Identifying Roles in Social Networks using Linguistic Analysis

by

Ahmed Mohamed Hassan Awadallah

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
2011

Doctoral Committee:

Professor Dragomir R. Radev, Chair
Professor Hosagrahar V. Jagadish
Associate Professor Lada A. Adamic
Assistant Professor Michael J. Cafarella

To my parents, my wife, and my lovely kids.

ACKNOWLEDGEMENTS

First and foremost, all praise and thanks be to God, the Most Gracious, the Most Merciful, for giving me the strength and determination to finish this work.

I am truly grateful to my advisor Dragomir Radev. This thesis would not have been possible without him and without the encouragement he has given me over the past few years. His guidance and support were extremely influential both in my thinking, and in my development as a scientist. I would also like to thank all members of my dissertation committee, H.V. Jagadish, Lada Adamic, and Michael Cafarella for their constructive criticism and insightful feedback that helped shape and improve this work.

I would like to thank all members of the Computational Linguistics and Information Retrieval (CLAIR) group in the University of Michigan, Amjad Abu-Jbara, Vahed Qazvinian, Arzucan Ozgur, and Pradeep Muthukrishnan, for the fruitful collaboration, and insightful discussions that we had together. Many thanks to our collaborators on the SCIL project from Columbia University, especially Kathy McKeown, Mona Diab, and Owen Rambow.

I spent three great summers at Yahoo! Research, and Microsoft Research. Even thought, none of the work I did there is part of this thesis, working there was a great experience that helped me learn a lot. I would especially like to thank Rosie Jones. The time I spent working with Rosie was very educational, and very fruitful. She introduced me to the field of information retrieval and significantly influenced the development of my academic career. I strongly value her constant encouragement, and excellent mentoring that helped to shape the ways in which I look at and think about research.

I am very grateful to all my friends here in Ann Arbor especially Amjad Abu-Jbara, Khaled Al-Ashmouny, Samer Kadous,Ahmed Mady, Ahmed Al-Muhtady, and Ahmed Al-Sabagh. They have been a great family for me here. I cannot imagine how life would have been here if it were not for their friendship and support.

Words cannot express my gratitude to my parents. They taught me the value of knowledge and education, putting me on the path that has led me to here. Not a single page of this thesis would exist without their great sacrifices, amazing inspiration, and constant support.

Special thanks go to my wife Radwa for her great support and patience. Thanks for being so patient through the years I worked on this thesis. Thanks for bearing up with me when I had to spend almost all the time working. Thanks for putting off your goals in order for me to pursue mine. I truly appreciate all what you. Finally, I would like to thank my two beautiful children, Haya and Yassin for putting a smile on my face when I really needed it.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Social media sites have been significantly growing in the past few years. This resulted in the emergence of several communities of communicating groups, and a huge amount of text exchanged between members of those groups. In our work, we study how linguistic analysis techniques can be used for understanding the implicit relations that develop in on-line communities. We use this understanding to develop models that explain the processes that govern language use and how it reveals the formation of social relations. We study the relation between language choices and attitude between participants and how they may lead to or reveal antagonisms and rifts in social groups. Both positive (friendly) and negative (antagonistic) relations exist between individuals in communicating communities. Negative relations have received very little attention, when compared to positive relations, because of the lack of an explicit notion of labeling negative relations in most social computing applications. We alleviate this problem by studying text exchanged between participants to mine their attitude. Another important aspect of our research is the study of influence in discussions and how it affects participants' discourse. In any debate or discussion, there are certain types of persons who influence other people and affect their ideas and rhetoric. We rely on natural language processing techniques to find implicit connections between individuals that model this influence. We couple this with network analysis techniques for identifying the most authoritative or salient entities. We also study how salience evolves over time. Our work is uniquely characterized by combining linguistic features and network analysis to reveal social roles in different communities. The

methods we developed can find several interesting areas of applications. For example, they can be used for identifying authoritative sources in social media, finding influential people in communities, mining attitude toward events and topics, detecting rifts and subgroup formation, summarizing different viewpoints with respect to some topic or entity, and many other such applications.

CHAPTER I

Introduction

The main interest of this dissertation is in understanding the relation between the linguistic behavior of people who are communicating with each other and the social relations that develop between those people. We seek to answer several interesting questions like: What linguistic constructs do people use to show agreement, praise, etc.? What other constructs do they use to show disagreement, criticism, insults, etc.? What makes a group of communicating people split into subgroups? How can linguistic analysis reveal rifts in social groups? How do people influence other discourse participants opinions? Answering such questions is very important for a range of different applications. For example, they can be used for identifying authoritative sources in social media, finding influential people in the community, detecting subgroups, mining attitude toward events and topics, measuring public opinion and dispute levels of different topics, summarizing different view points with respect to some topic or entity, and many other such applications.

Our research will uncover how linguistic features can be used to predict social roles and behavior. We study two such cases where linguistic analysis can be used to identify social roles between people who are communicating with each other:

- **Mining attitude:** We study multi-party dialogs and discussions and mine the attitude of participants toward each other. We study the relation between language choices and attitude between participants and how they may lead to or reveal antagonisms and rifts in social groups. If we examine the relations between individuals in on-line communities, we will find that both positive (friendly) and negative (antagonistic) relations exist. Most of the research literature on social network analysis has almost exclusively focused on positive interactions only. The main reason behind that is the lack of an explicit notion of labeling negative relations in most social computing applications. We alleviate this problem by studying text exchanged between participants to mine their attitude.

- **Mining salience:** In any debate or discussion, there are certain types of persons who influence other people and pass information or advice to them. Those persons could be experts in the field. They may also have influence on the ideas and rhetoric of other participants. We study how linguistic analysis may reveal how people influence other discourse participants opinions. We rely on natural language processing techniques to find implicit connections between individuals that model how they were influenced by each other. We couple this with network analysis techniques for identifying the most authoritative or salient entities. We also extend our study to examining how this salience evolves over time.

1.1 Motivation

The move toward Web 2.0 gave rise to a lot of interest in research efforts that focus on searching and mining of the new content associated with it. The term Web 2.0 refers to Web applications based on user-centered design that encourages interaction and information sharing between users. Web 2.0 applications are different from non-interactive websites in that they allow their users to contribute to the content rather than limiting them to passively viewing it. This enabled the masses to easily create Web content. Examples of Web 2.0 sites include social networking sites, blogs, instant messaging, discussion forums and several other sites. The widespread use of the Internet in general, and the user-centered applications in particular resulted in several opportunities for studying the relations that develop between those users. Social network analysis focuses on the structure of relationships that emerge in social applications.

There is a great body of work that studied several aspects of networks in general and social networks in particular. For example, degree distributions of different networks have been studied for the Web, on-line social networks and other networks in [74, 15, 22, 117]. Clustering and clustering coefficients, which measures transitivity in networks, have been studied in [136]. The "small-world" and the effective network diameter have been studied in [6, 136]. Identifying communities in social and information networks has been studied in [45, 23]. Identifying basic building blocks of complex networks (i.e. network motifs) has been studied in [102].

Another line of research has focused on building models for network growth. The earliest such model was a random graph model introduced in [38]. The preferential attachment model and the power law degree distribution have been studied in [10, 25]. Finally the copying model was presented in [76].

