).
	Segment Classification (SC-S2-R1)	Some approaches have used WordNet for the generalization step (GTREF), others EM-based clustering (REF).

Table 8.2: Two example outputs produced by the three methods

To better understand which of the features listed in Table 8.1 are more important for the task, we use Guyon et al.s [71] method for feature selection using SVM to rank the features based on their importance. The results of the experiments and the feature analysis are presented and discussed in the following subsection.

8.4.6 Results and Discussion

Experimental Results

We ran the experiments described in the previous subsection on the testing data described in Section 8.4.1. Table 8.3 compares the precision, recall, F1, and accuracy for the three methods described in Section 8.3 and their variations. All the metrics were computed at the word level. The results show that all our methods outperform the baseline method AR-2011 that was proposed by Abu-Jbara and Radev [10]. In the word classification method, we notice no significant difference between the performance of the SVM vs Logistic Regression classifier. We also notice that the CRF-based sequence labeling method performs significantly better than the word classification method. This result corroborates our intuition that the labels of neigh-

Method	Accuracy	Precision	Recall	F-measure
WC-SVM	74.9%	74.5%	93.4%	82.9%
WC-LR	74.3%	76.8%	88.0%	82.0%
SL-CRF	78.2%	80.1%	94.2%	86.6%
SC-S1-R1	73.7%	72.1%	**97.8%**	83.0%
SC-S1-R2	69.3%	68.4%	98.9%	80.8%
SC-S1-R3	60.0%	61.8%	73.3%	60.9%
SC-S2-R1	**81.8%**	**81.2%**	93.8%	**87.0%**
SC-S2-R2	78.2%	77.3%	94.9%	85.2%
SC-S2-R3	66.1%	67.1%	71.2%	69.1%
PT	54.0%	63.3%	33.1%	41.5%

Table 8.3: Results of scope identification using the different algorithms described in the paper

boring words are dependent. The results also show that segment labeling generally performs better than word labeling. More specifically, the results indicate that segmentation based on chunking and the label aggregation based on plurality when used together (i.e., SC-S2-R1) achieve higher precision, accuracy, and F-measure than the punctuation-based segmentation and the other label aggregation rules.

Table 8.2 shows the output of the three methods on two example sentences. The underlined words are labeled by the system as scope words.

Feature Analysis

We performed an analysis of our classification features using Guyon et al. [71] method. The analysis revealed that both structural and syntactic features are important. Among the syntactic features, the dependency path is the most important. Among the structural features, the *segment* feature (as described in Table 8.1) is the most important.

8.5 Conclusions

We presented and compared three different methods for reference scope identification: word classification, sequence labeling, and segment classification. Our results indicate that segment classification achieves the best performance. The next direction in this research is to extract the scope of a given reference as a standalone grammatical sentence. In many cases, the scope identified by our method can form a grammatical sentence with no or minimal postprocessing. In other cases, more advanced *text regeneration* techniques are needed for scope extraction.

CHAPTER IX

Identifying Citation Purpose and Polarity

An objective and fair evaluation of the impact of published research requires both quantitative and qualitative assessment. Existing bibliometric measures such as *H-Index* [84, 85], *G-index* [52], and *Impact Factor* [63] focus on the quantitative aspect of this evaluation which dose not always correlate with the qualitative aspect.

For example, the number of papers published by a researcher only tells how productive she or he is. It does not say anything about the quality or the impact of the work. Similarly, the number of citations that a paper receives should not be used to gauge the quality of the work as it really only measures the popularity of the work and the interest of other researchers in it [60]. Controversial papers or those based on fabricated data or experiments may receive a large number of citations. A popular example of fraudulent research that deceived many researchers and caught media attention was the case of a South Korean research scientist, Hwang Woo-suk, who was found to have faked his research results in the area of human stem cell cloning. His research was published in *Science* and received close to 200 citations after the fraud was discovered. The vast majority of those citations were negative.

This suggests that the *purpose* of citation should be taken into consideration when biblometric measures are computed. Negative citations should be weighted less than positive or neutral citations. This motivates the need to automatically distinguish between positive, negative, and neutral citations and to identify the purpose of a citation; i.e. the author's intention behind choosing a published article and citing it.

This analysis of citation purpose and polarity can be useful for many applications. For example, it can be used to build systems that help funding agencies and hiring committees at universities and research institutions evaluate researchers' work more accurately. It can also be used as a preprocessing step in systems that process scholarly data. For example, citation-based summarization systems [152, 155, 11] and survey generation systems [131, 154] can benefit from citation purpose and polarity analysis to improve paper and content selection.

In this chapter, we investigate the use of linguistic analysis techniques to automatically identify the purpose of citing a paper and the polarity of this citation. We first present a sequence labeling method for extracting the text that cites a given target reference; i.e. the text that appears in a scientific article and refers to another article and comments on it. We use the term *citation context* to refer to this text. Next, we use supervised classification techniques to analyze this text and identify the purpose and polarity of citation.

The rest of this chapter is organized as follows. Section 9.1 reviews the related work. We present our approach in Section 9.2. We then describe the data and experiments in Section 9.3. Finally, Section XII concludes the chapter.

9.1 Related Work

Our work is related to a large body of research on citations. Studying citation patterns and referencing practices has interested researchers for many years [87, 61]. White [196] provides a good survey of the different research directions that study or use citations. In the following subsections, we review three lines of research that are closely related to our work.

9.1.1 Citation Context Identification

The first line of related research addresses the problem of identifying citation context. The context of a citation that cites a given target paper can be a set of sentences, one sentence, or a fragment of a sentence.

Nanba and Okumura [143] use the term *citing area* to refer to the same concept. They define the citing area as the succession of sentences that appear around the location of a given reference in a scientific paper and have connection to it. Their algorithm starts by adding the sentence that contains the target reference as the first member sentence in the citing area. Then, they use a set of cue words and hand-crafted rules to determine whether the surrounding sentences should be added to the citing area or not. In [142], they use their algorithm to improve citation type classification and automatic survey generation.

Qazvinian and Radev [153] addressed a similar problem. They proposed a method based on probabilistic inference to extract non-explicit citing sentences; i.e., sentences that appear around the sentence that contains the target reference and are related to it. They showed experimentally that citation-based survey generation produces better results when using both explicit and non-explicit citing sentences rather than using the explicit ones alone.

In previous work, we addressed the issue of identifying the scope of a given target reference in citing sentences that contain multiple references [14]. Our definition of *reference scope* was limited to fragments of the explicit citing sentence (i.e. the sentence in which actual citation appears). That method does not identify related text in surrounding sentences.

In this work, we propose a supervised sequence labeling method for identifying the citation context of given reference which includes the explicit citing sentence and the related surrounding sentences.

9.1.2 Citation Purpose Classification

Several research efforts have focused on studying the different purposes for citing a paper [62, 194, 139, 1, 32]. Bonzi [32] studied the characteristics of citing and cited works that may aid in determining the relatedness between them. Garfield [62] enumerated several reasons why authors cite other publications, including "alerting researchers to forthcoming work", paying homage to the leading scholars in the area, and citations which provide pointers to background readings. Weinstock [194] adopted the same scheme that Garfield proposed in her study of citations.

Spiegel-Rosing [176] proposed 13 categories for citation purpose based on her analysis of the first four volumes of Science Studies. Some of them are: Cited source is the specific point of departure for the research question investigated, Cited source contains the concepts, definitions, interpretations used, Cited source contains the data used by the citing paper. Nanba and Okumura [143] came up with a simple schema composed of only three categories: *Basis*, *Comparison*, and other *Other*. They proposed a rule-based method that uses a set of statistically selected cue words to determine the category of a citation. They used this classification as a first step for scientific paper summarization. Teufel et al. [184], in their work on citation function classification, adopted 12 categories from Spiegel-Rosing's taxonomy. They trained an SVM classifier and used it to label each citing sentence with exactly one category. Further, they mapped the twelve categories to four top level categories namely: weakness, contrast (4 categories), positive (6 categories) and neutral.

The taxonomy that we use in this work is based on previous work. We adopt a scheme that contains six categories. We selected the six categories after studying all the previously used citation taxonomies. We included the ones we believed are important for improving bibliometric measures and for the applications that we are planning to pursue in the future (Section XII).

9.1.3 Citation Polarity Classification

The polarity (or sentiment) of a citation has also been studied previously. Previous work showed that positive and negative citations are common, although negative citations might be expressed indirectly or in an implicit way [215, 121, 186]. Athar [23] addressed the problem of identifying sentiment in citing sentences. He used a set of structure-based features to train a machine learning classifier using annotated data. This work uses the citing sentence only to predict sentiment. Context sentences were ignored. Athar and Teufel [24] observed that taking the context into consideration when judging sentiment in citations increases the number of negative citations by a factor of 3. They proposed two methods for utilizing the context. In the first method, they treat the citing sentence and a fixed context (a window of four sentences around the citing sentence) as if they were a single sentence. They extract features from the merged text and train a classifier similar to what they did in their 2011 paper. In the second method, they use a four-class annotation scheme. Each sentence in a window of four sentences around the citation is labeled as positive, negative, neutral, or excluded (unrelated to the cited work). There experiments surprisingly gave negative results and showed that classifying sentiment *without* considering the context achieves better results. They attributed this to the small size of their training data and to the noise that including the context text introduces to the data. In [25], the authors present a method for automatically identifying all the mentions of the cited paper in the citing paper. They show that considering all the mentions improves the

Feature	Description
Demonstrative determiners	Takes a value of 1 if the current sentence contains contains a *demonstrative determiner* (this, these, etc.), and 0 otherwise.
Conjunctive adverbs	Takes a value of 1 if the current sentence starts with a *conjunctive adverb* (However, Furthermore, Accordingly, etc.), and 0 otherwise.
Position	Position of the current sentence with respect to the citing sentence. This feature takes one of four values: -1, 0, 1, and 2.
Contains Closest Noun Phrase	Takes a value of 1 if the current sentence contains closest noun phrase (if any) immediately before the reference position in the citing sentence, and 0 otherwise. This noun phrase often is the name of a method, a tool, or corpus originating from the cited reference.
2-3 grams	The first bigram and trigram in the sentence (*This approach*, *One problem with*, etc.).
Contains Other references	Takes a value of 1 if the current sentence contains references other than the target, and 0 otherwise.
Contains a Mention of target reference	Takes a value of 1 if the current sentence contains a mention (explicit or anaphoric) of the target reference, and 0 otherwise.
Multiple references	Takes a value of 1 if the citing sentence contains multiple references, and 0 otherwise. If the citing sentence contains multiple references, it becomes less likely that the surrounding sentences are related.

Table 9.1: Features used for citation context identification

performance of detecting sentiment in citations.

In our work, we propose a sequence labeling method for identifying the citation context first, and then use a supervised approach to determine the polarity of a given citation.

9.2 Approach

In this section, we describe our approach to three tasks: citation context identification, citation purpose classification, and citation polarity identification. We also describe a preprocessing stage that is applied to the citation text before performing any of the three tasks.

9.2.1 Preprocessing

The goal of the preprocessing stage is to clean and prepare the citation text for part-of-speech tagging and parsing. The available POS taggers and parsers are not trained on citation text. Citation text is different from normal text in that it contains references written in a special format (e.g., author names and publication year written in parentheses; or reference indices written in square brackets). Many citing sentences contain multiple references, some of which might be grouped together in a pair of parentheses and separated by a comma or a semi-colons. These references are usually not syntactic nor semantic constituents of the sentences they appear in. This results in many POS tagging and parsing errors. We address this issue in the pre-processing stage to improve the performance of the feature extraction component. We perform three pre-processing steps:

a. Reference Tagging: In the first step, we find and tag all the references that appear in the text. We use a regular expression to find references and replace each reference with a placeholder. The reference to the target paper is replaced by the placeholder *TREF*. Each other reference is replaced by *REF*.

b. Reference Grouping: In this step, we identify grouped references (i.e. multiple references listed between one pair of parentheses separated by semi-colons). Each such group is replaced by a placeholder, *GREF*. If the target reference is a member of the group, we use a different placeholder: *GTREF*.

c. Non-syntactic Reference Removal: A reference or a group of references could either be a syntactic constituent and has a semantic role in the sentence or not [195, 14]. If the reference is not a syntactic component in the sentence, we remove it to reduce parsing errors. Following our previous work [14], we use a rule-based algorithm to determine whether a reference should be removed from the sentence or kept. The algorithm uses stylistic and linguistic features such as the style of the reference, the position of the reference, and the surrounding words to make the decision. When a reference is removed, the head of the closest noun phrase (NP) immediately before the position of the removed reference is used as a representative of the reference. This is needed for feature extraction as shown later in the paper.

9.2.2 Citation Context Identification

The task of identifying the citation context of a given target reference can be formally defined as follows. Given a scientific article A that cites another article B, find a set of sentences in A that talk about the work done in B such that at least one of these sentences contains an explicit reference to B.

We treat this problem as a sequence labeling problem. The goal is to find the globally best sequence of labels for all the sentences that appear within a window around the *citing sentence*. The *citing sentence* is the one that contains an explicit reference to the cited paper. Each sentence within the window is labeled as *INCLUDED* or *EXCLUDED* from the citation context of the given target paper. To determine the size of the window, we examined a development set of 300 sentences. We noticed that the related context almost always falls within a window of four sentences. The window includes the citing sentence, one sentence before the citing sentence, and two sentences after the citing sentence.

We use Conditional Random Fields (CRFs) for sequence labeling. In particular, we use a first-order chain-structured CRF. The chain consists of two sets of nodes: 1) a set of hidden nodes $\mathbf{Y}$ which represent the context labels of sentences (INCLUDED or EXCLUDED), and 2) a set of observed nodes $\mathbf{X}$ which represent the features extracted from the sentences. The task is to estimate the probability of a sequence of labels Y given the sequence of observed features X: $P(\mathbf{Y}|\mathbf{X})$

Lafferty et al. [107] define this probability to be a normalized product of potential functions ψ:

Category	Description	Example
Criticizing	Criticism can be positive or negative. A citing sentence is classified as ”criticizing” when it mentions the weakness/strengths of the cited approach, negatively/positively criticizes the cited approach, negatively/positively evaluates the cited source.	Chiang (2005) introduced a constituent feature to reward phrases that match a syntactic tree but did not yield significant improvement.
Comparison	A citing sentence is classified as ”comparison” when it compares or contrasts the work in the cited paper to the author's work. It overlaps with the first category when the citing sentence says one approach is not as good as the other approach. In this case we use the first category.	Our approach permits an alternative to minimum error-rate training (MERT; Och, 2003);
Use	A citing sentence is classified as ”use” when the citing paper uses the method, idea or tool of the cited paper.	We perform the MERT training (Och, 2003) to tune the optimal feature weights on the development set.
Substantiating	A citing sentence is classified as ”substantiating” when the results, claims of the citing work substantiate, verify the cited paper and support each other.	It was found to produce automated scores, which strongly correlate with human judgements about translation fluency (Papineni et al. , 2002).
Basis	A citing sentence is classified as ”basis” when the author uses the cited work as starting point or motivation and extends on the cited work.	Our model is derived from the hidden-markov model for word alignment (Vogel et al., 1996; Och and Ney, 2000).
Neutral (Other)	A citing sentence is classified as ”neutral” when it is a neutral description of the cited work or if it doesn't come under any of the above categories.	The solutions of these problems depend heavily on the quality of the word alignment (Och and Ney, 2000).

Table 9.2: Annotation scheme for citation purpose. Motivated by the work of [176] and [184]

$$P(\mathbf{y}|\mathbf{x}) = \prod_t \psi_k(y_t, y_{t-1}, \mathbf{x}) \tag{9.1}$$

Where $\psi_k(y_t, y_{t-1}, \mathbf{x})$ is defined as

$$\psi_k(y_t, y_{t-1}, \mathbf{x}) = exp(\sum_k \lambda_k f(y_t, y_{t-1}, \mathbf{x})) \tag{9.2}$$

where $f(y_t, y_{t-1}, \mathbf{x})$ is a transition feature function of the label at positions $i-1$ and i and the observation sequence $\mathbf{x}$; and λ_j is a parameter that the algorithm estimates from training data.

The features we use to train the CRF model include structural and lexical features that attempt to capture indicators of relatedness to the given target reference. The features that we used and their descriptions are listed in table 9.1.