Diffusion models that try to model the process of adoption of an idea or a product over the network have also been extensively studied. Examples include the Threshold model [47], and the Independent Cascade model [46].

These techniques have been applied to several domains including Blogs [1, 3], viral marketing and recommendation systems [118], scientific literature networks [108, 97], community answering websites [2], and on-line discussions [137].

In the mean time, there has been a lot of research work in the area of sociolinguistics. Sociolinguistics is the study of the effect of any and all aspects of society on the way language is used, and the effects of language use on society [49]. For example, prior work on sociolinguistics has studied *Honorifics*, words or expressions often used to indicate levels of esteem and respect [32]. Other work has studied *Politeness*. Several sociolinguistic studies have examined politeness and its relation to social roles and interaction between group members [16, 104]. Sociolinguistics is closely related to social networks. A social network can be thought of as a way of describing a particular community in terms of relations between individual members in that community [135]. Hence, Sociolinguistics is also concerned with understanding the social networks in which language is embedded.

Several related problems have also been studied in the computational linguistics literature. For example, research on lexical choice has studied the distinction between word selection for some one who is in a position of power vs. words that signify a peer. This has been used for developing lexicons for use in language generation [33, 64]. Other related work has studied the problem of syntactic choice and how alternative syntactic structures are used [44]. Prior work has also addressed the problem of studying dialog discourse structure. For example, identifying question-answering and dialog pairs in email exchanges has been studied in [121, 95].

In this work, we apply Natural Language Processing techniques to text correspondences exchanged between individuals to identify the underlying social structure in on-line communities. Our work is uniquely characterized by the fact that it uses linguistics analysis, along with other non-linguistic features, to shed light on the social relations developing between people given the text they exchange.

Most social network applications have some text embedded in the network. Analyzing this text could reveal an implicit rich relation structure between entities in the social network. For example, most of the social network analysis literature has almost exclusively focused on positive links between individuals. Most social networks have both positive and negative relations between participants. Considering both positive and negative relations could add much more insights that would help us understand the dynamics of the network. The main reason behind the little attention negative edges received is the lack of an explicit notion for individuals to label negative relations. Using linguistic analysis to implicitly identify positive and negative relations from text is a very reliable way for alleviating this problem.

Implicit links extracted from text could be also helpful for identifying salient or authoritative participants in discussions. Explicit links between nodes have been successfully used as a way of conferring authority. Unfortunately, such explicit links do not exist in many domains. Using linguistic analysis to predict implicit links that represent influence could be used to model salience in discussions.

1.2 Mining Attitude

The first part of this dissertation studies the problem of mining attitude from text. We study multi-party dialogs and discussions and mine the attitude of participants toward each other. We study the relation between language choices and attitude between participants and how they may lead to or reveal antagonisms and rifts in social groups. If we examine the relations between individuals in on-line communities, we will find that both positive (friendly) and negative (antagonistic) relations exist. Most of the research literature on social network analysis has almost exclusively focused on positive interactions only. The main reason behind that is the lack of an explicit mechanism for labeling negative relations in most social computing applications. We alleviate this problem by studying text exchanged between participants to mine their attitude.

A great body of research work has focused on social network analysis. Social network analysis plays a huge role in understanding and improving social computing applications.

Most of this research has almost exclusively focused on positive links between individuals (e.g. friends, fans, followers, etc.). However, if we carefully examine the relationships between individuals in on-line communities, we will find out that limiting links to positive interactions is a very simplistic assumption. It is true that people show agreement toward others, and label others as friends, but they also show disagreement, and antagonism toward other members of the on-line community. Discussion forums are one example that makes it clear that considering both positive and negative interactions is essential for understanding the rich relationships that develop between individuals in on-line communities.

If considering both negative and positive interactions will provide much more insight toward understanding the social network, why did most of previous work only focus on positive interactions? We think that one of the main reasons behind this is the lack of a notion for explicitly labeling negative relations. For example, most social Web applications allow people to mark others as friends, like them, follow them, etc. However, they do not allow people to explicitly label negative relations with others.

We propose a suite of methods that analyze text, exchanged during discussions, and extract a signed network representing the interactions between participants in this discussion. Signed networks have signs associated with every edge. The sign could be either positive or negative. Positive edges represent agreement, praise, etc. Negative edges represent disagreement, insult, etc.

A few recent papers have taken the signs of edges into account while studying social networks [17, 77, 84, 83]. All this work has been limited to analyzing a handful of datasets for which an explicit notion of both positive and negative relations exists. Our work goes beyond this limitation by leveraging the power of natural language processing to automate the discovery of signed social networks using the text embedded in the network.

In the first part of this dissertation, we describe how we can apply Natural Language Processing techniques to text correspondences exchanged between individuals to identify the underlying social structure in on-line communities. Our work is uniquely characterized by the fact that we mine both positive (friendly), and negative (antagonistic) interactions from text.

1.3 Mining Salience

In the second part of this dissertation, we analyze discussions and group dynamics to identify how participants influence other participants discourse. In any discussion or debate, certain people act as experts or influential people that get to affect the ideas and rhetoric of other participants. Assigning salience scores to participants in a discussion has several useful applications. This includes identifying authoritative sources of information, summarizing discussions, blog distillation, and many others.

The objective of the second part of this dissertation is to present the concept of content based networks and describe how they can be used for assigning salience scores to participants in discussions. We also extend the proposed methods to the case where the discussion spans a long period of time. In this case salience becomes a dynamic property that evolves with time.

Content similarity networks are networks that connect entities based on the similarity of their textual representation. This enables us to adopt link analysis techniques to new domains. Link analysis techniques are very popular in both Web search and social network studies. They have been successfully applied to several domains including Web page ranking, community finding and several other applications. Those techniques rely on explicit links between entities which limits their applicability to domains where explicit links are available.

Content similarity networks use linguistic analysis to find implicit relations between textual representations of entities. We will describe content based networks and compare their topological and temporal properties to those of explicit link based networks using several datasets from different domains.

Given a set of participants who have a set of posts, we can represent them using a content similarity based network. We define the salience scores of a participant recursively in terms of participants with similar discourse. This is based on the hypothesis that important or representative posts tend to be lexically similar to other important or representative posts. We also study how to take diversity into consideration when assigning salience to nodes. In addition, we study how initial independent quality priors may be incorporated in the proposed method.

We also proposed a technique for analyzing the temporal evolution of the salience of participants in a discussion. This method can dynamically track how the relative importance of speakers evolve over time using graph based techniques. To capture the dynamic nature of this process, the network connecting entities must evolve over time such that we have a different network at each instance of time that reflects the interaction of entities at this instant.

In the second part of this dissertation, we introduce content similarity networks that use linguistic analysis to find implicit relations between entities. We also proposed methods for identifying both static and dynamic salience in discussions. One of the unique characteristics of this work is that it mainly depends on identifying implicit relations from text.

1.4 Thesis Overview

Chapter II is the first chapter in the first part of this dissertation. This part is mainly concerned with analyzing multi-party dialogs and discussions to identify how participants show their attitude toward other participants and how this leads to or reveals rifts in social groups. The main goal of this part is building a signed network representation of discussions given the text. The first chapter in this part, Chapter II, looks into the problem of automatically identifying the polarity of words. This problem is a very important task in Natural Language Processing. It has applications in text classification, text filtering, analysis of product review, analysis of responses to surveys, and mining online discussions. In this chapter, we propose a method for identifying the polarity of words. We apply a Markov random walk model to a large word relatedness graph, producing a polarity estimate for any given word. A key advantage of the model is its ability to accurately and quickly assign a polarity sign and magnitude to any word. The method could be used both in a semi-supervised setting, where a training set of labeled words is used, and in an unsupervised setting, where a handful of seeds are used to define the two polarity classes. The method is experimentally tested using a manually labeled set of positive and negative words. It outperforms the state of the art methods in the semi-supervised setting. The results in the unsupervised setting are comparable to the best reported values. However, the proposed method is faster and does not need a large corpus. This chapter is based on the work published in [54, 52].

In Chapter III, we study the problem of identifying sentences that show an attitude in a discussion. We define an attitude as the mental position of one participant with regard to another participant. This problem lies under the area of mining sentiment from user generated content. We study threaded discussions which act as a very important tool for communication and collaboration in the Web. Threaded discussions include e-mails, e-mail lists, bulletin boards, newsgroups, and Internet forums. Most of the work on sentiment analysis has been centered on finding the sentiment toward products or topics. In this chapter, we present a method to identify the attitude of participants in an on-line discussion toward one another. This would enable us to build a signed network representation of participant interactions where every edge has a sign that indicates whether the interaction is positive or negative. This is different from most of the research on social networks that has focused almost exclusively on positive links. The method is experimentally tested using a manually labeled set of discussion posts. The results show that the proposed method is capable of identifying attitudinal sentences, and their signs, with high accuracy and that it outperforms several other baselines. This chapter is based on the work published as [53].

Chapter IV concludes the first part of this dissertation and describes how a signed social network can be extracted from text. Most of the research on social networks has almost exclusively focused on positive links between entities. There are much more insights that we may gain by considering the signed case where both positive and negative edges are considered. One of the reasons why signed social networks have received less attention is the lack of an explicit notion of negative relations in most social network applications. However, most such applications have text embedded in the social network. Applying linguistic analysis techniques to this text enables us to identify both positive and negative interactions. In this chapter, we propose a new method to automatically construct a signed social network from text. The resulting networks have a polarity associated with every edge. Edge polarity is a means for indicating a positive or a negative affinity between two individuals. We apply the proposed method to a large amount of on-line discussion posts. Experiments show that the proposed method is capable of constructing networks from text with high accuracy. We also connect our analysis to social psychology theories of signed networks, namely the structural balance theory. Finally, we show several applications that can benefit from this signed network representation of discussions. This chapter is based on the work in [51].

Chapter V starts the second part of this dissertation. In this chapter, we introduce the concept of content similarity based networks and compare them to link based networks. We study the different topological and temporal properties of the two networks. We show that the two networks bear a great deal of similarity in terms of their topological and temporal properties. We also study how correlated the two networks are and whether we can predict the edges of any of them using the other. Content similarity based networks are used to model influence between different participants in a discussion. We rely on those networks to find implicit connections between individuals that model how they were influenced by each other. We couple this with network analysis techniques for identifying the most authoritative or salient entities in the following chapters. This chapter is based on the in [55].

The following chapter, Chapter VI, presents a stochastic graph based method for assigning salience to participants in a discussion. Each participant is assigned a score that reflects how salient he/she is. Node scores are calculated recursively in terms of the scores of their neighbors in a lexical similarity graph. A random walk is performed on a graph where nodes represent participants and edges link participants with lexically similar content. Lexical similarity is measured using either the cosine similarity measure, or the Kullback-Leibler (KL) divergence. In addition, the proposed method combines lexical centrality with information novelty to reduce redundancy in the ranked list. Nodes similar to highly ranked nodes are discounted to make sure that diversity is maintained in the final rank. The presented method also allows us to include additional initial quality priors to assess the quality of the nodes, such text fluency measured by n-gram model probabilities.

We evaluate our approach using data from two large blog datasets. We measure the selection quality by the number of blogs covered in the network as calculated by an information diffusion model. We compare our method to other heuristic and greedy selection methods and show that it significantly outperforms them. This chapter is based on the work in [56, 53].

In Chapter VII, We introduce a technique for analyzing the temporal evolution of the salience of participants in a discussion. Our method can dynamically track how the relative importance of participants evolve over time using graph based techniques. Participant salience is computed based on the eigenvector centrality in a graph representation of participants in a discussion. Two participants in a discussion are linked with an edge if they use similar rhetoric. The method is dynamic in the sense that the graph evolves over time to capture the evolution inherent to the salience of participants. We used our method to track the salience of members of the US Senate using data from the US Congressional Record. Our analysis investigated how the salience of speakers changes over time. Our results show that the scores can capture speaker centrality in topics as well as events that result in change of salience or influence among different participants. This chapter is based on the work published as [50, 53]. Finally, we present conclusions and description of the future work, in Chapter VIII.

Part I. Mining Attitude from Text

CHAPTER II

Identifying Word Polarity

2.1 Introduction

The first part of this work focuses on using linguistic analysis to closely analyze attitude in multi-party discussions. We work on building a signed network representation of discussions given the text. In this type of networks, nodes represent participants and edges represent interaction between participants. Interaction between participants could be classified as either positive (agreement), negative (disagreements), or neutral (unknown). This kind of interactions could be represented by a signed network. Signed networks have polarity (+ or -) on each edge. Edge polarity is a means for indicating a positive or negative affinity between two individuals. As a first step toward this goal, we propose a novel method for identifying the polarity of words.

Identifying emotions and attitudes from unstructured text has a variety of possible applications. For example, there has been a great body of work for mining product reputation on the Web [105, 132]. Knowing the reputation of a product is very important for marketing and customer relation management [105]. Another interesting application is mining on-line discussions. A threaded discussion is an electronic discussion in which software tools are used to help individuals post messages and respond to other messages. Threaded discussions act as a very important tool for communication and collaboration in the Web.

An enormous number of discussion groups exist on the Web. Millions of users post content to these groups covering pretty much every possible topic. Tracking participant attitude toward different topics and toward other participants is a very interesting task. For example, Tong [130] presented the concept of sentiment timelines. His system classifies discussion posts about movies as either positive or negative. This is used to produce a plot of the number of positive and negative sentiment messages over time. All those applications could benefit much from an automatic way of identifying semantic orientation of words.

In this chapter, we study the problem of automatically identifying semantic orientation of any word by analyzing its relations to other words. Automatically classifying words as either positive or negative enables us to automatically identify the polarity of larger pieces of text. This could be a very useful building block for mining surveys, product reviews and on-line discussions. We apply a Markov random walk model to a large semantic word graph, producing a polarity estimate for any given word. Previous work on identifying the semantic orientation of words has addressed the problem as both a semi-supervised [128] and an unsupervised [131] learning problem. In the semi-supervised setting, a training set of labeled words is used to train the model. In the unsupervised setting, only a handful of seeds are used to define the two polarity classes. The proposed method could be used both in a semi-supervised and in an unsupervised setting. Empirical experiments on a labeled set of words show that the proposed method outperforms the state of the art methods in the semi-supervised setting. The results in the unsupervised setting are comparable to the best reported values. The proposed method has the advantages that it is faster and it does not need a large training corpus.