9.2.3 Citation Purpose Classification

In this section, we describe the citation purpose classification task. Given a target paper B and its citation context (extracted using the method described above) in a given article A, we want to determine the purpose of citing B by A. The purpose is defined as intention behind selecting B and citing it by the author of A [62].

We use a taxonomy that consists of six categories. We designed this taxonomy based on our study of similar taxonomies proposed in previous work. We selected the categories that we believe are more important and useful from a bibliometric point of view, and the ones that can be detected through citation text analysis. We also tried to limit the number of categories by grouping similar categories proposed in previous work under one category. The six categories, their descriptions, and an example for each category are listed in Table 9.2.

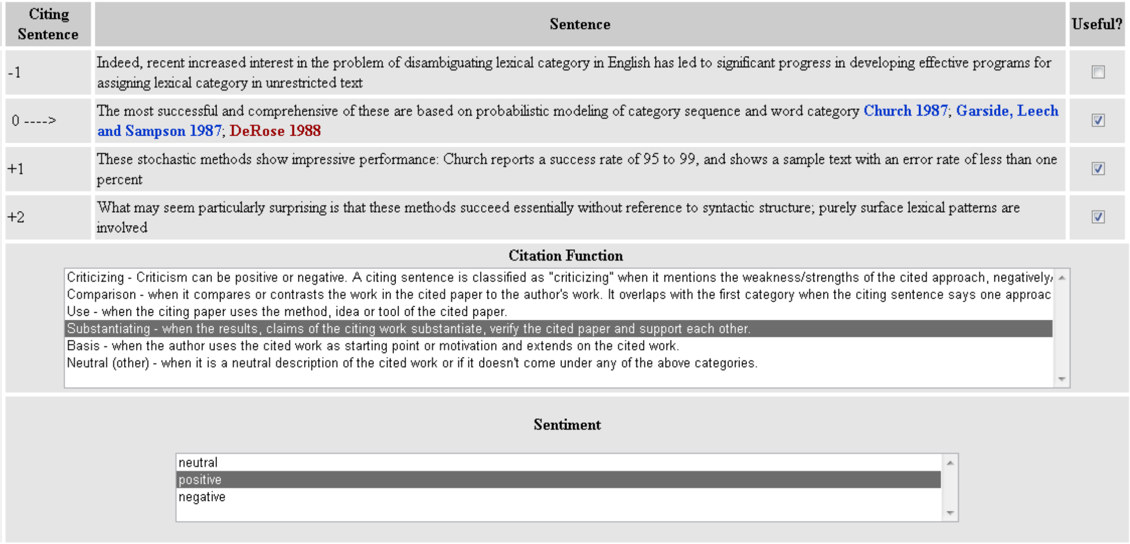
Feature	Description
Reference count	The number of references that appear in the citation context.
Is Separate	Whether the target reference appears within a group of references or separate (i.e. single reference).
Closest Verb / Adjective / Adverb	The lemmatized form of the closest verb/adjective/adverb to the target reference or its representative or any mention of it. Distance is measure based on the shortest path in the dependency tree.
Self Citation	Whether the citation from the source paper to the target reference is a self citation.
Contains 1st/3rd PP	Whether the citation context contains a first/third person pronoun.
Negation	Whether the citation context contains a negation cue. The list of negation cues is taken from the training data of the *SEM 2012 negation detection shared task [134].
Speculation	Whether the citation context contains a speculation cue. The list is taken from Quirk et al. [157]
Closest Subjectivity Cue	The closest subjectivity cue to the target reference or its representative or any anaphoric mention of it. The list of cues is taken from OpinionFinder [203]
Contrary Expressions	Whether the citation context contains a contrary expression. The list is taken from Biber [30]
Section	The headline of the section in which the citation appears. We identify five title categorizes: 1) *Introduction, Motivation, etc.* 2) *Background, Prior Work, Previous Work, etc.* 3) *Experiments, Data, Results, Evaluation, etc.* 4) *Discussion, Conclusion, Future work, etc.*. 5) All other section headlines. Headlines are identified using regular expressions.
Dependency Relations	All the dependency relations that appear in the citation context. For example, $nsubj(outperform, algorithm)$ is one of the relations extracted from "This algorithm outperforms the one proposed by...". The arguments of the dependency relation are replaced by their lemmatized forms. This type of features has been shown to give good results in similar tasks [24].

Table 9.3: The features used for citation purpose and polarity classification

We use a supervised approach whereby a classification model is trained on a number of lexical and structural features extracted from a set of labeled citation contexts. Some of the features that we use to train the classifier are listed in table 9.3.

9.2.4 Citation Polarity Identification

In this section, we describe the citation polarity identification task. Given a target paper B and its citation context in a given article A, we want to determine the polarity of the citation text with respect to B. The polarity can be: *positive*,



Citing Sentence	Sentence	Useful?
-1	Indeed, recent increased interest in the problem of disambiguating lexical category in English has led to significant progress in developing effective programs for assigning lexical category in unrestricted text	☐
0 ---->	The most successful and comprehensive of these are based on probabilistic modeling of category sequence and word category **Church 1987**; **Garside, Leech and Sampson 1987**; **DeRose 1988**	☑
+1	These stochastic methods show impressive performance: Church reports a success rate of 95 to 99, and shows a sample text with an error rate of less than one percent	☑
+2	What may seem particularly surprising is that these methods succeed essentially without reference to syntactic structure; purely surface lexical patterns are involved	☑

Citation Function

Criticizing - Criticism can be positive or negative. A citing sentence is classified as "criticizing" when it mentions the weakness/strengths of the cited approach, negatively,
Comparison - when it compares or contrasts the work in the cited paper to the author's work. It overlaps with the first category when the citing sentence says one approac
Use - when the citing paper uses the method, idea or tool of the cited paper.
Substantiating - when the results, claims of the citing work substantiate, verify the cited paper and support each other.
Basis - when the author uses the cited work as starting point or motivation and extends on the cited work.
Neutral (other) - when it is a neutral description of the cited work or if it doesn't come under any of the above categories.

Sentiment

neutral
positive
negative

Figure 9.1: The screen shot of the interface used for annotating the data

negative, or *neutral (objective)*. Positive, negative, and neutral in this context are defined in a slightly different way than their usual sense. A citation is marked positive if it either explicitly states a strength of the target paper or indicates that the work done in the target paper has been used either by the author or a third-party. It is also marked as positive if it is compared to another paper (possibly by the same authors) and deemed better in some way. A citation is marked negative if it explicitly points to a weakness of the target paper. It is also marked as negative if it is compared to another paper and deemed worse in some way. A citation is marked as neutral if it is only descriptive.

Similar to citation purpose classification, we use a supervised approach for this problem. We train a classification model using the same features listed in Table 9.3. Due to the high skewness in the data (more than half of the citations are neutral), we use two setups for binary classification. In the first setup, the citation is classified as *Polarized (Subjective)* or *(Neutral) Objective*. In the second one, *Subjective* citations are classified as *Positive* or *Negative*. We find that this method gives more intuitive

Category	Example
Positive	Three approaches are dominating, i.e. knowledge-based approach (REF), information retrieval-based approach (REF) and machine learning approach (TREF), in which the last approach is found very popular.
Negative	Mining the Web for bilingual text (TREF) is not likely to provide sufficient quantities of high quality data.
Neutral	There has been work on detecting relations within noun phrases (GREF), clauses (REF) and syntax-based comma resolution (TREF).

Table 9.4: Examples of positive, negative, and neutral citing sentences. The reference marked as TREF is the target reference.

results than using a 3-way classifier.

9.3 Evaluation

In this section, we describe the data that we used for evaluation and the experiments that we conducted.

9.3.1 Data

We use the ACL Anthology Network corpus (AAN) [160, 161] in our evaluation. AAN is a publicly available collection of more than 19,000 NLP papers. It includes a manually curated citation network of its papers as well as the full text of the papers and the citing sentences associated with each edge in the citation network. From this set, we selected 30 papers that have different numbers of incoming citations and that were consistently cited since they were published. These 30 papers received a total of about 3,500 citations from within AAN (average = 115 citation/paper, Min = 30, and Max = 338). These citations come from 1,493 unique papers. For each of these citations, we extracted a window of 4 sentences around the reference position. This brings the number of sentences in our dataset to a total of roughly 14,000 sentences. We refer to this dataset as *training/testing dataset.*

In addition to this dataset, we created another dataset that contains 300 citations that cite 5 papers from AAN. We refer to this dataset as the *development* dataset.

This dataset was used to determine the size of the citation context window, and to develop the feature sets used in the three tasks described in Section 9.2 above.

9.3.2 Annotation

In this section, we describe the annotation process. We asked graduate students with good background in NLP (the topic of the annotated sentences) to provide three annotations for each citation example (a window of 4 sentences around the reference anchor) in the *training/testing dataset.* We asked them to mark the sentences that are related to a given target reference. In addition, we asked them to determine the purpose of citing the target reference by choosing from the six purpose categories that we described earlier. We also asked them to determine whether the citation is negative, positive, or neutral.

To estimate the inter-annotator agreement, we picked 400 sentences from the *training/testing dataset* and assigned them to two different annotators. We use the Kappa coefficient [43] to measure the agreement. The Kappa coefficient is defined as follows:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{9.3}$$

where P(A) is the relative observed agreement among annotators and P(E) is the hypothetical probability of chance agreement. The agreement between the two annotators on the context identification task was $K = 0.89$. On Landis and Kochs [109] scale, this value indicates *almost perfect* agreement. The agreement on the purpose and the polarity classification task were $K = 0.61$ and $K = 0.66$, respectively; which indicates *substantial agreement* on the same scale.

The annotation shows that in 22% of the citation examples, the citation context consists of 2 or more sentences. The distribution of the purpose categories in the data was: 14.7% criticism, 8.5% comparison, 17.7% use, 7% substantiation, 5% basis, and 47% other. The distribution of the polarity categories was: 30% positive, 12% negative, and 58% neutral.

9.3.3 Experimental Setup

We use the CRF++[1] toolkit for CRF training and testing. We use the Stanford parser to parse the citation text and generate the dependency parse trees of sentences. We use Weka for classification experiments. We experimented with several classifiers including: SVM, Logistic Regression (LR), and Naive Bayes. All the experiments that we conducted used the *training/testing dataset* in a 10-fold cross validation mode. All the results have been tested for statistical significance using a 2-tailed paired t-test.

9.3.4 Evaluation of Citation Context Identification

We compare the CRF approach to three baselines. The first baseline (ALL) labels all the sentences in the citation window of size 4 as *INCLUDED* in the citation context. The second baseline (CS-ONLY) labels the citing sentence only as *INCLUDED* in the citation context. In the third baseline, we use a supervised classification method instead of sequence labeling. We use Support Vector Machines (SVM) to train a model using the same set of features as in the CRF approach.

Table 9.5 shows the precision, recall, and F1 score of the CRF approach and the baselines. The results show that our CRF approach outperforms all the baselines. It also asserts our expectation that addressing this problem as a sequence labeling problem leads to better performance than individual sentence classification, which is also clear from the nature of the task.

[1] http://crfpp.googlecode.com/svn/trunk/doc/index.html

	Precision	Recall	F1
CRFs	**98.5%**	**82.0%**	**89.5%**
ALL	30.7%	100.0%	46.9%
CS-ONLY	88.0%	74.0%	80.4%
SVM	92.0%	76.4%	83.5%

Table 9.5: Results of citation context identification

	Criticism	Comparison	Use	Substantiating	Basis	Other
Precision	53.0%	55.2%	60.0%	50.1%	47.3%	64.0%
Recall	77.4%	43.1%	73.0%	57.3%	39.1%	85.1%
F1	63.0%	48.4%	66.0%	53.5%	42.1%	73.1%
Accuracy: 70.5% Macro-F: 58.0%						

Table 9.6: Summary of Citation Purpose Classification Results (10-fold cross validation, SVM: Linear Kernel, c = 1.0)

Feature Analysis: We evaluated the importance of the features listed in Table 9.1 by computing the chi-squared statistic for every feature with respect to the class. We found that the lexical features (such as determiners and conjunction adverbs) are generally more important than the structural features (such as position and reference count). The features shown in Table 9.1 are listed in the order of their importance based on this analysis.

9.3.5 Evaluation of Citation Purpose Classification

Our experiments with several classification algorithms showed that the SVM classifier outperforms Logistic Regression and Naive Bayes classifiers. Due to space limitations, we only show the results for SVM. Table 9.6 shows the precision, recall, and F1 for each of the six categories. It also shows the overall accuracy and the Macro-F measure.

Feature Analysis: The chi-squared evaluation of the features listed in Table 9.3 shows that both lexical and structural features are important. It also shows that among lexical features, the ones that are limited to the existence of a direct relation to the target reference (such as *closest* verb, adjective, adverb, subjective cue, etc.) are most useful. This can be explained by the fact that the restricting the features to having direct dependency relation introduces much less noise than other features (such as *Dependency Triplets*). Among the structural features, the number of references in the citation context showed to be more useful.

9.3.6 Evaluation of Citation Polarity Identification

Similar to the case of citation purpose classification, our experiments showed that the SVM classifier outperforms the other classifiers that we experimented with. Table 9.7 shows the precision, recall, and F1 for each of the three categories. It also shows the overall accuracy and the Macro-F measure. The analysis of the features used to train this classifier using chi-squared analysis leads to the same conclusions about the relative importance of the features as described in the previous subsection. However, we noticed that features that are related to subjectivity (*Subjectivity Cues*, *Negation*, *Speculation*) are ranked higher which makes sense in the case of polarity classification.

9.3.7 Impact of Context on Classification Accuracy

To study the impact of using citation context in addition to the citing sentence on classification performance, we ran two polarity classification experiments. In the first experiment, we used the citing sentence only to extract the features that are used to train the classifiers. In the second experiment, we used the gold context sentences (the ones labeled *INCLUDED* by human annotators). Table 9.7 shows the results of the first experiment between rounded parentheses and the results of the second experiments in square brackets. The results show that adding citation context improves the classification accuracy especially in the *subjective* categories, specially in the negative category if we want to be more specific. This supports our

	Negative %	Positive %	Neutral %
Precision	68.7 (66.4) [69.8]	54.9 (52.1) [55.4]	83.6 (82.8) [84.2]
Recall	79.2 (71.1) [81.1]	48.1 (45.6) [46.3]	95.5 (95.1) [95.3]
F1	73.6 (68.7) [75.0]	51.3 (48.6) [50.4]	89.1 (88.5) [89.4]
Accuracy: 81.4 (74.2) [84.2] % Macro-F: 71.3 (62.1) [74.2] %			

Table 9.7: Summary of Citation Polarity Classification Results (10-fold cross validation, SVM: Linear Kernel, c = 1.0). Numbers between rounded parentheses are when only the explicit citing sentence is used (i.e. no context). Numbers in square brackets are when the gold standard context is used.

intuition about polarized citations that authors start their review of the cited work with an objective (neutral) sentence and then follow it with their criticism if they have any. We also reached to similar conclusions with purpose classification, but we are not showing the numbers due to space limitations.

9.3.8 Other Experiments

Can We Do Better?

In this section, we investigate whether it is possible to improve the performance in the two classification tasks. One factor that we believe could have an impact on the result is the size of the training data. To examine this hypothesis, we ran the experiment on different sizes of data. Figure 9.2 shows the learning curve of the two classifiers for different sizes of training data. The accuracy increases as more training data is available so we can expect that with even more data, we can do even better.

Relation Between Citation Purpose/Polarity and Citation Count

The main motivation of this work is our hypothetical assumption that using NLP for analyzing citations gives a clearer picture of the impact of the cited work. As a way to check the validity of this assumption, we study the correlation between the counts of the different purpose and polarity categories. We also study the correlation between these categories and the total number of citations that a paper received since it was published. We use the *training/testing dataset* and the gold annotations for this study.

We compute the Pearson correlation coefficient between the counts of citations from the different categories that a paper received per year since its publication. We found that, on average, the correlation between positive and negative citations is negative (AVG P = -0.194) and that the correlation between the count of positive citations and the total number of citations is higher than the correlation between negative citations and total citations (AVG P = 0.531 for positive vs. AVG P = 0.054 for negative).

Similarly, we noticed that there is a higher positive correlation between *Use* citations and total citations than in the case of both *Substantiation* and *Basis*. This can be explained by the intuition that publications that present new algorithms, tools, or corpora that are used by the research community become more and more popular with time and thus receive more and more citations.

Figure 9.3 shows the result of running our purpose classifier on all the citations to Papineni et al.'s [147] paper about Bleu, an automatic metric for evaluating Machine Translation (MT) systems. The figure shows that this paper receives a high number of *Use* citations. This makes sense for a paper that describes an evaluation metric that has been widely used in the MT area. The figure also shows that in the recent years, this metric started to receive some *Criticizing* citations that resulted in a slight decrease in the number of *Use* citations. Such a temporal analysis of citation purpose and polarity is useful for studying the dynamics of research. It can also be used to detect the emergence or de-emergence of research techniques.

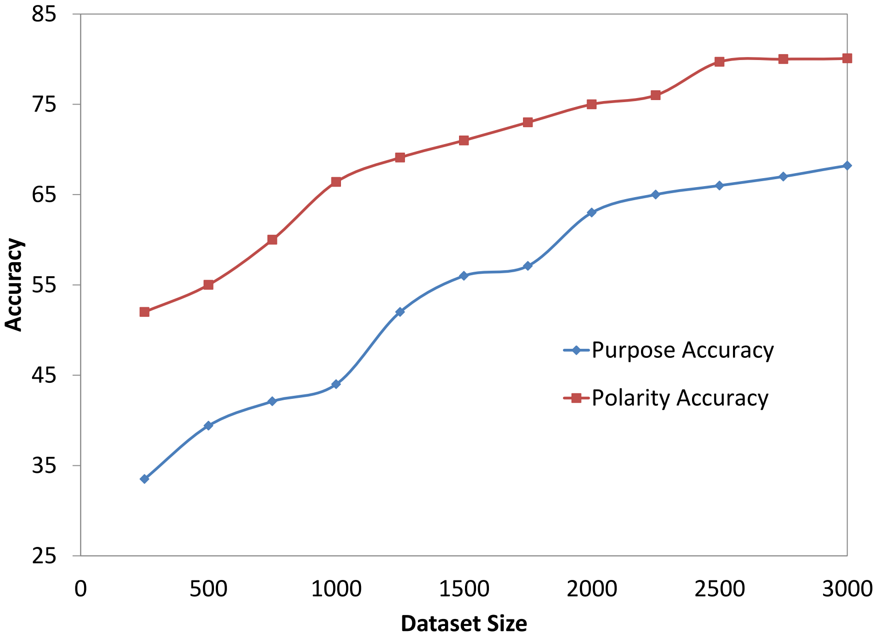


Figure 9.2: The effect of size of the data set size on the classifiers accuracy.

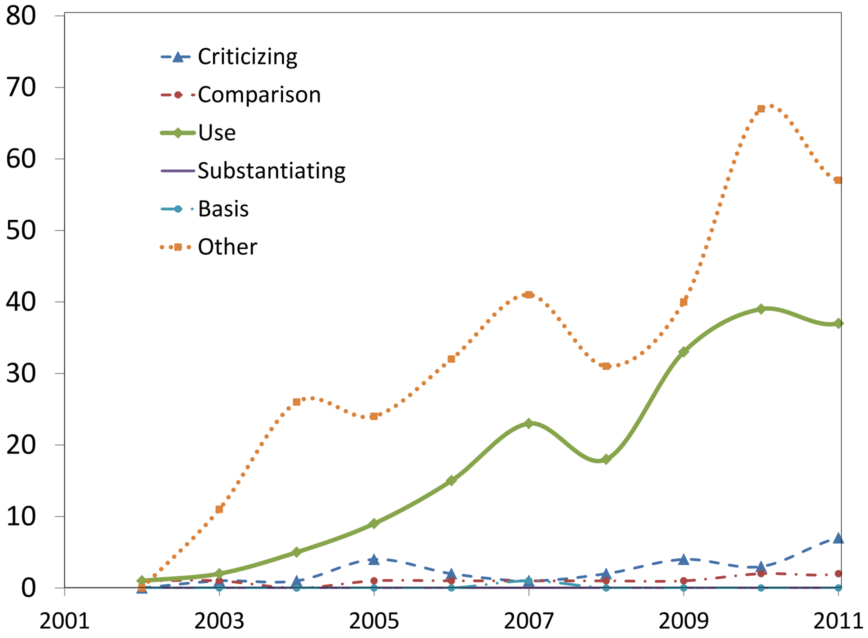


Figure 9.3: Change in the purpose of the citations to Papineni et al. (2002)

9.4 Conclusion

In this chapter, we presented methods for three tasks: citation context identification, citation purpose classification, and citation polarity classification. This work is motivated by the need for more accurate bibliometric measures that evaluates the impact of research both qualitatively and quantitatively. Our experiments showed that we can classify the purpose and polarity of citation with a good accuracy. It also showed that using the citation context improves the classification accuracy and increases the number of polarized citations detected. For future work, we plan to use the output of this research in several applications such as predicting future prominence of publications, studying the dynamics of research, and designing more accurate bibliometric measures.

CHAPTER X

Generating Multi-perspective Summaries of Scientific Articles

The fact that citing sentences cover different aspects of the cited paper and highlight its most important contributions from the viewpoints for expert researchers, motivates the idea of using citing sentences to summarize research. The comparison that Elkiss et al [53] performed between abstracts and citing sentences suggests that a summary generated from citing sentences will be different and probably more concise and informative than the paper abstract or a summary generated from the full text of the paper. For example, Table 10.1 shows the abstract of Resnik (1999) and 5 selected sentences that cite it. We notice that citing sentences contain additional facts that are not in the abstract, not only ones that summarize the paper contributions, but also those that criticize it (the last citing sentence in Table 10.1, for example).

Previous work has explored this research direction. Qazvinian and Radev [152] proposed a method for summarizing scientific articles by building a similarity network of the sentences that cite it, and then applying network analysis techniques to find a set of sentences that covers as much of the paper facts as possible. Qazvinian et al. [155] proposed another summarization method that first extracts a number of important keyphrases from the set of citing sentences, and then finds the best subset

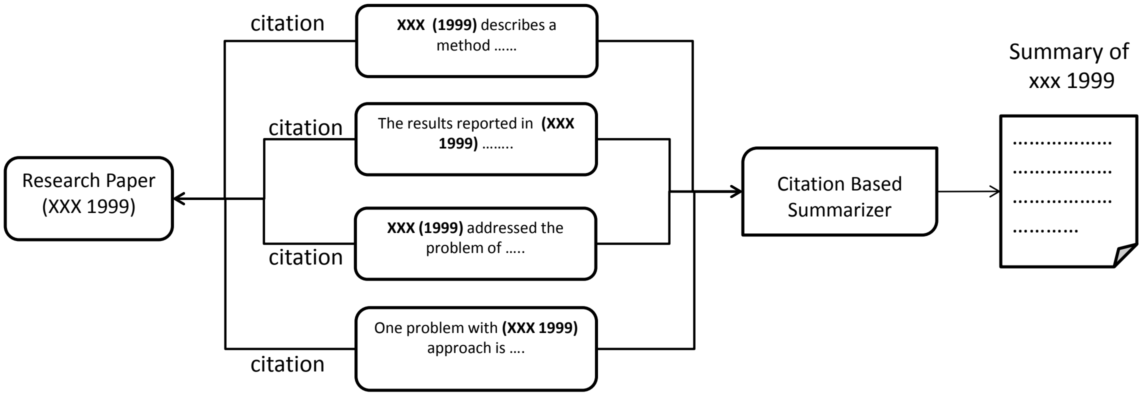

citation
XXX (1999) describes a method ……
citation
The results reported in (XXX 1999) ……..
Research Paper (XXX 1999)
XXX (1999) addressed the problem of …..
citation
One problem with (XXX 1999) approach is ….
citation
Citation Based Summarizer
Summary of xxx 1999
……………..
……………..
……………..
…………

Figure 10.1: We use sentences that contain citations to a given paper to produce a summary of the paper contributions

of sentences that covers as many keyphrases as possible.

Mohammed et al. [131] went beyond single paper summarization. They investigated the usefulness of directly summarizing citation texts in the automatic creation of technical surveys. They generated surveys from a set of Question Answering (QA) and Dependency Parsing (DP) papers, their abstracts, and their citation texts. The evaluation of the generated surveys shows that both citation texts and abstracts have unique survey-worthy information.

These works focused on analyzing the citing sentences and selecting a representative subset that covers the different aspects of the summarized article. In our work, we address the issue of coherence and readability in summaries generated from citing sentences. We add preprocessing and postprocessing steps to the summarization pipeline. In the preprocessing step, we use a supervised classification approach to rule out irrelevant sentences or fragments of sentences that are not suitable for summarization. In the postprocessing step, we improve the summary coherence and readability by reordering the sentences, removing extraneous text (e.g. redundant

Abstract	STRAND (Resnik, 1998) is a language-independent system for automatic discovery of text in parallel translation on the World Wide Web. This paper extends the preliminary STRAND results by adding automatic language identification, scaling up by orders of magnitude, and formally evaluating performance. The most recent end-product is an automatically acquired parallel corpus comprising 2491 English-French document pairs, approximately 1.5 million words per language.
Selected Citing Sentences	Many research ideas have exploited the Web in unsupervised or weakly supervised algorithms for natural language processing (e.g. , Resnik (1999)) Resnik (1999) addressed the issue of language identification for finding Web pages in the languages of interest. In Resnik (1999), the Web is harvested in search of pages that are available in two languages, with the aim of building parallel corpora for any pair of target languages. The STRAND system of (Resnik, 1999), uses structural markup information from the pages, without looking at their content, to attempt to align them. Mining the Web for bilingual text (Resnik, 1999) is not likely to provide sufficient quantities of high quality data.

Table 10.1: Comparison of the abstract and a selected set of sentences that cite Resnik (1999) work

mentions of author names and publication year). Our experiments show that our approach produces better summaries than several baseline summarization systems. This work is based on the work published in [10].

The rest of this chapter is organized as follows. We present the motivation of our approach in Section 10.1. Section 10.2 describes the three stages of our summarization system. The evaluation process, the experimental setup, and the results are presented in Section 10.3. Section 10.4 concludes.

10.1 Motivation

The fluency of citation-based summaries is impeded by several factors. First, many of the citation sentences cite multiple papers besides the target. For example, the following is a citation sentence that appeared in the NLP literature and talked about Resnik's 1999 work.

(1) Grefenstette and Nioche (2000) and Jones and Ghani (2000) use the web to generate corpora for languages where electronic resources are scarce, while ***Resnik(1999)***

describes a method for mining the web for bilingual texts.

The first fragment of this sentence describes different work other than Resnik's. The contribution of Resnik is mentioned in the underlined fragment. Adding the whole sentence to the summary breaks the context and confuses the reader. It would be better to extract the underlined fragment and use it in the summary rather than the whole sentence.

A second factor is associated with the ordering of the sentences included in the summary. For example, the following are two other citation sentences for Resnik 1999.

*(2) Mining the Web for bilingual text (**Resnik, 1999**) is not likely to provide sufficient quantities of high quality data.*

*(3) **Resnik (1999)** addressed the issue of language identification for finding Web pages in the languages of interest.*

If these two sentences are to be included in the summary, the reasonable ordering would be to put the second sentence first.

Thirdly, in some instances of citation sentences, the reference is not a syntactic constituent in the sentence. It is added just to indicate the existence of citation. For example, in sentence (2) above, the reference could be safely removed from the sentence without hurting its grammaticality.

In other instances (e.g. sentence (3) above), the reference is a syntactic constituent of the sentence and removing it makes the sentence ungrammatical. However, in certain cases, the reference could be replaced with a suitable pronoun (i.e. he, she or they). This helps avoid the redundancy of repeating the author(s) name(s) in every sentence.

Finally, a significant number of citation sentences are not suitable for summarization [184] and should be filtered out. The following sentences are two examples.

(4) The two algorithms we employed in our dependency parsing model are the Eisner parsing ***(Eisner, 1996)*** *and Chu-Lius algorithm (Chu and Liu, 1965).*

(5) This type of model has been used by, among others, ***Eisner (1996)****, McDonald et al.*

Sentence (5) appeared in a paper by Nguyen et al (2007). It does not describe any aspect of Eisner's work, rather it informs the reader that Nguyen et al used Esiner's algorithm in their model. There is no value in adding this sentence to the citation summary of Eisner's paper. Likewise, the comprehension of sentence (6) depends on knowing its context (i.e. its surrounding sentences). This sentence alone does not provide any valuable information about the Eisner's paper and should not be added to the summary unless its context is extracted and considered for the summary as well.

In our approach, we address these issues to achieve the goal of improving the fluency of citation-based summaries. Our approach is described in the next section.

10.2 Approach

In this section we describe a system that takes a scientific paper and its set of citing papers as input, and outputs a fluent citation summary of the paper. Our system produces the summaries in three stages. In the first stage, the citation sentences are preprocessed to rule out the unsuitable sentences and the irrelevant fragments of sentences. In the second stage, a number of citation sentences that cover the various aspects of the paper are selected. In the last stage, the selected sentences are post-processed to enhance fluency and cohesiveness of the summary. Figure 10.2

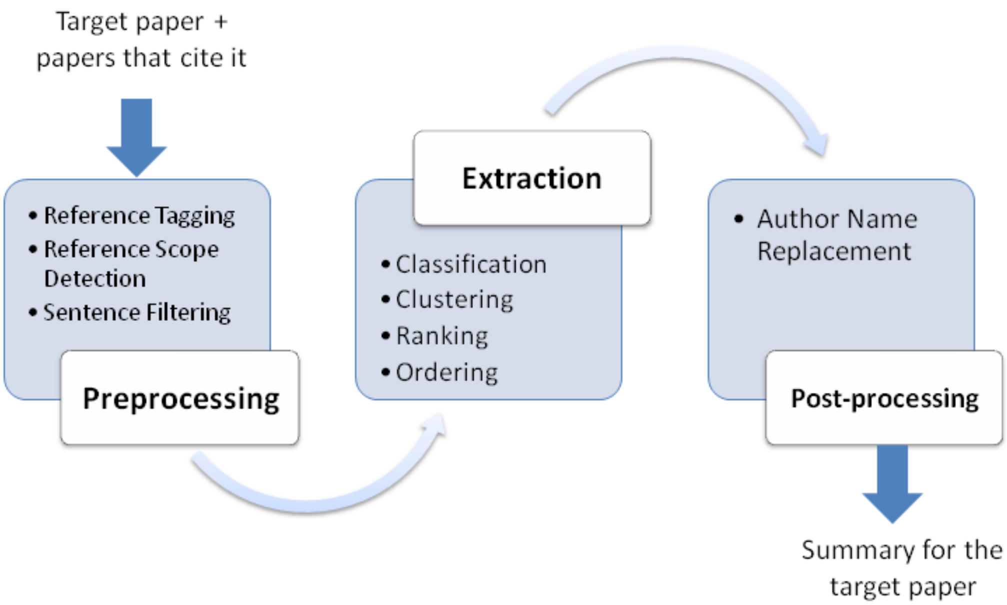


Figure 10.2: Overview of the summarization system

We describe the stages in the following three subsections.

10.2.1 Sentence Preprocessing

The aim of this stage is to determine which pieces of text (sentences or fragments of sentences) should be considered for selection in the next stage and which ones should be excluded. This stage involves three tasks: reference tagging, reference scope detection, and sentence filtering. Reference tagging and reference scope extraction are done as described in Chapter VIII. In the following subsection, we describe the filtering task.

Sentence Filtering

Teufel (2007) reported that a significant number of citation sentences (67% of the sentences in her dataset) are of type *OWN*; i.e. describe the own work of their author not the work of the cited paper. In our dataset (described below in Section 10.3),

Feature	**Description**
Similarity to the target paper	The value of the cosine similarity (using TF-IDF vectors) between the citation sentence and the target paper.
Headlines	The section in which the citation sentence appeared in the citing paper. We recognize 10 groups of section such as *Introduction, Related Work, Approach, etc.*
Relative position	The relative positions of the sentence in the section and the paragraph in which it appears
First person pronouns	This feature takes a value of 1 if the sentence contains a first person pronoun (I, we, our, us, etc.), and 0 otherwise.
Tense of the first verb	A sentence that contains a past tense verb near its beginning is more likely to be describing previous work.
Determiners	Demonstrative Determiners (this, that, these, those, and which) and Alternative Determiners (another, other). The value of this feature is the relative position of the first determiner (if one exists) in the sentence.