The rest of the chapter is structured as follows. In Section 2.2, we discuss related work. Section 2.3 presents our method for identifying word polarity. In Section 2.4, we describe how the proposed method can be extended to cover non English words. Section 2.5 describes our experimental setup. We conclude in Section 2.6.

2.2 Related Work

Hatzivassiloglou and McKeown [57] proposed a method for identifying word polarity of adjectives. They extract all conjunctions of adjectives from a given corpus and then they classify each conjunctive expression as either the same orientation such as “simple and well-received” or different orientation such as “simplistic but well-received”. The result is a graph that they cluster into two subsets of adjectives. They classify the cluster with the higher average frequency as positive. They created and labeled their own dataset for experiments. Their approach works only with adjectives because there is nothing wrong with conjunctions of nouns or verbs with opposite polarities (e.g., “war and peace”, “rise and fall”, etc.).

Turney and Littman [131] identify word polarity by looking at its statistical association with a set of positive/negative seed words. They use two statistical measures for estimating association: Pointwise Mutual Information (PMI) and Latent Semantic Analysis (LSA). To get co-occurrence statistics, they submit several queries to a search engine. Each query consists of the given word and one of the seed words. They use the search engine near operator to look for instances where the given word is physically close to the seed word in the returned document. They present their method as an unsupervised method where a very small amount of seed words are used to define semantic orientation rather than train the model. One of the limitations of their method is that it requires a large corpus of text to achieve good performance. They use several corpora; the size of the best performing dataset is roughly one hundred billion words [131].

Takamura et al. [128] proposed using spin models for extracting semantic orientation of words. They construct a network of words using gloss definitions, thesaurus, and co-occurrence statistics. They regard each word as an electron. Each electron has a spin and each spin has a direction taking one of two values: up or down. Two neighboring spins tend to have the same orientation from an energetic point of view. Their hypothesis is that as neighboring electrons tend to have the same spin direction, neighboring words tend to have similar polarity. They pose the problem as an optimization problem and use the mean field method to find the best solution. The analogy with electrons leads them to assume that each word should be either positive or negative. This assumption is not accurate because most of the words in the language do not have any semantic orientation. They report that their method could get misled by noise in the gloss definition and their computations sometimes get trapped in a local optimum because of its greedy optimization flavor.

Kamps et al. [67] construct a network based on WordNet synonyms and then use the shortest paths between any given word and the words “good” and “bad” to determine word polarity. They report that using shortest paths could be very noisy. For example, “good” and “bad” themselves are closely related in WordNet with a 5-long sequence “good, sound, heavy, big, bad”. A given word w may be more connected to one set of words (e.g., positive words); yet have a shorter path connecting it to one word in the other set. Restricting seed words to only two words affects their accuracy. Adding more seed words could help but it will make their method extremely costly from the computation point of view. They evaluate their method only using adjectives.

Hu and Liu [60] use WordNet synonyms and antonyms to predict the polarity of words. For any word, whose polarity is unknown, they search WordNet and a list of seed labeled words to predict its polarity. They check if any of the synonyms of the given word has known polarity. If so, they label it with the label of its synonym. Otherwise, they check if any of the antonyms of the given word has known polarity. If so, they label it with the opposite label of the antonym. They continue in a bootstrapping manner till they label all possible word. This method is quite similar to the shortest-path method proposed in [67].

There are some other methods that try to build lexicons of polarized words. Esuli and Sebastiani [41, 42] use a textual representation of words by collating all the glosses of the word as found in some dictionary. Then, a binary text classifier is trained using the textual representation and applied to new words. Kim and Hovy [71] start with two lists of positive and negative seed words. WordNet is used to expand these lists. Synonyms of positive words and antonyms of negative words are considered positive, while synonyms of negative words and antonyms of positive words are considered negative. A similar method is presented in [7], where WordNet synonyms, antonyms, and glosses are used to iteratively expand a list of seeds. The sentiment classes are treated as fuzzy categories where some words are very central to one category, while others may be interpreted differently. Kanayama and Nasukawa [68] use syntactic features and context coherency, the tendency for same polarities to appear successively, to acquire polar atoms.

Other related work is concerned with subjectivity analysis. Subjectivity analysis is the task of identifying text that present opinions as opposed to objective text that present factual information [138]. Text could be either words, phrases, sentences, or any other chunks. There are two main categories of work on subjectivity analysis. In the first category, subjective words and phrases are identified without considering their context [138, 58, 9]. In the second category, the context of subjective text is used [119, 143, 107, 113] Wiebe et al. [139] lists a lot of applications of subjectivity analysis such as classifying emails and mining reviews. Subjectivity analysis is related to the proposed method because identifying the polarity of text is the natural next step that should follow identifying subjective text.

2.3 Word Polarity

We use a Markov random walk model to identify polarity of words. Assume that we have a network of words, some of which are labeled as either positive or negative. In this network, two words are connected if they are related. Different sources of information could be used to decide whether two words are related or not. For example, the synonyms of any word are semantically related to it. The intuition behind connecting semantically related words is that those words tend to have similar polarity. Now imagine a random surfer walking along the network starting from an unlabeled word w. The random walk continues until the surfer hits a labeled word. If the word w is positive then the probability that the random walk hits a positive word is higher and if w is negative then the probability that the random walk hits a negative word is higher. Similarly, if the word w is positive then the average time it takes a random walk starting at w to hit a positive node is less than the average time it takes a random walk starting at w to hit a negative node.

In the rest of this section, we will describe how we can construct a word relatedness graph in Section 2.3.1. The random walk model is described in Section 2.3.2. Hitting time is defined in Section 2.3.3. Finally, an algorithm for computing a sign and magnitude for the polarity of any given word is described in Section 2.3.4.

2.3.1 Network Construction

We construct a network where two nodes are linked if they are semantically related. Several sources of information could be used as indicators of the relatedness of words. One such important source is WordNet [101]. WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept [101]. Synsets are interlinked by means of conceptual-semantic and lexical relations.

The simplest approach is to connect words that occur in the same WordNet synset. We can collect all words in WordNet, and add links between any two words that occur in the same synset. The resulting graph is a graph $G(W, E)$ where W is a set of word / part-of-speech pairs for all the words in WordNet. E is the set of edges connecting each pair of synonymous words. Nodes represent word/pos pairs rather than words because the part of speech tags are helpful in disambiguating the different senses for a given word. For example, the word "fine" has two different meanings, with two opposite polarity, when used as an adjective and as a noun.

Several other methods could be used to link words. For example, we can use other WordNet relations: hypernyms, similar to, etc. Another source of links between words is co-occurrence statistics from a corpus. Following the method presented in [57], we can connect words if they appear in a conjunctive form in the corpus. This method is only applicable to adjectives. If two adjectives are connected by "and" in conjunctive form, it is highly likely that they have the same semantic orientation. In all our experiments, we restricted the network to only WordNet relations. We study the effect of using co-occurrence statistics to connect words later at the end of our experiments. If more than one relation exists between any two words, the strength of the corresponding edge is adjusted accordingly.