Table 10.2: The features used for sentence filtering

almost 45% of the citation sentences were of this type. Sentences of this type are not useful for our summarization task and should not be considered for extraction.

Another type of sentence are the ones that depend on their context. These sentences can be useful if their context (i.e. the relevant surrounding sentences) could be identified and included. In this work, however, we assume that context information is not available.

The task in this step is to detect sentences of these types and rule them out. Formally, we need to classify the citation sentences into two classes: suitable and unsuitable sentences. We use a machine learning technique for this purpose. We extract a number of features from each sentence and train a classification model on these features. The trained model is then used to classify the sentences.

The features that we use in this step and their descriptions are as shown in table10.2

10.2.2 Extraction

In the first stage, the sentences and the fragments of sentences that are not useful for our summarization task are ruled out. The input to this stage is a number of citation sentences that are believed to be suitable candidates for the summary. From these sentences, we need to select a representative subset. The selected sentences should have three main properties:

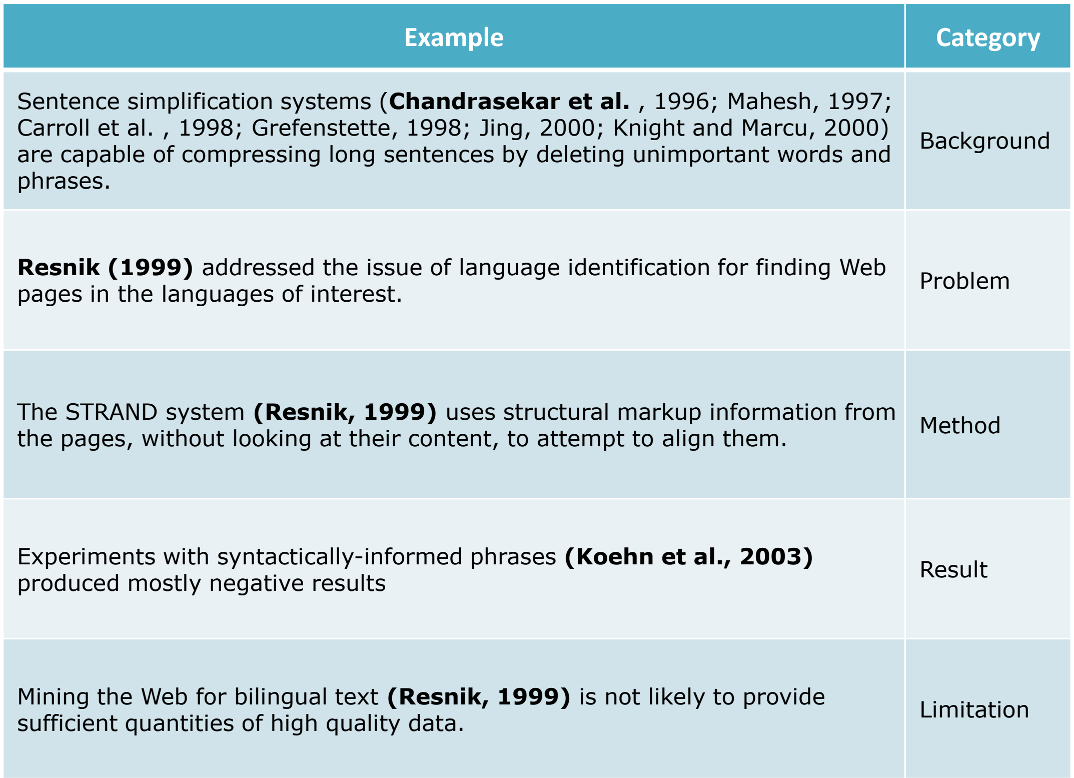
First, they should cover diverse aspects of the paper. For example, one sentence can describe the problem the paper addresses. Another sentence should describe the approach it proposes. A third sentence should discuss the results it reports and so on.

Second, the sentences that cover one aspect of the paper should not contain redundant information. For example, if two sentences talk about the limitations of the target paper, one sentence can mention the computation inefficiency, while the other criticize the assumptions the paper makes.

Third, the sentences should contain the most important information and cover as much facts of the target paper as possible using minimal text.

In this stage, the summary sentences are selected in three steps. In the first step, the sentences are classified into five functional categories: *Background, Problem Statement, Method, Results*, and *Limitations.* This classification task is described in the following subsection.

In the second and the third steps, we follow the work of Qazvinian and Radev (2008). In the second step, we cluster the sentences within each category into clusters of similar sentences. In the third step, we compute the LexRank [55] values for the sentences within each cluster. The summary sentences are selected based on the classification, the clustering, and the LexRank values. The details follow.



Example	Category
Sentence simplification systems (**Chandrasekar et al.** , 1996; Mahesh, 1997; Carroll et al. , 1998; Grefenstette, 1998; Jing, 2000; Knight and Marcu, 2000) are capable of compressing long sentences by deleting unimportant words and phrases.	Background
Resnik (1999) addressed the issue of language identification for finding Web pages in the languages of interest.	Problem
The STRAND system **(Resnik, 1999)** uses structural markup information from the pages, without looking at their content, to attempt to align them.	Method
Experiments with syntactically-informed phrases **(Koehn et al., 2003)** produced mostly negative results	Result
Mining the Web for bilingual text **(Resnik, 1999)** is not likely to provide sufficient quantities of high quality data.	Limitation

Figure 10.3: Example citing sentences of the five citation categories

Classification of Citation Function

We classify the citation sentences into five categories: *Background*, *Problem*, *Method*, *Result*, and *Weakness*. Figure 10.3 shows an example of each category.

A classification model is trained on a number of features extracted from a labeled set of citation sentences. The features we use for this classifier are listed, along with their descriptions, in table 10.3.

Clustering

In the previous step we determined the category of each citation sentence. It is very likely that sentences from the same category contain similar or overlapping information. For example, Sentences (8), (9), and (10) below appear in the set of

Feature	**Description**
Similarity to the sections of the target paper	The sections of the target paper are categorized into five categories: 1) *Introduction, Motivation, Problem Statement.* 2) *Background, Prior Work, Previous Work,* and *Related Work.* 3) *Experiments, Results,* and *Evaluation.* 4) *Discussion, Conclusion,* and *Future work.* 5) All other headlines. The value of this feature is the cosine similarity (using TF-IDF vectors) between the sentence and the text of each of the five section categories.
Headlines	This is the same feature that we used for sentence filtering in Section 10.2.1.
Number of references in the sentence	Sentences that contain multiple references are more likely to be *Background* sentences.
Verbs	We use all the verbs that their lemmatized form appears in at least three sentences that belong to the same category in the training set. Auxiliary verbs are excluded. In our annotated dataset, for example, the verb *propose* appeared in 67 sentences from the *Methodology* category, while the verbs *outperform* and *achieve* appeared in 33 *Result* sentences.

Table 10.3: The features used for sentence classification

citation sentences of Goldwater and Griffiths (2007) paper. These sentences belong to the same category (i.e *Method*). Both Sentences (8) and (9) convey the same information about Goldwater and Griffiths (2007) contribution. Although Sentence (10) belongs to the same category, it describes a different aspect of the paper methodology.

(8) ***Goldwater and Griffiths (2007)*** *proposed an information-theoretic measure known as the Variation of Information (VI)*

(9) ***Goldwater and Griffiths (2007)*** *propose using the Variation of Information (VI) metric*

(10) A fully-Bayesian approach to unsupervised POS tagging has been developed by ***Goldwater and Griffiths (2007)*** *as a viable alternative to the traditional maximum likelihood-based HMM approach.*

The aim of the clustering step is to group the sentences within each category into clusters of similar sentences. To do this we follow the work of [152]. We build a cosine similarity graph out of the sentences of each category. This is an undirected graph in which nodes are sentences and edges represent similarity relations. Each edge is weighted by the value of the cosine similarity between the two sentences the edge connects. Once we have the similarity network constructed, we partition it into clusters using community finding technique. We use the Clauset algorithm [41], a hierarchical agglomerative community finding algorithm. This algorithm runs in linear time.

Ranking

Although the sentences that belong to the same cluster are similar, they are not necessarily equally important. Following [152], we rank the sentences within each cluster by computing their LexRank [55]. Sentences with higher ranks are more important.

Sentence Selection

At this point we have determined, for each sentence, its category, its cluster, and its relative importance. Figure 10.4 illustrates this situation. Sentences are added to the summary in order based on their category, the size of their clusters, their LexRank values. The categories are ordered as *Background, Problem, Method, Results,* then *Limitations*. Clusters within each category are ordered by size (number of sentences in the cluster). The sentences of each cluster are ordered by their LexRank Value.

In the example shown in Figure 10.4 we have three categories. If the desired length of the summary is 3 sentences, the selected sentences will be in order S1, S12, then S18. If the desired length is 5, the selected sentences will be S1, S5, S12, S15, then S18.

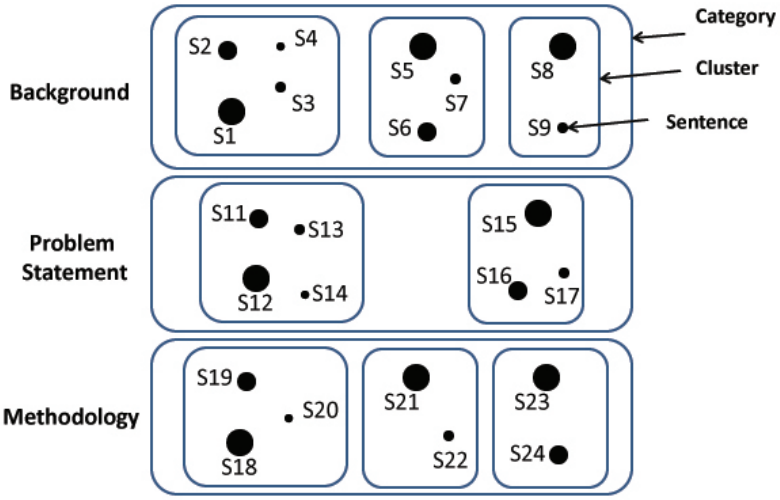


Figure 10.4: Example illustrating sentence selection

10.2.3 Postprocessing

In this stage, we refine the sentences that we extracted in the previous stage. Each citation sentence will have the target reference (the author's names and the publication year) mentioned at least once. The reference could be either syntactically and semantically part of the sentence (e.g. Sentence (3) above) or not (e.g. Sentence (2)). The aim of this refinement step is to avoid the redundancy of repeating the author's names and the publication year in every sentence. We keep the author name and the publication year only in the first sentence of the summary. In the following sentences, we either replace the reference with a suitable personal pronoun or remove it at all. The reference is replaced with a pronoun if it is part of the sentence and this replacement does not make the sentence non-grammatical. The reference is removed if it is not part of the sentence. If the sentence contains references for other papers, they are either removed (if possible) or kept because replacing them with pronouns will confuse the reader.

To determine whether a reference is part of the sentence or not, we again use a

Feature	**Description**
Part-of-speech (POS) tag	we consider the POS tags of the reference, the word before, and the word after. Before passing the sentence to the POS tagger, all the references in the sentence are replaced by placeholders.
Style of the reference	It is common practice in writing scientific papers to put the whole citation between parenthesis when the authors are not a constitutive part of the enclosing sentence, and to enclose just the year between parenthesis when the author's name is a syntactic constituent in the sentence.
Relative position of the reference	this feature takes one of three values: *first*, *last*, and *inside*.
Grammaticality	gramaticality of the sentence if the reference is removed/replaced. Again, we use the Link Grammar parser [174] to check the grammaticality

Table 10.4: The features used for sentence classification

machine learning approach. We train a model on a set of annotated sentences. The features used in this step are listed in Table 10.4. The trained model is then used to classify the references that appear in a sentence into three classes: *keep*, *remove*, *replace*. If a reference is to be replaced, and the paper has one author, we use he/she (we do not know if the author is male or female). If the paper has two or more authors, we use *they*.

10.3 Evaluation

We provide three levels of evaluation. First, we evaluate each of the subcomponents in our system separately. Then we evaluate the summaries that our system generate in terms of extraction quality. Finally, we evaluate the coherence of the summaries.

10.3.1 Data

We use the ACL Anthology Network (AAN) [160] in our evaluation. AAN is a collection of more than 16000 papers from the Computational Linguistics journal, and the proceedings of the ACL conferences and workshops. AAN provides all citation information from within the network including the citation network, the citation sentences, and the citation context for each paper.

We used 55 papers from AAN as our data. The papers have variable number of citation sentences ranging from 15 to 348. The total number of citation sentences in the dataset is 4335 sentences. We split the data randomly into two different sets; one for evaluating the subcomponents of the system, and the other for evaluating the extraction quality and the fluency of the generated summaries. The first set (*dataset1*, henceforth) contained 2284 sentences coming from 25 papers. We asked humans to provide three annotations for each sentence in this set: 1) label the sentence as *suitable* or *Unsuitable*, 2) label each sentence as *Background, Problem, Method, Results* or *Limitations*, and 3) for each reference in the sentence, determine whether it could be *replaced* with a pronoun, *removed*, or should be *kept*. Each sentence was given to 3 different annotators. We used the majority vote labels.

We use Kappa coefficient K (Siegel and Castellan 1988) to measure the inter-annotator agreement. The agreement among the three annotators on distinguishing *unsuitable* sentences from the other five categories is 0.85. On Landis and Kochs (1977) scale, this value indicates an *almost perfect* agreement. The agreement on classifying the sentences into the five functional categories is 0.67. On Landis and Kochs (1977) scale, this value indicates *substantial agreement.*

The second set (*dataset2*, henceforth) contained 30 papers (2051 sentences). We asked humans to generate a fluent summary for each paper in the set using its citation sentences as the source text. We asked them to fix the length of the summaries to 5 sentences. Each paper was assigned to two humans to summaries.

-	Bkgrnd	Prob	Method	Results	Limit.
Precision	64.62%	32.64%	88.66%	76.05%	33.53%
Recall	72.47%	64.75%	75.03%	82.29%	59.36%
F1	68.32%	43.40%	81.27%	79.04%	42.85%

Table 10.5: Precision and recall results achieved by our citation sentence classifier

10.3.2 Component Evaluation

Sentence Filtering Evaluation

We used Support Vector Machines (SVM) as our classifier. We performed 10-fold cross validation on the labeled sentences (*unsuitable* vs *all other categories*) in *dataset1*. Our classifier achieved 80.3% accuracy.

Sentence Classification Evaluation

We used SVM in this step as well. We also performed 10-fold cross validation on the labeled sentences (the five functional categories). This classifier achieved 70.1% accuracy. The precision and recall for each category are given in Table10.5

Author Name Replacement Evaluation

The classifier used in this task is also SVM. We performed 10-fold cross validation of the labeled sentences of *dataset1*. Our classifer achieved 77.41% accuracy.

10.3.3 Extraction Evaluation

To evaluate the extraction quality, we use *dataset2* (that has never been used for training or tuning any of the system subcomponents). We use our system to generate summaries for each of the 30 papers in *dataset2*. We also generate summaries for the papers using a number of baseline systems. All the generated summaries were 5 sentences long. We use the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) based on the longest common substrings (ROUGE-L) as our evaluation metric.

Baselines

We evaluate the extraction quality of our system (FL) against 8 different baselines. In the first baseline, the sentences are selected randomly from the set of citation sentences and added to the summary. The second baseline is the MEAD summarizer (Radev et al. 2004) with all its settings set to default. The third baseline is LexRank [55] run on the entire set of citation sentences of the target paper. The forth baseline is Qzvinian and Radev (2008) citation-based summarizer (QR08) in which the citation sentences are first clustered then the sentences within each cluster are ranked using LexRank. The remaining baselines are variations of our system produced by removing one component from the pipeline at a time. In one variation (FL-1), we remove the sentence filtering component. In another variation (FL-2), we remove the sentence classification component; so, all the sentences are assumed coming from one category in the subsequent steps. In a third variation (FL-3), the clustering component is removed. To make the comparison to those baselines fair, we remove the author name replacement component from our system and all its variations.