2.3.2 Random Walk Model

Imagine a random surfer walking along the word relatedness graph G. Starting from a word with unknown polarity i, it moves to a node j with probability P_{ij} after the first step. The walk continues until the surfer hits a word with a known polarity. Seed words with known polarity act as an absorbing boundary for the random walk. If we repeat the number of random walks N times, the percentage of time at which the walk ends at a positive/negative word could be used as an indicator of its positive/negative polarity. The average time a random walk starting at w takes to hit the set of positive/negative nodes is also an indicator of its polarity. This view is closely related to the partially labeled classification with random walks approach in [127] and the semi-supervised learning using harmonic functions approach in [144].

Let W be the set of words in our lexicon. We construct a graph whose nodes V are all words in W. Edges E correspond to relatedness between words. We define transition probabilities $P_{t+1|t}(j|i)$ from i to j by normalizing the weights of the edges out of node i, so:

$$P_{t+1|t}(j|i) = Wij / \sum_k W_{ik} \tag{2.1}$$

where k represents all nodes in the neighborhood of i. $P_{t+1|t}(j|i)$ denotes the transition probability from node i at step t to node j at time step $t+1$. We note that the weights W_{ij} are symmetric and the transition probabilities $P_{t+1|t}(j|i)$ are not necessarily symmetric because of the node out degree normalization.

2.3.3 First-Passage Time

The mean first-passage (hitting) time $h(i|k)$ is defined as the average number of steps a random walker, starting in state $i \neq k$, will take to enter state k for the first time [110]. Let $G = (V, E)$ be a graph with a set of vertices V, and a set of edges E. Consider a subset of vertices $S \subset V$. Consider a random walk on G starting at node $i \notin S$. Let N_t denote the position of the random surfer at time t. Let $h(i|S)$ be the average number of steps a random walker, starting in state $i \notin S$, will take to enter a state $k \in S$ for the first time. Let T_S be the first-passage for any vertex in S.

$$\begin{aligned} & P(T_S = t | N_0 = i) = \\ & \sum_{j \in V} p_{ij} \times P(T_S = t - 1 | N_0 = j) \end{aligned} \tag{2.2}$$

$h(i|S)$ is the expectation of T_S. Hence:

$$\begin{aligned}
h(i|S) &= E(T_S|N_0 = i) \\
&= \sum_{t=1}^{\infty} t \times P(T_S = t|N_0 = i) \\
&= \sum_{t=1}^{\infty} t \sum_{j \in V} p_{ij} P(T_S = t-1|N_0 = j) \\
&= \sum_{j \in V} \sum_{t=1}^{\infty} (t-1) p_{ij} P(T_S = t-1|N_0 = j) \\
&+ \sum_{j \in V} \sum_{t=1}^{\infty} p_{ij} P(T_S = t-1|N_0 = j) \\
&= \sum_{j \in V} p_{ij} \sum_{t=1}^{\infty} t P(T_S = t|N_0 = j) + 1 \\
&= \sum_{j \in V} p_{ij} \times h(j|S) + 1 \qquad (2.3)
\end{aligned}$$

Hence the first-passage (hitting) time can be formally defined as:

$$h(i|S) = \begin{cases} 0 & i \in S \\ \sum_{j \in V} p_{ij} \times h(j|S) + 1 & \text{otherwise} \end{cases} \qquad (2.4)$$

2.3.4 Word Polarity Calculation

Based on the description of the random walk model and the first-passage (hitting) time above, we now propose our word polarity identification algorithm. We begin by constructing a word relatedness graph and defining a random walk on that graph as described above. Let S^+ and S^- be two sets of vertices representing seed words that are already labeled as either positive or negative respectively. For any given word w, we compute the hitting time $h(w|S^+)$, and $h(w|S^-)$ for the two sets iteratively as described earlier. If $h(w|S^+)$ is greater than $h(w|S^-)$, the word is classified as negative, otherwise it is classified as positive. The ratio between the two hitting times could be used as an indication of how positive/negative the given word is. This is useful in case we need to provide a confidence measure for the prediction. This could be used to allow the model to abstain from classifying words when the confidence level is low.

Computing hitting time as described earlier may be time consuming especially if the graph is large. To overcome this problem, we propose a Monte Carlo based algorithm for estimating it. The algorithm is shown in Algorithm 1.

Algorithm 1 Word Polarity using Random Walks

```
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 $h^*(w|S^+) \leq h^*(w|S^-)$ then
    Classify $w$ as positive
else
    Classify $w$ as negative
end if
```

2.4 Foreign Word Polarity

As we mentioned earlier in this chapter, a great body of research work, including ours, has focused on identifying the semantic orientation of words. This work has almost exclusively focused on English. Most of this work used several language dependent resources. For example Turney and Littman [131] use the entire English Web corpus by submitting queries consisting of the given word and a set of seeds to a search engine. In addition, several other methods have used WordNet [101] for connecting semantically related words [67, 128, 54].

When we try to apply those methods to other languages, we run into the problem of the lack of resources in other languages when compared to English. For example, the General Inquirer lexicon [126] has thousands of English words labeled with semantic orientation.

Most of the literature has used it as a source of labeled seeds or for evaluation. Such lexicons are not readily available in other languages. Another source that has been widely used for this task is WordNet [101]. Even though other WordNets have been built for other languages, their coverage is very limited when compared to the English WordNet.

In this section, we show how we can extend the methods presented in this chapter to predict the semantic orientation of foreign words. The proposed method is based on creating a multilingual network of words that represents both English and foreign words. The network has English-English connections, as well as foreign-foreign connections and English-foreign connections. This allows us to benefit from the richness of the resources built for the English language and in the mean time utilize resources specific to foreign languages. We then define a random walk model over the multilingual network and predict the semantic orientation of any given word by comparing the mean hitting time of a random walk starting from it to a positive and a negative set of seed English words.

2.4.1 Multilingual Word Network

We build a network $G(V, E)$ where $V = V_{en} \cup V_{fr}$ is the union of a set of English and foreign words. E is a set of edges connecting nodes in V. There are three types of connections: English-English connections, Foreign-Foreign connections, and English-Foreign connections.

For the English-English connections, we use the same methodologies we described earlier in this chapter. Foreign-Foreign connections are created in a similar way to the English connections. Some foreign languages have lexical resources based on the design of the Princeton English WordNet. For example: Euro WordNet (EWN) [134], Arabic WordNet (AWN) [35, 11, 34], and the Hindi WordNet [106, 63]. We also use co-occurrence statistics similar to the work of Hatzivassiloglou and McKeown [57]. Finally, to connect foreign words to English words, we use a foreign to English dictionary. For every word in a list of foreign words, we look up its meaning in a dictionary and add an edge between the foreign word and every other English word that appeared as a possible meaning for it.

2.4.2 Foreign Word Semantic Orientation Prediction

We use the multilingual network we described above to predict the semantic orientation of words based on the mean hitting time to two sets of positive and negative seeds. Given two lists of seed English words with known polarity, we define two sets of nodes S^+ and S^- representing those seeds. For any given word w, we calculate the mean hitting time between w, and the two seed sets $h(w|S^+)$, and $h(w|S^-)$. If $h(w|S^+)$ is greater than $h(w|S^-)$, the word is classified as negative, otherwise it is classified as positive. We used the list of labeled seeds from [57] and [126].

2.5 Experiments

We performed experiments on the General Inquirer lexicon [126]. We used it as a gold standard data set for positive/negative words. The dataset contains 4206 words, 1915 of which are positive and 2291 are negative. Some of the ambiguous words were removed like [132, 128]. Some examples of positive/negative words are listed in Table 2.1.