Results

Table 10.6 shows the average ROUGE-L scores (with 95% confidence interval) for the summaries of the 30 papers in *dataset2* generated using our system and the different baselines. The two human summaries were used as models for comparison. Statistical significance was tested using a 2-tailed paired t-test. The results are statistically significant at the 0.05 level.

The results show that our approach outperforms all the other baseline techniques. It achieves higher ROUGE-L score for most of the papers in our testing set. Compar-

ROUGE	Human	Random	MEAD	LexRank	QR08	FL-1	FL-2	FL-3	FL
ROUGE-L	0.733	0.398	0.410	0.408	0.435	0.475	0.511	0.525	**0.539**

Table 10.6: Extraction Evaluation

ing the score of FL-1 to the score of FL shows that sentence filtering has a significant impact on the results. It also shows that the classification and clustering components both improve the results.

10.3.4 Coherence and Readability Evaluation

We asked human judges (not including ourselves) to rate the coherence of a number of summaries for each of *dataset2* papers. For each paper we evaluated 3 summaries. The summary that our system produced, the human summary, and a summary produced by Qazvinian and Radev (2008) summarizer (the second best baseline - after our system and its variations - in terms of extraction quality as shown in the pervious subsection.) The summaries were randomized and given to the judges without telling them how each summary was produced. The judges were not given access to the source text. They were asked to use a five point-scale to rate how coherent and readable the summaries are, where 1 means that the summary is totally incoherent and needs significant modifications to improve its readability, and 5 means that the summary is coherent and no modifications needed to improve its readability. We gave each summary to 5 different judges and took the average of their ratings for each summary. Table 10.7 shows the number of summaries in each rating range. The results show that our approach significantly improves the coherence of citation-based summarization. Table 10.8 shows two sample summaries (each 5 sentences long) for Voutilainen (1995) paper in which he describes his rule based tagger. One summary was produced using our system and the other was produced using Qzvinian and Radev (2008) system.

Average Coherence Rating	Number of summaries		
	Human	FL	QV08
1≤ coherence <2	0	7	17
2≤ coherence <3	3	4	3
3≤ coherence <4	17	10	1
4≤ coherence ≤5	10	1	0

Table 10.7: Coherence Evaluation

Produced using our system
There has been a large number of studies in tagging and morphological disambiguation using various techniques such as statistical techniques, e.g. constraint-based techniques and transformation-based techniques. A thorough removal of ambiguity requires a syntactic process. A rule-based tagger described in Voutilainen (1995) was equipped with a set of guessing rules that had been hand-crafted using knowledge of English morphology and intuitions. The precision of rule-based taggers may exceed that of the probabilistic ones. The construction of a linguistic rule-based tagger, however, has been considered a difficult and time-consuming task.
Produced using Qazvinian and Radev (2008) system
Another approach is the rule-based or constraint-based approach, recently most prominently exemplified by the Constraint Grammar work (Karlsson et al. , 1995; Voutilainen, 1995b; Voutilainen et al. , 1992; Voutilainen and Tapanainen, 1993), where a large number of hand-crafted linguistic constraints are used to eliminate impossible tags or morphological parses for a given word in a given context. Some systems even perform the POS tagging as part of a syntactic analysis process (Voutilainen, 1995). A rule-based tagger described in (Voutilainen, 1995) is equipped with a set of guessing rules which has been hand-crafted using knowledge of English morphology and intuition. Older versions of EngCG (using about 1,150 constraints) are reported (butilainen et al. 1992; Voutilainen and HeikkiUi 1994; Tapanainen and Voutilainen 1994; Voutilainen 1995) to assign a correct analysis to about 99.7% of all words while each word in the output retains 1.04-1.09 alternative analyses on an average, i.e. some of the ambiguities remait unresolved. We evaluate the resulting disambiguated text by a number of metrics defined as follows (Voutilainen, 1995a).

Table 10.8: Sample Output

10.4 Conclusion

In this paper, we presented a new method for citation-based summarization of scientific papers that produces fluent summaries. Our approach involves three stages. The first stage preprocesses the set of citation sentences to determine the irrelevant sentences or fragments of sentences and rule them out. In the second stage, the remaining sentences are classified into 5 functional categories. The sentences within each category are then grouped into clusters of similar sentences. Then, the sentences are ranked within their clusters using LexRank. Sentences are added to the summary in order based on their category, the size of their cluster, and their ranks. In the last stage, each sentence is refined to avoid the redundancy caused by repeating the authors' names. The results of our experiments confirmed that our system outperforms other baseline systems.

CHAPTER XI

Other Applications

In this chapter, we hypothesize several uses of citing sentences such as analyzing the trends of research, understanding the impact of research and how this impact changes over time, summarizing the contributions of a researcher, summarizing the discoveries in a certain research field, and providing high quality data for Natural Language Processing tasks. In the rest of this paper we present some of these ideas and provide examples to demonstrate their applicability. Some of these ideas have been explored in previous work, but we believe that they still need further exploration. However, most of the ideas are novel to our knowledge. We present our ideas in the following sections.

11.1 Predicting Future Prominence of Papers

In any scientific community, certain papers may end up being much more prominent than others. If we look at all the citations received by the papers in the ACL Anthology till 2012, the top 10% of the papers receive 70% of the total number of citations. We investigate the possibility of detecting prominent papers within a year or two of the publication of the paper.

This work is most related to [210], which try to predict the response of a scientific community to an article. They look at the problem of predicting whether a paper will receive any citations within the first 3 years given data at publication time. Our work can be seen as complementary to this work. We want to study how accurately can we predict the prominence of a paper in the horizon of 10 years given the first few years of data.

There has been interest in the task of citation prediction in the data mining community. One of the tasks of the popular KDD Cup competition that was held in year 2003[64] asked the participants to predict changes in the number of citations received by well-cited papers over time. The citation network of 30,119 papers and the full text of each paper was provided to participants. All participating systems used network based features, time series based features, keyword diffusion features, and metadata features (such as author prestige, etc.). None of the systems used NLP-based features of citation relations. We believe that applying our analysis of citation text can lead to a more accurate citation prediction.

Yan et al.[207] use a variety of features including topic models, diversity and recency to predict the exact future citation counts of papers and report R^2 values of .75 and .79 for 5-year and 10-year predictions respectively. In later work [206], they do more comprehensive analysis and report higher R^2 values of 0.87 and 0.92 for forecast periods of 5 and 10 years respectively.

Reference lag	**I-set**	**S-set**	**R-set**
0	341	3924	2730
1	194	4341	2460
2	78	4771	2147
3	49	4932	2015
4	15	5142	1839
5	9	5280	1707

Table 11.1: Distribution across I, S and R sets for 5 different values of reference lag

Given the data for a paper after one or two years of its publication, we would like to predict the prominence of the paper in a large horizon of the next 10 years. One way to look at it is as a regression problem where we try to predict the number of citations the paper will receive. However, the exact counts of citations to papers are dependent on the rate of new publications and is not comparable from one year to the other. Therefore, we formulate the problem in such a way that we compare papers that are published in the same year. The intuition is that if some of these papers get many more citations than other papers published in the same year, then they are more prominent. We first introduce a few terms.

We define *reference year* as the time till which our system is allowed to look at the data. The *forecast year* is defined as the year for which we have to make a prediction for the paper. The idea is that our system takes the data between the publication year and the reference year and makes a prediction about whether the paper showed prominence between the reference year and the forecast year. We compute the percentile rank of the paper at the forecast year with respect to all the papers published in the same year based on the cumulative citations accumulated by paper since it was published.

We studied two different problem formulations of predicting the prominence of papers in the future. In the first formulation, our goal is to detect the papers that are in the top 10 percentile at the forecast year amongst the papers published in any given year. We refer to this set as the P-set. We refer to the rest of the papers as the R-set.

In the second formulation we divide the papers into three sets as follows. Set-I: represents the set of papers for which the percentile rank increased by more than some upper threshold T_u between the reference year and forecast year. Set-N: represents the set of papers for which the percentile rank did not increase more than some lower threshold T_l. Set-R: represents the rest of the papers. Set-I represents the papers that could not be distinguished as prominent papers within at reference year, but became prominent by the forecast year.

Reference lag	**Accuracy**	**F-score, label I**	**F-score, label S**	**F-score, label R**
0	.80	.45	.86	.72
1	.71	.49	.80	.53
2	.75	.26	.83	.51
3	.79	.20	.86	.57

Table 11.2: Accuracy and per class F-scores for 3-Class classification set up.

For experimentation, we generate data from the ACL Anthology Network [161]. Our publication years vary from 1980 to 2002. We fix the forecast year for a paper to be 10 years after the publication year of that paper. For the 3-class formulation T_u is set to 0.4 and T_l is set to 0. The data that we generated comprised 6996 papers.

The task now is to determine the set that the paper will belong to at the forecast year. We adopt a supervised classification approach.

We use several features derived from the different entities associated with the paper along with linguistic features derived from the text of the paper. We divide the features into several groups:

- **Author based features**: such as the number of publications of the author, the H-index of the author and the citation count. If the paper has multiple authors, we use the maximum, minimum and average of these values as features.

- **Citation based features**: count of cited papers, the pagerank of cited papers etc. Similarly, we derive features from the papers that cite the current paper in the reference period, e.g. the average H-index of the authors of papers citing the current papers.

- **Citation text features**: for the papers citing this paper, we look at the sentences used while citing this paper and compute features using the citation sentiment and citation purpose of these sentences.

- **Heterogeneous network features**: we use a network that combines authors, papers, venues, institutions and terms into a single network. For each of the entities in the network, we compute its pagerank and the slope of the change in the pagerank over the last 5 years.

We trained a logistic regression classifier for each of the problem formulations. The results showed that we can predict the accuracy of a paper being in Set-P with more than 80% F-score by looking at just two years of data. Our results showed that a paper accumulates enough evidence within the first two years to be able to predict its forecast for the next 10 years.

The results for the 3-class classification show that it is a much more difficult task to predict the accuracy of a paper being in Set-I. Additionally, the problem becomes more difficult as we increase the reference lag. Table 11.2 summarizes the results of our classification. The maximum F-score we achieve for Set-I in this setup is 0.49, after a reference lag of 1 year. The F-score decreases consistently as we increase the reference lag. This is consistent with the results of the 2-class classification. Both of these results show that the percentile ranks of papers stabilize after 1-2 years of publication, and do not change after that. This is also reflected in the decreasing size of Set-I as we increase the reference year in Table 11.1.

11.2 Temporal Analysis of Citations

The interest in studying citations stems from the fact that bibliometric measures are commonly used to estimate the impact of a researcher's work [33, 117]. Several

Comparison	Contrast/Comparison in Results, Method, or Goals
Basis	Author uses cited work as basis or starting point
Use	Author uses tools, algorithms, data, or definitions
Description	Neutral description of cited work
Weakness	Limitation or weakness of cited work

Table 11.3: Annotation scheme for citation purpose

previous studies have performed temporal analysis of citation links [20, 124, 164] to see how the impact of research and the relations between research topics evolve overtime. These studies focused on observing how the number of incoming citations to a given article or a set of related articles change over time. They overlooked the fact that the number of incoming citations is often not the only factor that changes with time. We believe that analyzing the text of citing sentences allows researchers to observe the change in other dimensions such as the purpose of citation, the polarity of citations, and the research trends. The following subsections discuss some of these dimensions.

11.2.1 Temporal Analysis of Citation Purpose

Teufel et al. [184] have shown that the purpose of a citation can be determined by analyzing the text of citing sentences. We hypothesize that performing a temporal analysis of the purpose for citing a paper gives a better picture about its impact. As a proof of concept, we annotated all the citing sentences in the ACL Anthology Network (AAN) that cite the top 10 cited papers from the 1980's with *citation purpose* labels. The labels we used for annotation are based on Teufel et al. [184] annotation scheme and are described in Table 11.3. We counted the number of times the paper was cited for each *purpose* in each year since its publication date. This analysis revealed interesting facts about the impact of each paper. We will discuss our observations in Section 11.2.3. Figure 11.1 shows the change in the ratio of each

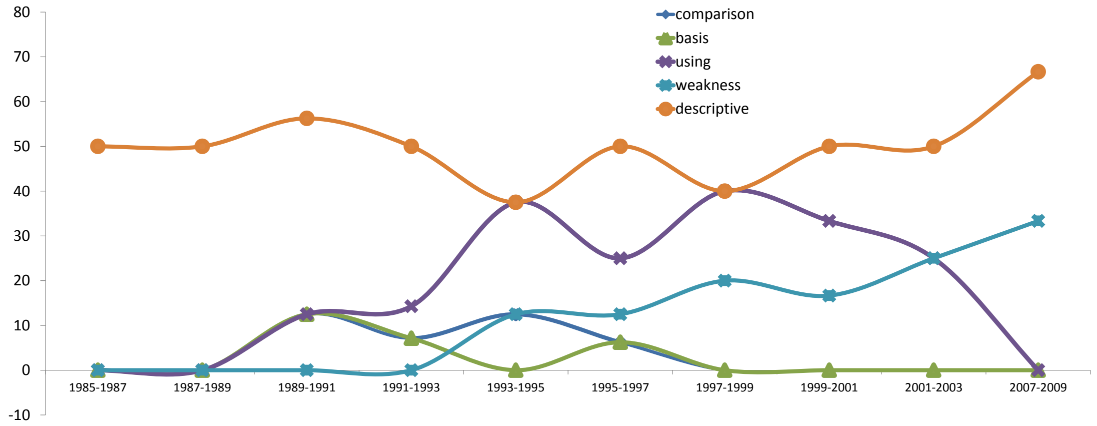


Figure 11.1: Change in the citation purpose of Shieber (1985) paper

purpose with time for Shieber's (1985) work on parsing.

11.2.2 Temporal Analysis of Citation Polarity

The bibliometric measures that are used to estimate the impact of research are often computed based on the number of citations it received. This number is taken as a proxy for the relevance and the quality of the published work. It, however, ignores the fact that citations do not necessarily always represent positive feedback. Many of the citations that a publication receives are neutral citations, and citations that represent negative criticism are not uncommon. To validate this intuition, we annotated 2000 citing sentences from AAN for citation polarity. We found that only 30% of citations are positive, 4.3% are negative, and the rest are neutral. In another published study, Athar [23] annotated 8736 citations from AAN with their polarity and found that only 10% of citations are positive, 3% are negative and the rest were all neutral. We believe that considering the polarity of citations when conducting temporal analysis of citations gives more insight about how the way a published work is perceived by the research community over time. As a proof of concept, we annotated the polarity of citing sentences for the top 10 cited papers in AAN that were published in the 1980's. We split the year range of citations into slots of 2 years

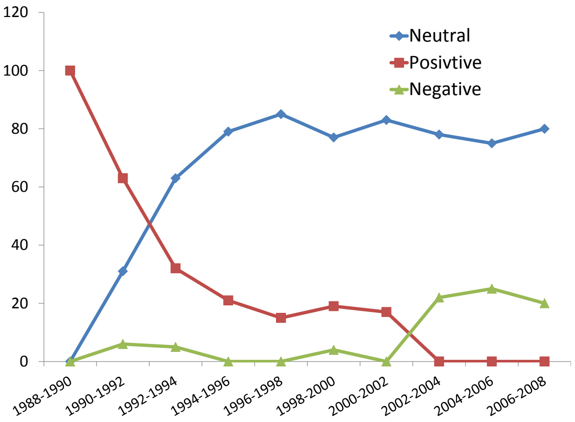


Figure 11.2: Change in the polarity of the sentences citing Church (1988) paper

and counted the number of positive, negative, and neutral citations that each paper received during that time slot. We observed how the ratios of each category changed overtime. Figure 11.2 shows the result of this analysis when applied to the work of Kenneth Church [40] on part-of-speech tagging.

11.2.3 Predict Emergence of New Techniques or Decline of Impact of Old Techniques.

The ideas discussed in Sections 11.2.1 and 11.2.2 and the results illustrated in Figures 11.1 and 11.2 suggest that studying the change in citation purpose and citation polarity allow us to predict the emergence of new techniques or the decline in impact of old techniques. For example, the analysis illustrated in Figure 11.2 shows that the work of Kenneth Church (1988) on part-of-speech tagging received significant positive feedback during the 1990s and until early 2000s before it started to receive more negative feedback. This probably can be explained by the emergence of better statistical models for part-of-speech (POS) tagging (e.g. Conditional Random Fields [108]) that outperformed Church's approach. However, as indicated by the neutral citation curve, Church's work continued to be cited as a classical pioneering research on the POS tagging task, but not as the state-of-the-art approach. A similar analysis can be applied to the change in citation purpose of Shieber (1985) as illustrated in Figure 11.1

11.2.4 Study the Dynamics of Research

In recent research, Gupta and Manning [70] conducted a study that aims at understanding the dynamics of research in the computational linguistics area. They analyzed the abstracts of the papers included in the ACL Anthology Reference Corpus. They extracted the contributions, the domain of application, and the techniques and tools used in each paper. They combined this information with pre-calculated article-to-community assignments to study the influence of a community on others in terms of techniques borrowed and the maturing of some communities to solve problems from other domains. We hypothesize that conducting such an analysis using the citing sentences of papers instead of (or in combination with) abstracts leads to a more accurate picture of research dynamics and the interaction between different research communities. There are several intuitive observations that support this hypothesis.

First, previous research [53] has shown that the sentences that cite a paper are more focused and more concise than the paper abstract, and that they consistently contain additional information that does not appear in abstracts. This means that additional characteristics of a paper can be extracted from citing sentences that cannot be extracted from abstracts. To verify this, we compared abstracts vs citations (within AAN) in terms of the number of occurrences of the *trigger words* that Gupta and Manning [70] deemed to be indicative of paper characteristics (Table 11.4). The numbers clearly show that the trigger words appear more frequently in the set of citing sentences of papers than they do in the paper abstracts. We also found that many papers that none of the *trigger words* appeared in their abstracts, while

	apply	propose	extend	system
Abstracts	1368	2856	425	5065
Citing Sentences	**2534**	**3902**	**917**	**6633**

Table 11.4: Comparison of trigger word occurrences in abstracts vs citing sentences.

they do appear in their citing sentences. This suggests that more paper properties (contributions, techniques used, etc.) could be extracted from citations than from abstracts.

Second, while the contributions included in an abstract are the claims of the paper author(s), the contributions highlighted in citing sentences are collectively deemed to be important by peer researchers. This means that the contributions extracted from citations are more important from the viewpoint of the community and are likely to reflect research trends more accurately.

We performed another simple experiment that demonstrates the use of citing sentences to track the changes in the focus of research. We split the set of citing sentences in AAN into three subsets: the set of citing sentences that cite papers from 1980s, the set of citing sentences that cite papers from 1990s, and the set of citing sentences that cite papers from 2000s. We counted the frequencies of words in each of the three sets. Then, we ranked the words in each set by the decreasing order of their frequencies. We selected a number of keywords and compared their ranks in the three year ranges. Some of these keywords are listed in Table 11.5. This analysis shows, for example, that there was more focus on ”grammar” in the computational linguistics research in the 1980s then this focus declined with time as indicated by the lower rank of the keyword ”grammar” in the 1990s and 2000s. Similarly, rule based methods were popular in the 1980s and 1990s but their popularity declined significantly in the 2000s.

	Rank		
word	1980s	1990s	2000s
grammar	22	71	123
model	75	72	26
rules	77	89	148
statistical	-	69	74
syntax	257	1018	683
summarization	-	880	359

Table 11.5: Ranks of selected keywords in citing sentences to papers published in 80s, 90s and 2000s

11.3 Controversy Identification

Some arguments and claims made by researchers may get disputed by other researchers. The following are examples of citing sentences that dispute previous work.

(1) Even though prior work (Teufel et al., 2006) argues that citation text is unsuitable for summarization, we show that in the framework of multi-document survey creation, citation texts can play a crucial role.

(2) Mining the Web for bilingual text (Resnik, 1999) is not likely to provide sufficient quantities of high quality data.

In many cases, it is useful to know which arguments were confirmed and accepted by the research community and which ones where disputed or even rejected. We believe that analyzing citation text helps identify these contrasting views automatically.

11.4 Comparison of Different Techniques

Citing sentences that compare different techniques or compare the techniques proposed by the author to previous work are common. The following sentences are examples of such comparisons.

In (Zollmann et al., 2008), an interesting comparison between phrase-based, hierarchical and syntax-augmented models is carried out, concluding that hierarchical and *syntax-based models slightly outperform phrase-based models under large data conditions and for sufficiently non-monotonic language pairs.*

(4) Brill's results demonstrate that this approach can outperform the Hidden Markov Model approaches that are frequently used for part-of-speech tagging (Jelinek, 1985; Church, 1988; DeRose, 1988; Cutting et al. , 1992; Weischedel et al. , 1993, as well as showing promise for other applications.

(5) Our highest scores of 90.8% LP and 90.5% LR outperform the scores of the best previously published parser by Charniak (2000) who obtains 90.1% for both LP and LR.

Extracting such comparisons from citations can be of a great benefit to researchers. It will allow them to quickly determine which technique works better for their tasks. To verify that citation text could be a good source for extracting comparisons, we created a list of words and phrases that are usually used to express comparisons and counted their frequency in AAN citing sentences. We found, for example, that the word *compare* (at its variations) appears in about 4000 different sentences, and that the words *outperform* and *contrast* each appears in about 1000 citing sentences.

11.5 Ontology Creation

It is useful for researchers to know which tasks and research problems are important, and what techniques and tools are usually used with them. Citation text is a good source of such information. For example, sentence (6) below shows three different techniques (underlined) that were used to extend tools and resources that were created for English so that they work for other languages. For another example, sentence (7) shows different tasks in which re-ranking has been successfully applied. These relations can be easily extracted from citing sentences and can be possibly used to build an ontology of tasks, methods, tools, and the relations between them.

(6) Another strain of research has sought to exploit resources and tools in some languages (especially English) to construct similar resources and tools for other languages, through heuristic projection (Yarowsky and Ngai, 2001; Xi and Hwa, 2005) or constraints in learning (Burkett and Klein, 2008; Smith and Eisner, 2009; Das and Petrov, 2011; McDonald et al., 2011) or inference (Smith and Smith, 2004).

(7) (Re)rankers have been successfully applied to numerous NLP tasks, such as parse selection (Osborne and Baldridge, 2004; Toutanova et al., 2004), parse reranking (Collins and Duffy, 2002; Charniak and Johnson, 2005), question-answering (Ravichandran et al., 2003).

11.6 Paraphrase Extraction

It is common that multiple citing sentences highlight the same facts about a cited paper. Since these sentences were written by different authors, they often use different wording to describe the cited paper facts. This motivates the idea of using citing sentences to create datasets for paraphrase extraction. For example, sentences (8) and (9) below both cite (Turney, 2002) and highlight the same aspect of Turney's work using slightly different wordings. Therefore, sentences (8) and (9) can be considered paraphrases of each other.

(8) In (Turney, 2002), an unsupervised learning algorithm was proposed to classify reviews as recommended or not recommended by averaging sentiment annotation of phrases in reviews that contain adjectives or adverbs.

(9) For example, Turney (2002) proposes a method to classify reviews as recommended/not recommended, based on the average semantic orientation of the review.

The paraphrase annotation of citing sentences consists of manually labeling which sentence consists of what facts. Then, if two citing sentences consist of the same set of facts, they are labeled as paraphrases of each other. For example, if a paper has 50 sentences citing it, this gives us a paraphrasing dataset that consists of 50*49 = 2,450 pairs. As a proof of concept, we annotated 25 papers from AAN using the annotation method described above. This data set consisted of 33,683 sentence pairs of which 8,704 are paraphrases.

The idea of using citing sentences to create datasets for paraphrase extraction was initially suggested by Nakov et al. [141] who proposed an algorithm that extracts paraphrases from citing sentences using rules based on automatic named entity annotation and the dependency paths between them.

11.7 Scientific Article Classification

Automatic classification of scientific articles is one of the important tasks for creating publication databases. A variety of machine learning algorithms have been proposed for this task. Many of these methods perform the classification based on the title, the abstract, or the full text of the article. Some other methods used citation links in addition to content to make classification decisions. Cao and Gao [37] proposed a two-phase classification system. The system first applies a content-based statistical classification method which is similar to general text classification. In the second phase, the system uses an iterative method to update the labels of classified instances using citation links. A similar approach is also proposed by Zhang et al. [213]. These approaches use citation links only to improve classification decisions that were made based on content. We hypothesize that using the text of citing sentences in addition to citation structure and content leads to more accurate classification than using the content and citation links only.

11.8 Terminology Translation

Citing sentences can also be used to improve machine translation systems by using citing sentences from different languages to build parallel corpus of terms and their translations. This can be done by identifying articles written in different languages that cite a common target paper, then extracting the citing sentences from each paper. Word alignment techniques can then be applied to the text surrounding the reference to the common target paper. The aligned words from each source can then be extracted and used as translations of the same term. Sentences (10) and (11) below illustrate how the application of this proposed method can identify that the underlined terms in sentence 10 (Spanish) and sentence 11 (English) are translations of each other.

(10) Spanish: Se comprob que la agrupacin por bloques ofreca mejores resultados que, la introduccin de vocabulario (Hearst, 1997) o ***las cadenas lxicas*** *(Hearst, 1994) y, por tanto, es la que se ha utilizado en la segunda fase del algoritmo.*

(11) English: This can be done either by analyzing the number of overlapping ***lexical chains*** *(Hearst, 1994) or by building a short-range and long-range language model (Beeferman et al. , 1999).*

11.9 Other Uses of Citing Sentences

Nakov et al. [141] proposed several other possible uses of citing sentences. First, they suggested using them as a source for unannotated comparable corpora. Such comparable corpora can be used in several applications such as paraphrase extracted as we showed earlier. They also noticed that the scientific literature is rife with abbreviations and synonyms, and hence, citing sentences referring to the same article may allow synonyms to be identified and recorded. They also proposed using citing sentences to build a model of the different ways used to express a relationship between two entities. They hypothesized that this model can help improve both relation extraction and named entity recognition systems. Finally, they proposed improving the indexing and ranking of publications by considering, in addition to the content of the publication, the text of citing sentences that cite it and their contexts.

11.10 Conclusion

We have motivated and discussed different uses of citing sentences, the text surrounding citations. We showed that citing sentences can be used to analyze the dynamics of research and observe how it trends. We also gave examples on how analyzing the text of citing sentences can give a better understanding of the impact of a researcher's work and how this impact changes over time. In addition, we presented several applications that can benefit from citing sentences. These include scientific literature summarization, identification of controversial arguments, and relation extraction between techniques, tools and tasks. We also showed how citing sentences can provide high-quality data for NLP tasks such as information extraction, paraphrase extraction, and machine translation.

Much work still needs be done before citing sentences can be put to full use. The ideas that we proposed and motivated in this paper opens several new dimensions for research in different directions. We presented some simple experiments as proof of concepts to support our hypotheses. In future work, we plan to thoroughly explore and publish the proposed ideas.

Part III. Conclusion and Future Work

CHAPTER XII

Conclusion

Language is the medium through which humans express their thoughts and communicate them to others. People express happiness in different words than sadness and use different expressions to show agreement than the expressions they use to show disagreement. Psycholinguistic and Sociolinguistics studies have shown that the language used by humans is affected by cognitive, psychological, and social factors. In this thesis, we attempted to explore this relationship between language on one side and thoughts, feelings, beliefs, and perspectives on the other side. In particular, we studied how linguistic analysis techniques can be used to identify, analyze, and summarize people's perspectives. We applied our study to two different domains: social media and scientific literature.

In the social media domain, we focused on threaded discussions that discuss political and ideological topics. This type of discussions can be found on social networking sites, on discussion forums, on image and video sharing sites, on news web sites, etc. The main research question that we tried to answer is: how natural language analysis techniques can be used to identify the different opinions and viewpoints of the participants of online discussions. To answer this question, we first studied the genre of online discussions. We studied the language used by participants for arguing for or against an opinion. We studied how participants support other participants who share the same opinion with them and how they attempt to dispute the opinions of opposing participants.

We observed that polarized expressions are commonly used in discussions to express opinion. A significant portion of these polarized expressions are colloquial words or acronyms that do not exist in the available polarity lexicon. This motivated the need for methods that detect the polarity of out-of-vocabulary words. We also observed that participants may switch back and forth, in the same post, between emphasizing their opinions and disputing opposing opinions. This means that both positive and negative polarities may be used in the same post. This motivated our need for identifying the targets of opinion and associate each polarity expression with its target. We also noticed that most discussion groups that discuss controversial topics split into subgroups with contrasting opinions. This motivated us to work on automatically identifying opinion subgroups by analyzing the text posted by the discussants.

In the scientific literature domain, we focused on studying citations. We used NLP techniques to analyze the text that accompanies citations in scientific articles. The goal of this study is to see how important a researcher's work is from the viewpoints of other researchers. This is useful for developing more accurate bibliometric measures that evaluate both the quantitative and the qualitative aspects of published research. This is also useful for many applications such as paper summarization, survey generation, and studying the dynamics of research. We proposed methods for identifying, extracting, and cleaning citation text from scientific articles. We used linguistic analysis techniques to analyze this text and identify the author intention behind selecting a paper and citing it. We also used citation text to identify the polarity of citation; i.e. the author sentiment towards the cited work. We use a supervised approach for these problems. We present a number of applications in which the analysis of citations can be useful such as generating citation-based summaries of scientific articles, predicting future prominence of scientific articles, and studying the dynamics of research.

The thesis falls in two parts. The first part consists of 5 chapters (II - VI) and covers the work we did in the social media domain. The second part consists of 5 chapters (VII - XI) and covers the work we did in the scientific literature domain.

In Chapter III, we addressed the problem of identifying opinion expressions. We extended an existing random-walk based method [76] to make it capable of identifying the polarity of out-of-vocabulary (OOV) words. OOV words are ones that do not exist in the available polarity lexicons and are not defined in the standard dictionaries of the language. We augment a semantic graph constructed from Wordnet synsets with the OOV words that we want to identify the polarity of. The relatedness of the OOV word and the other words that exist in the graph is determined based on co-occurrence statistics computed from a large corpus of social text. In our experiments we used a large corpus of tweets. Once the OOV word bercomes part of the semantic graph, the random-walk method is applied to identify its polarity. The random walk method that we used identifies the polarity of words in a semi-supervised classification fashion. The polarity of a given word is computed based on the mean hitting time to a set of positive and negative seed words. If the difference between the mean hitting time to positive and the mean hitting time to negative is smaller than some threshold (picked experimentally), the word is classified as neutral. Otherwise, the word is classified as positive if the mean hitting time to the positive seeds is higher or as negative if the mean hitting time to the negative seeds is higher. We compared our method to several baselines including ones that use co-occurrence statistics computed using the whole web as a corpus. We showed that using a corpus of social text gives better results for this task than using the whole web or a generic text corpus.

In Chapter IV, we addressed the problem of detecting negation and identifying its scope. Handling negation is important for sentiment analysis and opinion mining. The polarity of a word is reversed when it occurs in a negated context. Handling negation involves two tasks. First, we detect negation cues. A negation cue is a word, a prefix, or a postfix that triggers negation. Second, we identify the scope of a negation in the sentence it appears in. The scope of negation is the part of the sentence that is negated. We proposed a sequence labeling method for the two tasks.

Chapter V addressed the problem of identifying the targets of attitudes in discussions. Addressing this problem is important in discussions because we observed that both positive and negative opinion expressions are used by participants. Even in one post, the same participant may switch back and forth between arguing for his/her opinion and arguing against the contrasting opinions. This means that we cannot rely only on the polarity of opinion expressions to identify perspective. It is important that we take the targets of opinion into account as well. We experimented with three methods for this task. The first method is unsupervised. It uses a set of hand-crafted rules to identify the targets of opinion expressions. The rules are based on the dependency parse tree of the sentence that contains the opinion expression and the candidate target. The two other methods are supervised. In one of them, we addressed the problem as a sequence labeling task. For every opinion expression, we label every word in the sentence as being part of the target of that opinion word or not. In the other supervised method, we use a classification setup. We first identify all candidate targets of opinion and then determine for every possible opinion-target pair whether the candidate target is actually a target of the opinion word or not. We use as candidate targets named entities and frequent noun phrases. Our experiments showed that the classification and the sequence labeling approaches achieve better results than the rule-based method. The rule based method achieves higher precision but significantly lower recall.