We use WordNet [101] as a source of synonyms and hypernyms for the word relatedness graph. We used the Reuters Corpus, Volume 1 [86] o generate co-occurrence statistics in the experiments that used them. We used 10-fold cross validation for all tests. We evaluate our results in terms of accuracy. Statistical significance was tested using a 2-tailed paired t-test. All reported results are statistically significant at the 0.05 level. We perform experiments varying the parameters and the network. We also look at the performance of the proposed method for different parts of speech, and for different confidence levels We compare our method to the Semantic Orientation from PMI (SO-PMI) method described in [132], the Spin model (Spin) described in [128], the shortest path (short-path) described

Table 2.1: Examples of positive and negative words

Positive		Negative	
able	adjective	abandon	verb
acceptable	adjective	abuse	verb
admire	verb	burglar	noun
amazing	adjective	chaos	noun
careful	adjective	contagious	adjective
ease	noun	corruption	noun
guide	verb	lie	verb
inspire	verb	reluctant	adjective
truthful	adjective	wrong	adjective

in [67], and the bootstrapping (bootstrap) method described in [60].

2.5.1 Comparisons with other methods

This method could be used in a semi-supervised setting where a set of labeled words are used and the system learns from these labeled nodes and from other unlabeled nodes. Under this setting, we compare our method to the spin model described in [128]. Table 2.2 compares the performance using 10-fold cross validation. The table shows that the proposed method outperforms the spin model. The spin model approach uses word glosses, WordNet synonym, hypernym, and antonym relations, in addition to co-occurrence statistics extracted from corpus. The proposed method achieves better performance by only using WordNet synonym, hypernym and similar to relations. Adding co-occurrence statistics slightly improved performance, while using glosses did not help at all.

We also compare our method to the SO-PMI method presented in [132]. Turney and Littman [132] propose two methods for predicting the semantic orientation of words. They use Latent Semantic Analysis (SO-LSA) and Pointwise Mutual Information (SO-PMI) for measuring the statistical association between any given word and a set of 14 seed words. They describe this method as unsupervised because they only use 14 seeds as paradigm words that define the semantic orientation rather than train the model [132]. After [132], we use our method to predict semantic orientation of words in the General Inquirer lexicon [126] using only 14 seed words. The network we used contains only WordNet relations. No glosses or co-occurrence statistics are used. The results comparing the SO-PMI method with different dataset sizes, the spin model, and the proposed method using only 14 seeds is shown in Table 2.2. We notice that the random walk method outperforms SO-PMI when SO-PMI uses datasets of sizes 1×10^7 and 2×10^9 words. The performance of SO-PMI and the random walk methods are comparable when SO-PMI uses a very large dataset (1×10^{11} words). The performance of the spin model approach is also comparable to the other 2 methods. The advantages of the random walk method over SO-PMI is that it is faster and it does not need a very large corpus like the one used by SO-PMI. Another advantage is that the random walk method can be used along with the labeled data from the General Inquirer lexicon [126] to get much better performance. This is costly for the SO-PMI method because that will require the submission of almost 4000 queries to a commercial search engine.

We also compare our method to the bootstrapping method described in [60], and the shortest path method described in [67]. We build a network using only WordNet synonyms and hypernyms. We restrict the test set to the set of adjectives in the General Inquirer lexicon [126] because this method is mainly interested in classifying adjectives. The performance of the spin model method, the bootstrapping method, the shortest path method, and the random walk method for only adjectives is shown in Table 2.3. We notice from the table that the random walk method outperforms the spin model, the bootstrapping method, and the shortest path method for adjectives. The reported accuracy for the shortest path method only considers the words it could assign a non-zero orientation value. If we consider all words, the accuracy will drop to around 61%.

Table 2.2: Accuracy for SO-PMI with different dataset sizes, the spin model, and the random walks model for 10-fold cross validation and 14 seeds.

-	CV	14 seeds
SO-PMI (1×10^7)	-	61.3
SO-PMI (2×10^9)	-	76.1
SO-PMI (1×10^{11})	-	**82.8**
Spin Model	91.5	81.9
Random Walks	**93.1**	82.1

Table 2.3: Accuracy for adjectives only for the spin model, the bootstrap method, and the random walk model.

Method	Spin Model	Bootstrap	Shortest Path	Random Walks
Accuracy	83.6	72.8	68.8	88.8

Varying Parameters

As we mentioned in Section 2.3.4, we use a parameter m to put an upper bound on the length of random walks. In this section, we explore the impact of this parameter on our method's performance.

Figure 2.1 shows the accuracy of the random walk method as a function of the maximum number of steps m. m varies from 5 to 50. We use a network built from WordNet synonyms and hypernyms only. The number of samples k was set to 1000. We perform 10-fold cross validation using the General Inquirer lexicon. We notice that the maximum number of steps m has very little impact on performance until it rises above 30. When it does, the performance drops by no more than 1%, and then it does not change anymore as m increases. An interesting observation is that the proposed method performs quite well with a very small number of steps (around 10). We looked at the dataset to understand why increasing the number of steps beyond 30 negatively affects performance. We found out that when the number of steps is very large, compared to the diameter of the graph, the random walk that starts at ambiguous words, which are hard to classify, have the chance

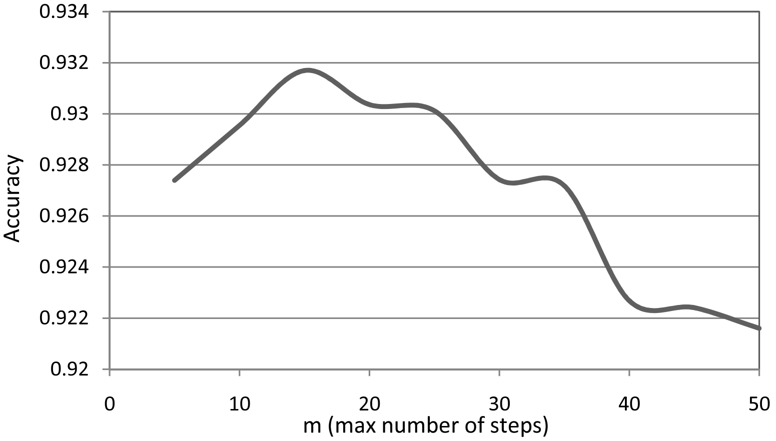


Figure 2.1: The effect of varying the maximum number of steps (m) on accuracy.

of moving till it hits a node in the opposite class. That does not happen when the limit on the number of steps is smaller because those walks are then terminated without hitting any labeled nodes and hence ignored.

Next, we study the effect of the number of samples k on our method's performance. As explained in Section 2.3.4, k is the number of samples used by the Monte Carlo algorithm to find an estimate for the hitting time. Figure 2.2 shows the accuracy of the random walks method as a function of the number of samples k. We use the same settings as in the previous experiment. The only difference is that we fix m at 15 and vary k from 10 to 20000 (note the logarithmic scale). We notice that the performance is badly affected, when the value of k is very small (less than 100). We also notice that after 1000, varying k has very little, if any, effect on performance. This shows that the Monte Carlo algorithm for computing the random walks hitting time performs quite well with values of the number of samples as small as 1000.