Chapter VI describes how we encode the opinion and attitude information identified using the methods presented in the previous chapters in a formal representation. We describe two representations. A signed network representation and a vector representation. In the signed network representation each participant is represented by a node in a network. Edges connect participants who interact in the discussion. The sign of the edge is positive if the textual analysis of their posts shows that they share the same opinion with respect to the discussion topic, otherwise the sign is negative. In the second representation, each participant is represented by a vector. Each vector acts as an attitude profile that stores the attitude information of the participant towards every other participant and towards the topical targets. The chapter also describes an application in which we use network partitioning and vector space clustering techniques to detect subgroups of discussants who share the same opinion.

The second part of the thesis focuses on identifying and analyzing researchers' viewpoints towards previous work through the lens of citations. Chapter VIII presents methods for identifying citation text and extracting citing sentences and citation context. Citation text is the succession of sentences that appear around a reference anchor and comment on the referenced work [143]. The first step to identify citation text is identifying reference anchors. We used a set of regular expression rules to identify such anchors. The sentence that contains a reference anchor is termed *citing sentence*. The text that comments on a cited work may span multiple sentences around the citing sentence. We use the term *citation context* to refer to the sentences adjacent to the citing sentence and comment on the cited work. We propose a sequence labeling method for identifying citation context using a set of structural and lexical features that capture the semantic relation between the surrounding sentences and the citing sentence.

We also addressed the problem of identifying the scope of a reference in citing sentences that cite multiple papers. The scope of a reference is the fragment of the sentence that talks about a given target reference. We experimented with two supervised methods for addressing this problem. The first method tackles the problem as a classification task where each word in the sentence is classified as *included* or *excluded* from the scope of a given target reference. The second method tackles the problem as a sequence labeling problem where the goal is to find the best sequence of *included* and *excluded* labels for all the words in the sentence. Identifying the scope of a reference is an important preprocessing step for many applications that use citation text. For example, in citation-based summarization, only the text that describes the summarized paper should be included in the summary [10].

Chapter X presents a method that uses citation text to generate multi-perspective summaries of scientific articles. This type of summaries is different from abstracts or summaries generated from the paper itself. Citation-based summaries summarize the contributions of the paper from the perspectives of other scholars who read the paper and identified its strengths and weaknesses and its important contributions.

Our method is different from previous work in that it focuses on the coherence and the readability of the produced summaries. Our approach achieved better performance than previous methods in terms of both content selection and readability.

Chapter XI concludes the second part of the thesis. It presents a set of applications in which citation text can be used to analyze scientific literature. These applications include predicating the future prominence of research articles, analyzing the dynamics of research, identifying controversial scientific topics, and others.

CHAPTER XIII

Future Directions

In this chapter, we present directions for future work that can build on the findings and the conclusions of this thesis.

13.1 Computational Psycholinguistics and Sociolinguistics

Psycholinguistic and Sociolinguistics are two areas that are underexplored by computational linguistic researchers [46, 171]. We believe that there exists a large gap between the progress achieved in sociolinguistics and psycholinguistics and the efforts made in the NLP and IR communities to build computational models that verify, utilize, and apply the findings made by sociolinguists and psycholinguists. In future work, we plan attempt to narrow this gap by spending more time on studying the literature and basing our computational approaches to mining perspectives on sociolinguistic and psycholinguistic theories.

13.2 Detecting Influencers in Online Discussions

One interesting research direction we are interested in exploring in the future is detecting influencers in discussions [31]. This involves studying the behavioral and linguistic features that could make someone an opinion leader who can influence others' opinions. This problem has not been thoroughly explored and there are a lot of sociolinguistic theories around influence and its relation to language that have not been investigated computationally. There are several discussion forums on the internet that can be good sources of data for studying influence and training computational model for detection influences.

13.3 Multiple viewpoint summarization of discussions

In this thesis, we showed how we mine opinion and opinion targets from discussions. We also presented two approaches for detecting opinion subgroups in discussion communities. The output of this work can be used to build systems for summarizing the different viewpoints that the participants in a discussion have with respect to the discussion topic. Such a system should be able to identify the most important points raised by each subgroup and include them in the summary. The produced summary should list the disputed aspects of the discussed topic and select sentences that express the different viewpoints for each aspect.

13.4 Studying Discussion Group Dynamics

Another interesting research direction that can build on our work is studying the dynamics of discussion groups. For example, it is interesting to study how the rift in a discussion group starts and how it leads to the split of discussants into multiple subgroups with contrasting opinions, and how these subgroups evolve with time. It is also interesting see what correlation there is between the evolution of subgroups and the language used by the members of those subgroups.

13.5 Predicting Future Prominence of Scientific Articles

In chapter XI, we presented initial results on using textual analysis of citations to predict the number of citations that a paper may receive in the future. We are planning to continue working on this problem with a focus on indicators that can be extracted from citation text. We are specifically interested in studying *change in prominence* with a focus on detecting papers that do not attract attention at the time of publication, but becomes prominent in the future, or those that receive a lot of attention for a short time after publication and then their prominence drops quickly. Our hypothesis is that analyzing citation text provides useful indicators of future prominence. This work has many useful applications such as developing more accurate bibliometric measures and building systems that aid in funding and hiring decisions at research funding agencies and research institutions.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Daryl E. and Soumyo D. Moitra. Content analysis of references: Adjunct or alternative to citation counting? *Social Studies of Science*, 5(4):pp. 423–441, 1975.

[2] Ahmed Abbasi, Hsinchun Chen, and Arab Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.*, 26(3):12:1–12:34, June 2008.

[3] Muhammad Abdul-Mageed and Mona Diab. Subjectivity and sentiment annotation of modern standard arabic newswire. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 110–118, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[4] Muhammad Abdul-Mageed and Mona Diab. Awatif: A multi-genre corpus for modern standard arabic subjectivity and sentiment analysis. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[5] Muhammad Abdul-Mageed and Mona Diab. Toward building a large-scale arabic sentiment lexicon. In *Proceedings of the 6th International Global Word-Net Conference, Matsue, Japan*, 2012.

[6] Muhammad Abdul-Mageed, Sandra Kuebler, and Mona Diab. Samar: A system for subjectivity and sentiment analysis of arabic social media. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 19–28, Jeju, Korea, July 2012. Association for Computational Linguistics.

[7] Amjad Abu-Jbara, Jefferson Ezra, and Dragomir R. Radev. Purpose and polarity of citation: Towards nlp-based bibliometrics. In *Proceedings of the North American Association for Computational Linguistics*, 2013.

[8] Amjad Abu-Jbara, Ahmed Hassan, and Dragomir Radev. Attitudeminer: A system for mining attitude from discussions. In *Proceedings of the North American Chapter of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Montreal, Canada, July 2012. North American Chapter of the Association for Computational Linguistics.

[9] Amjad Abu-Jbara, Benjamin King, Mona Diab, and Dragomir R. Radev. Identifying opinion subgroups in arabic online discussions. In *Proceedings of The Association for Computational Linguistics (short paper)*, 2013.

[10] Amjad Abu-Jbara and Dragomir Radev. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 500–509, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[11] Amjad Abu-Jbara and Dragomir Radev. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 500–509, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[12] Amjad Abu-Jbara and Dragomir Radev. A conditional random field model for detecting negation cues and their scopes. In **SEM Shared Task 2012*, Montreal, Canada, June 2012.

[13] Amjad Abu-Jbara and Dragomir Radev. Identifying reference scope in citation setences. In *Proceedings of the North American Chapter of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Montreal, Canada, June 2012. North American Chapter of the Association for Computational Linguistics.

[14] Amjad Abu Jbara and Dragomir Radev. Reference scope identification in citing sentences. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 80–90, Montréal, Canada, June 2012. Association for Computational Linguistics.

[15] Amjad Abu-Jbara and Dragomir Radev. Subgroup detection in ideological discussions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Jeju, Korea, July 2012. The Association for Computational Linguistics.

[16] Amjad Abu-Jbara and Dragomir Radev. Subgroup detector: A system for detecting subgroups in online discussions. In *Proceedings of the ACL 2012 System Demonstrations*, pages 133–138, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[17] Shashank Agarwal and Hong Yu. Biomedical negation scope detection with conditional random fields. *Journal of the American Medical Informatics Association*, 17(6):696–701, 2010.

[18] Eneko Agirre, Arantxa Otegi, and Hugo Zaragoza. Using semantic relatedness and word sense disambiguation for (cl)ir. In *Proceedings of the 10th cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments*, CLEF'09, pages 166–173, Berlin, Heidelberg, 2009. Springer-Verlag.

[19] Toni Ahlqvist. *Social media roadmaps: exploring the futures triggered by social media.* VTT, 2008.

[20] F. Amblard, A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. On the temporal analysis of scientific network evolution. In *Computational Aspects of Social Networks (CASoN), 2011 International Conference on*, pages 169 –174, oct. 2011.

[21] Pranav Anand, Marilyn Walker, Rob Abbott, Jean E. Fox Tree, Robeson Bowmani, and Michael Minor. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 1–9, Portland, Oregon, June 2011. Association for Computational Linguistics.

[22] Alina Andreevskaia and Sabine Bergler. Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *EACL'06*, 2006.

[23] Awais Athar. Sentiment analysis of citations using sentence structure-based features. In *Proceedings of the ACL 2011 Student Session*, pages 81–87, Portland, OR, USA, June 2011. Association for Computational Linguistics.

[24] Awais Athar and Simone Teufel. Context-enhanced citation sentiment detection. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 597–601, Montréal, Canada, June 2012. Association for Computational Linguistics.

[25] Awais Athar and Simone Teufel. Detection of implicit citations for sentiment detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 18–26, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[26] Carmen Banea, Rada Mihalcea, and Janyce Wiebe. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In *LREC'08*, 2008.

[27] Ann Banfield. *Unspeakable Sentences: Narration and Representation in the Language of Fiction*. Routledge and Kegan Paul, 1982.

[28] Mohit Bansal, Claire Cardie, and Lillian Lee. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework, 2008.

[29] Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. Automatic extraction of opinion propositions and their holders. In *2004 AAAI Spring Symposium on Exploring Attitude and Affect in Text*, page 2224, 2004.

[30] Douglas Biber. *Variation across speech and writing*. Cambridge University Press, Cambridge, 1988.

[31] Or Biran, Sara Rosenthal, Jacob Andreas, Kathleen McKeown, and Owen Rambow. Detecting influencers in written online conversations. In *Proceedings of the Second Workshop on Language in Social Media*, pages 37–45, Montréal, Canada, June 2012. Association for Computational Linguistics.

[32] Susan Bonzi. Characteristics of a literature as predictors of relatedness between cited and citing works. *Journal of the American Society for Information Science*, 33(4):208–216, 1982.

[33] Christine L. Borgman and Jonathan Furner. Scholarly communication and bibliometrics. *Annual review of information science and technology*, 36(1):2–72, 2002.

[34] Tim Buckwalter. Issues in arabic orthography and morphology analysis. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Semitic '04, pages 31–34, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[35] Razvan Bunescu and Raymond Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[36] Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, and Sebastian Pado. *SALTO A Versatile Multi-Level Annotation Tool*, page 517520. European Language Resources Association (ELRA), 2006.

[37] Minh Duc Cao and Xiaoying Gao. Combining contents and citations for scientific document classification. In *Proceedings of the 18th Australian Joint conference on Advances in Artificial Intelligence*, AI'05, pages 143–152, Berlin, Heidelberg, 2005. Springer-Verlag.

[38] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[39] Wendy Webber Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, pages 301–310, 2001.

[40] Kenneth Ward Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA, February 1988. Association for Computational Linguistics.

[41] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70(6):066111, Dec 2004.

[42] Aaron Clauset, Mark E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004.

[43] J. Cohen. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70:213–220, 1968.

[44] Michael Collins. Three generative, lexicalised models for statistical parsing. In *ACL*, pages 16–23, 1997.

[45] Isaac G. Councill, Ryan McDonald, and Leonid Velikovich. What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, NeSp-NLP '10, pages 51–59, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[46] Matthew W Crocker. Computational psycholinguistics.

[47] Sanjoy Dasgupta. Performance guarantees for hierarchical clustering. In *15th Annual Conference on Computational Learning Theory*, pages 351–363. Springer, 2002.

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

[49] Mona Diab. Second generation amira tools for arabic processing: Fast and robust tokenization, pos tagging, and base phrase chunking, 2009.

[50] Lubomar Doleezel. *Narrative modes in Czech literature.* University of Toronto Press, 1973.

[51] Patrick Doreian and Andrej Mrvar. A partitioning approach to structural balance. *Social Networks*, 18(2):149–168, 1996.

[52] Leo Egghe. Theory and practise of the g-index. *Scientometrics*, 69:131–152, 2006.

[53] Aaron Elkiss, Siwei Shen, Anthony Fader, Güneş Erkan, David States, and Dragomir Radev. Blind men and elephants: What do citation summaries tell us about a research article? *J. Am. Soc. Inf. Sci. Technol.*, 59(1):51–62, 2008.

[54] Katrin Erk and Sebastian Pado. Shalmaneser - a flexible toolbox for semantic role assignment. In *Proceedings of LREC 2006*, Genoa, Italy, 2006.

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

[56] Andrea Esuli and Fabrizio Sebastiani. Determining the semantic orientation of terms through gloss classification. In *CIKM'05*, pages 617–624, 2005.

[57] Andrea Esuli and Fabrizio Sebastiani. Determining the semantic orientation of terms through gloss classification. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 617–624, New York, NY, USA, 2005. ACM.

[58] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC'06*, pages 417–422, 2006.

[59] Charles J Fillmore. The case for case., 1967.

[60] E. Garfield. Is citation analysis a legitimate evaluation tool? *Scientometrics*, 1(4):359–375, 1979.

[61] E. Garfield, Irving H. Sher, and R. J. Torpie. *The Use of Citation Data in Writing the History of Science*. Institute for Scientific Information Inc., Philadelphia, Pennsylvania, USA, 1984.

[62] Eugene Garfield. Can citation indexing be automated?, 1964.

[63] Eugene Garfield. The thomson reuters impact factor, 1994.

[64] Johannes Gehrke, Paul Ginsparg, and Jon Kleinberg. Overview of the 2003 kdd cup. *SIGKDD Explor. Newsl.*, 5(2):149–151, December 2003.

[65] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[66] Stephan Charles Greene. *Spin: lexical semantics, transitivity, and the identification of implicit sentiment*. ProQuest, 2007.

[67] Gregory Grefenstette, Yan Qu, James G Shanahan, and David A Evans. Coupling niche browsers and affect analysis for an opinion mining application. In *Proceedings of RIAO*, volume 4, pages 186–194. Citeseer, 2004.

[68] Claire Grover, Colin Matheson, Andrei Mikheev, and Marc Moens. Lt ttt - a flexible tokenisation tool. In *In Proceedings of Second International Conference on Language Resources and Evaluation*, pages 1147–1154, 2000.

[69] Weiwei Guo and Mona Diab. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 864–872, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[70] Sonal Gupta and Christopher Manning. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1–9, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

[71] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 46:389–422, March 2002.

[72] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[73] Ahmed Hassan, Amjad Abu-Jbara, Rahul Jha, and Dragomir Radev. Identifying the semantic orientation of foreign words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 592–597, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[74] Ahmed Hassan, Amjad Abu Jbara, and Dragomir R. Radev. Identifying polarity of foreign words. Submitted to ACL, 2011.

[75] Ahmed Hassan, Vahed Qazvinian, and Dragomir Radev. What's with the attitude?: identifying sentences with attitude in online discussions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1245–1255, 2010.

[76] Ahmed Hassan and Dragomir Radev. Identifying text polarity using random walks. In *ACL'10*, 2010.

[77] Ahmed Hassan and Dragomir Radev. Identifying text polarity using random walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 395–403, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[78] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *EACL'97*, pages 174–181, 1997.

[79] Vasileios Hatzivassiloglou and Janyce Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *COLING*, pages 299–305, 2000.

[80] Vasileios Hatzivassiloglou and Janyce M Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 299–305. Association for Computational Linguistics, 2000.

[81] John Digby Haynes. *Perspectival Thinking for Inquiring Organisations*. Informing Science, 2000.

[82] PatrickA. Heelan. Nietzsches perspectivalism: A hermeneutic philosophy of science. In BabetteE. Babich, editor, *Nietzsche, Epistemology, and Philosophy of Science*, volume 204 of *Boston Studies in the Philosophy of Science*, pages 203–220. Springer Netherlands, 1999.