The preceding experiments suggest that the parameter have very little impact on performance. This suggests that the approach is fairly robust (i.e., it is quite insensitive to different parameter settings).

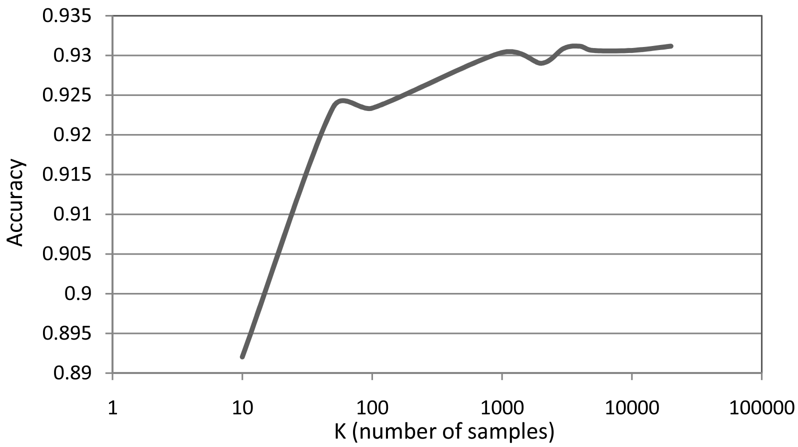


Figure 2.2: The effect of varying the number of samples (k) on accuracy.

Other Experiments

We now measure the performance of the proposed method when the system is allowed to abstain from classifying the words for which it has low confidence. We regard the ratio between the hitting time to positive words and hitting time to negative words as a confidence measure and evaluate the top words with the highest confidence level at different values of threshold. Figure 2.4 shows the accuracy for 10-fold cross validation and for using only 14 seeds at different thresholds. We notice that the accuracy improves by abstaining from classifying the difficult words. The figure shows that the top 60% words are classified with accuracy greater than 99% for 10-fold cross validation and 92% with 14 seed words. This may be compared to the work described in [128] where they achieve the 92% level when they only consider the top 1000 words (28%).

Figure 2.3 shows a learning curve displaying how the performance of the proposed method is affected with varying the labeled set size (i.e., the number of seeds). We notice that the accuracy exceeds 90% when the training set size rises above 20%. The accuracy steadily increases as the size of labeled data increases.

We also looked at the classification accuracy for different parts of speech in Figure 2.5.

We notice that, in the case of 10-fold cross validation, the performance is consistent across parts of speech. However, when we only use 14 seeds all of which are adjectives, similar to [131], we notice that the performance on adjectives is much better than other parts of speech. When we use 14 seeds but replace some of the adjectives with verbs and nouns like (love, harm, friend, enemy), the performance for nouns and verbs improves considerably at the cost of a small decay in the performance on adjectives. Finally, we tried adding edges to the network from glosses and co-occurrence statistics but we did not get any statistically significant improvement. Some of the words that were very weakly linked benefited from adding new types of links and they were correctly predicted. Others were misled by the noise and were incorrectly classified. We had a closer look at the results to find out what are the reasons behind incorrect predictions. We found two main reasons. First, some words are ambiguous and have more than one sense, possibly with different semantic orientations. Disambiguating the sense of words given their context before trying to predict their polarity should solve this problem. The second reason is that some words have very few connections in thesaurus. A possible solution to this might be identifying those words and adding more links to them from glosses of co-occurrence statistics in corpus.

2.5.2 Foreign Words

In addition to the English data we described earlier, we constructed a labeled set of 300 Arabic and 300 Hindi words for evaluation. For every language, we asked two native speakers to examine a large amount of text and identify a set of positive and negative words. We also used an Arabic-English and a Hindi-English dictionaries to generate Foreign-English links.

We compare our results to two baselines. The first is the $\mathrm{SO\text{-}PMI}$ method described in [131]. This method is based on finding the semantic association of any given word to a

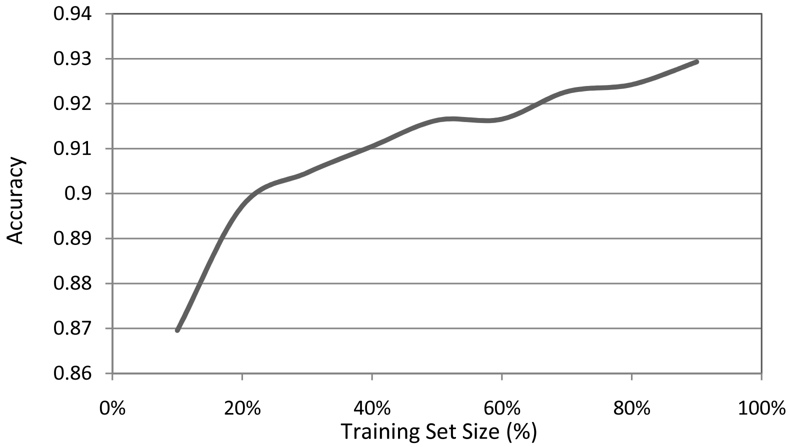


Figure 2.3: The effect of varying the number of seeds on accuracy.

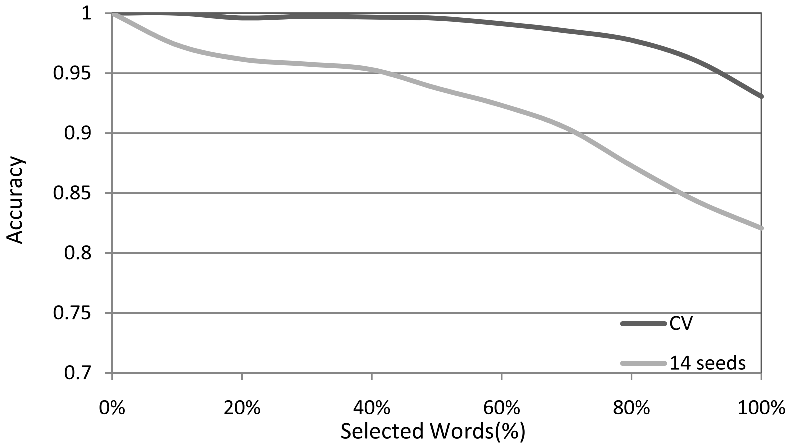


Figure 2.4: Accuracy for words with high confidence measure.

set of positive and a set of negative words. It can be calculated as follows:

$$\text{SO-PMI}(w) = \log \frac{hits_{w,pos} \times hits_{neg}}{hits_{w,neg} \times hits_{pos}} \tag{2.5}$$

where w is a word with unknown polarity, $hits_{w,pos}$ is the number of hits returned by a commercial search engine when the search query is the given word and the disjunction of all positive seed words. $hits_{pos}$ is the number of hits when we search for the disjunction of all positive seed words. $hits_{w,neg}$, and $hits_{neg}$ are defined similarly. We used 7 positive and 7 negative seeds as described in [131].