[83] Bas Heerschop, Paul van Iterson, Alexander Hogenboom, Flavius Frasincar, and Uzay Kaymak. Analyzing sentiment in a large set of web data while accounting for negation. In *AWIC*, pages 195–205, 2011.

[84] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46):16569–16572, November 2005.

[85] J. E. Hirsch. An index to quantify an individual's scientific research output that takes into account the effect of multiple coauthorship. *Scientometrics*, 85(3):741–754, December 2010.

[86] Hochbaum and Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.

[87] T. L. Hodges. Citation indexing-its theory and application in science, technology, and humanities. *Ph.D. thesis, University of California at Berkeley.Ph.D. thesis, University of California at Berkeley.*, 1972.

[88] Alexander Hogenboom, Paul van Iterson, Bas Heerschop, Flavius Frasincar, and Uzay Kaymak. Determining negation scope and strength in sentiment analysis. In *SMC*, pages 2589–2594, 2011.

[89] Laurence R. Horn. *A natural history of negation / Laurence R. Horn*. University of Chicago Press, Chicago :, 1989.

[90] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD'04*, pages 168–177, 2004.

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

[92] Rodney D. Huddleston and Geoffrey K. Pullum. *The Cambridge Grammar of the English Language*. Cambridge University Press, April 2002.

[93] Ken Hyland. The author in the text: Hedging scientific writing. *Hong Kong papers in linguistics and language teaching*, 18:33–42, 1995.

[94] Niklas Jakob and Iryna Gurevych. Using anaphora resolution to improve opinion target identification in movie reviews. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 263–268, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[95] Lifeng Jia, Clement Yu, and Weiyi Meng. The effect of negation on sentiment analysis and retrieval effectiveness. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1827–1830, New York, NY, USA, 2009. ACM.

[96] Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. Using wordnet to measure semantic orientation of adjectives. In *National Institute for*, pages 1115–1118, 2004.

[97] Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. Using wordnet to measure semantic orientations of adjectives. In *National Institute for*, pages 1115–1118, 2004.

[98] Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *EMNLP'06*, pages 355–363, 2006.

[99] Andreas M. Kaplan and Michael Haenlein. Users of the world, unite! The challenges and opportunities of social media. *Business Horizons*, 53(1):59 – 68, 2010.

[100] Jason S. Kessler, Miriam Eckert, Lyndsie Clark, and Nicolas Nicolov. The 2010 icwsm jdpa sentment corpus for the automotive domain. In *4th International AAAI Conference on Weblogs and Social Media Data Workshop Challenge (ICWSM-DWC 2010)*, 2010.

[101] Jung-Jae Kim and Jong C. Park. Extracting contrastive information from negation patterns in biomedical literature. *ACM Transactions on Asian Language Information Processing (TALIP)*, 5(1):44–60, March 2006.

[102] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *COLING*, pages 1367–1373, 2004.

[103] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[104] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *In Proceedings fo the 41st Annual Meeting of the Associations for Computational Linguistics*, pages 423–430, 2003.

[105] Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL*, 2007.

[106] Klaus H. Krippendorff. *Content Analysis: An Introduction to Its Methodology.* Sage Publications, Inc, 2nd edition, December 2003.

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

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

[109] J. Richard Landis and Gary G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174, March 1977.

[110] Michael Laver, Kenneth Benoit, and John Garry. Extracting policy positions from political texts using words as data. *American Political Science Review*, 97(02):311–331, 2003.

[111] Adrienne Lehrer. *Semantic fields and lexical structure*, volume 11. North-Holland Amsterdam, 1974.

[112] Adrienne Lehrer. Semantic fields and lezical structure. North Holland, Amsterdam and New York, 1974.

[113] Wei-Hao Lin and Alexander Hauptmann. Are these documents written from different perspectives?: a test of different perspectives based on statistical distribution divergence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1057–1064. Association for Computational Linguistics, 2006.

[114] Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann. Which side are you on?: identifying perspectives at the document and sentence levels. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 109–116. Association for Computational Linguistics, 2006.

[115] Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer, 1st ed. 2007. corr. 2nd printing edition, January 2009.

[116] Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 415–463. Springer US, 2012.

[117] Terttu Luukkonen. Is scientists' publishing behaviour rewardseeking? *Scientometrics*, 24:297–319, 1992. 10.1007/BF02017913.

[118] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, December 2007.

[119] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[120] Michael H MacRoberts and Barbara R MacRoberts. The negational reference: Or the art of dissembling. *Social Studies of Science*, pages 91–94, 1984.

[121] Michael H. MacRoberts and Barbara R. MacRoberts. The negational reference: Or the art of dissembling. *Social Studies of Science*, 14(1):pp. 91–94, 1984.

[122] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[123] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[124] Amin Mazloumian, Young-Ho Eom, Dirk Helbing, Sergi Lozano, and Santo Fortunato. How citation boosts promote scientific paradigm shifts and nobel prizes. *PLoS ONE*, 6(5):e18975, 05 2011.

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

[126] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM, 2007.

[127] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 171–180, New York, NY, USA, 2007. ACM.

[128] Qiaozhu Mei and ChengXiang Zhai. Generating impact-based summaries for scientific literature. In *Proceedings of ACL-08: HLT*, pages 816–824, Columbus, Ohio, June 2008. Association for Computational Linguistics.

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

[130] Soo min Kim and Eduard Hovy. Crystal: Analyzing predictive opinions on the web. In *In EMNLPCoNLL 2007*, 2007.

[131] Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. Using citations to generate surveys of scientific paradigms. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 584–592, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[132] Saif Mohammad, Cody Dunne, and Bonnie Dorr. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 599–608, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[133] Roser Morante. Descriptive analysis of negation cues in biomedical texts. *Language Resources And Evaluation*, pages 1–8, 2010.

[134] Roser Morante and Eduardo Blanco. *sem 2012 shared task: resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 265–274, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[135] Roser Morante and Walter Daelemans. Learning the scope of hedge cues in biomedical texts. *Proceedings of the Workshop on BioNLP BioNLP 09*, page 28, 2009.

[136] Roser Morante, Anthony Liekens, and Walter Daelemans. Learning the scope of negation in biomedical texts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP 08*, pages 715–724, 2008.

[137] Roser Morante, Anthony Liekens, and Walter Daelemans. Learning the scope of negation in biomedical texts. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 715–724, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

[138] Roser Morante, Sarah Schrauwen, and Walter Daelemans. Annotation of negation cues and their scope, 2011.

[139] M. J. Moravcsik and P. Murugesan. Some results on the function and quality of citations. *Social Studies of Science*, 5:86–92, 1975.

[140] P. G. Mutalik, A. Deshpande, and P. M. Nadkarni. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS. *Journal of the American Medical Informatics Association : JAMIA*, 8(6):598–609, 2001.

[141] Preslav I. Nakov, Ariel S. Schwartz, and Marti A. Hearst. Citances: Citation sentences for semantic analysis of bioscience text. In *In Proceedings of the SIGIR04 workshop on Search and Discovery in Bioinformatics*, 2004.

[142] Hidetsugu Nanba, Noriko Kando, Manabu Okumura, and Of Information Science. Classification of research papers using citation links and citation types: Towards automatic review article generation, 2000.

[143] Hidetsugu Nanba and Manabu Okumura. Towards multi-paper summarization using reference information. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 926–931, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[144] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: capturing favorability using natural language processing. In *K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77, 2003.

[145] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[146] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[147] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[148] Michael J. Paul, ChengXiang Zhai, and Roxana Girju. Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 66–76, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[149] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *HLT-EMNLP'05*, pages 339–346, 2005.

[150] Ana-Maria Popescu and Orena Etzioni. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer, 2007.

[151] Vern S Poythress. *Symphonic theology: The validity of multiple perspectives in theology.* Zondervan, 1987.

[152] Vahed Qazvinian and Dragomir R. Radev. Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 689–696, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[153] Vahed Qazvinian and Dragomir R. Radev. Identifying non-explicit citing sentences for citation-based summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 555–564, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[154] Vahed Qazvinian, Dragomir R. Radev, Saif Mohammad, Bonnie Dorr, David Zajic, Michael Whidby, and Taesun Moon. Generating extractive summaries of scientific paradigms. *Journal of Artificial Intelligence Research*, 2013.

[155] Vahed Qazvinian, Dragomir R. Radev, and Arzucan Ozgur. Citation summarization through keyphrase extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 895–903, Beijing, China, August 2010. Coling 2010 Organizing Committee.

[156] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27, 2011.

[157] Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. *A Comprehensive Grammar of the English Language*. Longman, London, 1985.

[158] Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, Jan Svartvik, and David Crystal. *A comprehensive grammar of the English language*, volume 397. Cambridge Univ Press, 1985.

[159] Dragomir Radev and Amjad Abu-Jbara. Rediscovering acl discoveries through the lens of acl anthology network citing sentences. In *The ACL Special Workshop: Rediscovering 50 Years of Discoveries*, Jeju, Korea, July 2012. The Association for Computational Linguistics.

[160] Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. The acl anthology network corpus. In *NLPIR4DL '09: Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 54–61, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[161] Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. The acl anthology network corpus. *Language Resources and Evaluation*, pages 1–26, 2013.

[162] Delip Rao and Deepak Ravichandran. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 675–682, Athens, Greece, March 2009. Association for Computational Linguistics.

[163] Jonathon Read, Erik Velldal, Stephan Oepen, and Lilja vrelid. Resolving speculation and negation scope in biomedical articles with a syntactic constituent ranker. In *Proceedings of the Fourth International Symposium on Languages in Biology and Medicine*, Singapore, 2011.

[164] Sidney Redner. Citation statistics from 110 years of physical review. *Physics Today*, 58(6):49–54, 2005.

[165] Ellen Riloff, Siddharth Patwardhan, and Janyce Wiebe. Feature subsumption for opinion analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 440–448. Association for Computational Linguistics, 2006.

[166] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *EMNLP'03*, pages 105–112, 2003.

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

[168] Christopher Scaffidi, Kevin Bierhoff, Eric Chang, Mikhael Felker, Herman Ng, and Chun Jin. Red opal: product-feature scoring from reviews. In *Proceedings of the 8th ACM conference on Electronic commerce*, EC '07, pages 182–191, New York, NY, USA, 2007. ACM.

[169] Richard Schacht. *Making sense of Nietzsche: Reflections timely and untimely*. University of Illinois Press, 1995.

[170] Ariel Schwartz, Anna Divoli, and Marti Hearst. Multiple alignment of citation sentences with conditional random fields and posterior decoding. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 847–857, 2007.

[171] Walter A Sedelow Jr. Computational sociolinguistics., 1967.

[172] Yohei Seki, Koji Eguchi, Noriko Kando, and Masaki Aono. Opinion-focused summarization and its analysis at duc 2006. In *Proceedings of the Document Understanding Conference (DUC)*, pages 122–130, 2006.

[173] Advaith Siddharthan and Simone Teufel. Whose idea was this, and why does it matter? attributing scientific work to citations. In *In Proceedings of NAACL/HLT-07*, 2007.

[174] Daniel D. K. Sleator and Davy Temperley. Parsing english with a link grammar. In *In Third International Workshop on Parsing Technologies*, 1991.

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

[176] Ina Spiegel-Rösing. Science Studies: Bibliometric and Content Analysis. *Social Studies of Science*, 7(1):97–113, February 1977.

[177] Philip Stone. Thematic text analysis: new agendas for analyzing text content. In Carl Roberts, editor, *Text Analysis for the Social Sciences.* Lawerence Erlbaum Associates, Mahwah, NJ, 1997.

[178] Philip Stone, Dexter Dunphy, Marchall Smith, and Daniel Ogilvie. The general inquirer: A computer approach to content analysis. *The MIT Press*, 1966.

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

[180] Veselin Stoyanov and Claire Cardie. Topic identification for fine-grained opinion analysis. In *In Coling*, 2008.

[181] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientations of words using spin model. In *ACL'05*, pages 133–140, 2005.

[182] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientations of words using spin model. In *In ACL*, pages 133–140, 2005.

[183] Simone Teufel. Argumentative zoning for improved citation indexing. computing attitude and affect in text. In *Theory and Applications, pages 159170*, 2007.

[184] Simone Teufel, Advaith Siddharthan, and Dan Tidhar. Automatic classification of citation function. In *In Proc. of EMNLP-06*, 2006.

[185] Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *In Proceedings of EMNLP*, pages 327–335, 2006.

[186] GEOFF THOMPSON and YE YIYUN. Evaluation in the reporting verbs used in academic papers. *Applied Linguistics*, 12(4):365–382, 1991.

[187] Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. *Urbana*, 51:61801, 2008.

[188] Richard Tong. An operational system for detecting and tracking opinions in on-line discussions. In *Working Notes of the SIGIR Workshop on Operational Text Classification*, pages 1–6, New Orleans, Louisianna, 2001.

[189] Gunnel Tottie. Negation in English Speech and Writing: A Study in Variation. *Language*, 69(3):590–593, 1993.

[190] Peter Turney and Michael Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346, 2003.

[191] Peter Turney and Michael Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346, 2003.

[192] Boris Andreevič Uspenskij. *A Poetics of Composition: The Structure of the Poetic Text and Typology of a Compositional Form.* Univ of California Press, 1973.

[193] Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–785, Los Angeles, California, June 2010. Association for Computational Linguistics.

[194] Melvin Weinstock. *Citation Indexes.* Encyclopedia of Library and Information Science, 1971.

[195] Michael Alan Whidby. Citation handling: Processing citation text in scientific documents. In *Master Thesis*, 2012.

[196] Howard D. White. Citation analysis and discourse analysis revisited. *Applied Linguistics*, 25(1):89–116, 2004.

[197] Janyce Wiebe. Learning subjective adjectives from corpora. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 735–740, 2000.

[198] Janyce Wiebe, Rebecca Bruce, Matthew Bell, Melanie Martin, and Theresa Wilson. A corpus study of evaluative and speculative language. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, pages 1–10, 2001.

[199] Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning subjective language. *Comput. Linguist.*, 30(3):277–308, September 2004.

[200] Janyce M Wiebe. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287, 1994.

[201] Janyce M. Wiebe. Learning subjective adjectives from corpora. In *In AAAI*, pages 735–740, 2000.

[202] Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. A survey on the role of negation in sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, NeSp-NLP '10, pages 60–68, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[203] Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Opinionfinder: a system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, HLT-Demo '05, pages 34–35, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

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

[205] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP'05*, Vancouver, Canada, 2005.

[206] Rui Yan, Congrui Huang, Jie Tang, Yan Zhang, and Xiaoming Li. To better stand on the shoulder of giants. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, pages 51–60. ACM, 2012.

[207] Rui Yan, Jie Tang, Xiaobing Liu, Dongdong Shan, and Xiaoming Li. Citation count prediction: learning to estimate future citations for literature. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 1247–1252, New York, NY, USA, 2011. ACM.

[208] Bo Yang, William Cheung, and Jiming Liu. Community mining from signed social networks. *IEEE Trans. on Knowl. and Data Eng.*, 19(10):1333–1348, 2007.

[209] Ainur Yessenalina, Yisong Yue, and Claire Cardie. Multi-level structured models for document-level sentiment classification. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP*, 2010.

[210] Dani Yogatama, Michael Heilman, Chris Dyer, Bryan R. Routledge, Noah A. Smith, Dani Yogatama, Michael Heilman, Chris Dyer, Bryan R. Routledge, and Noah A. Smith. www.lti.cs.cmu.edu predicting responses and discovering social factors in scientific literature, 2011.

[211] Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP'03*, pages 129–136, 2003.

[212] Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Grouping product features using semi-supervised learning with soft-constraints. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1272–1280. Association for Computational Linguistics, 2010.

[213] M. Zhang, X. Gao, M.D. Cao, and Yuejin Ma. Neural networks for scientific paper classification. In *Innovative Computing, Information and Control, 2006. ICICIC '06. First International Conference on*, volume 2, pages 51 –54, 30 2006-sept. 1 2006.

[214] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM, 2006.

[215] J. M. Ziman. *Public knowledge: An essay concerning the social dimension of science*. Cambridge U.P., London, 1968.

[216] John M Ziman. *Public knowledge: an essay concerning the social dimension of science*, volume 519. London: Cambridge UP, 1968.