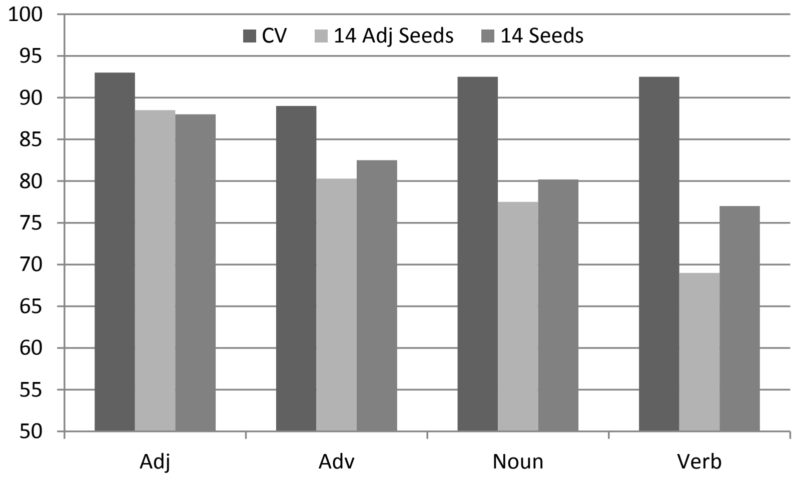


Figure 2.5: Accuracy for different parts of speech.

The second baseline constructs a network of foreign words only as described earlier. It uses mean hitting time to find the semantic association of any given word. We used 10 fold cross validation for this experiment. We will refer to this system as $\mathrm{HT\text{-}FR}$.

Finally, we build a multilingual network and use the hitting time as before to predict semantic orientation. We used the English words from [126] as seeds and the labeled foreign words for evaluation. We will refer to this system as $\mathrm{HT\text{-}FR\text{-}EN}$.

Figure 2.6 compares the accuracy of the three methods for Arabic and Hindi. We notice that the $\mathrm{SO\text{-}PMI}$ and the hitting time based methods perform poorly on both Arabic and Hindi. This is clearly evident when we consider that the accuracy of the two systems on English was 83%, and 93% respectively [131, 54]. This supports our hypothesis that state of the art methods, designed for English, perform poorly on foreign languages due to the limited amount of resources available in foreign languages compared to English. The figure also shows that the proposed method, which combines resources from both English and foreign languages, performs significantly better. Finally, we studied how much improvement is achieved by including links between foreign words from global WordNets. We found out that it improves the performance by 2.5%, and 4% for Arabic

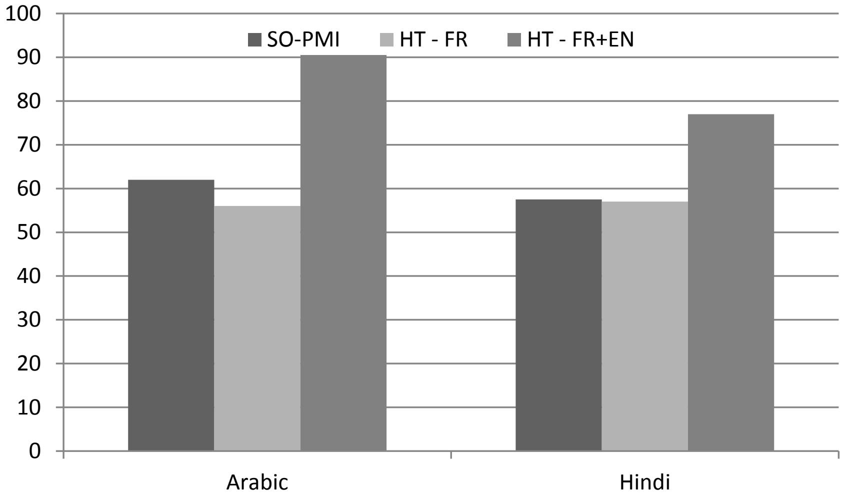


Figure 2.6: An example showing a signed social network along with evidence from text that justifies edge signs.

and Hindi respectively.

2.6 Conclusions

Predicting the semantic orientation of words is a very interesting task in Natural Language Processing and it has a wide variety of applications. We proposed a method for automatically predicting the semantic orientation of words using random walks and hitting time. The proposed method is based on the observation that a random walk starting at a given word is more likely to hit another word with the same semantic orientation before hitting a word with a different semantic orientation. The proposed method can be used in a semi-supervised setting, where a training set of labeled words is used, and in an unsupervised setting, where only a handful of seeds is used to define the two polarity classes. We predict semantic orientation with high accuracy. The proposed method is fast, simple to implement, and does not need any corpus. We also extended the proposed method to cover the problem of predicting the semantic orientation of foreign words. All previous work on this task has almost exclusively focused on English. Applying off-the-shelf methods developed for English to other languages does not work well because of the limited amount of resources available in foreign languages compared to English. We show that the proposed method can predict the semantic orientation of foreign words with high accuracy and outperforms state of the art methods limited to using language specific resources.

CHAPTER III

Identifying Sentences with Attitude

3.1 Introduction

Mining sentiment from text has a wide range of applications from mining product reviews on the Web [105, 131] to analyzing political speeches [129]. Automatic methods for sentiment mining are very important because manual extraction of sentiment is very costly, and inefficient. A new application of sentiment mining is to automatically identify attitudes between participants in an on-line discussion. An automatic tool to identify attitudes will enable us to build a signed network representation of participant interaction in which the interaction between two participants is represented using a positive or a negative edge. Even though using signed edges in social network studies is clearly important, most of the social networks research has focused almost exclusively on positive links between entities. Some work has recently investigated signed networks [84, 77], however this work was limited to a small number of datasets in which users were allowed to explicitly add negative, as well as positive, relations.

In this chapter, we study the problem of identifying sentences with attitude in on-line discussions. Although similar, identifying sentences that display an attitude in discussions is different from identifying opinionated sentences. A sentence in a discussion may bear opinions about a definite target (e.g., price of a camera) and yet have no attitude toward the other participants in the discussion. For instance, in the following discussion Alice's sentence has her opinion against something, yet no attitude toward the recipient of the sentence, Bob.

Alice: "You know what, he turned out to be a great disappointment"

Bob: "You are completely unqualified to judge this great person"

However, Bob shows strong attitude toward Alice. In this work, we look at ways to predict whether a sentence displays an attitude toward the text recipient. An attitude is the mental position of one participant with regard to another participant. It could be either positive or negative. We consider features which take into account the entire structure of sentences at different levels or generalization. Those features include lexical items, part-of-speech tags, and dependency relations.

3.2 Related Work

Identifying the polarity of individual words is a well studied problem. In previous work, Hatzivassiloglou and McKeown [57] proposed a method to identify the polarity of adjectives. In other work, Turney and Littman [131] use statistical measures to find the association between a given word and a set of positive/negative seed words. In more recent work, Takamura et al. [128] used the spin model to extract word semantic orientation. In our previous work [54], we uses a random walk model defined over a word relatedness graph to classify words as either positive or negative. Previous work has also used WordNet, a lexical database of English, to identify word polarity [60, 67, 7, 7]. A more detailed survey of this line of work can be found in Chapter II.

All the work mentioned above focuses on the task of identifying the polarity of individual words. Our proposed work is identifying attitudes in sentences that appear in online discussions. Perhaps the most similar work to ours is the prior work on subjectivity analysis, which is to identify text that presents opinions as opposed to objective text that presents factual information [138]. Prior work on subjectivity analysis mainly consists of two main categories: The first category is concerned with identifying the subjectivity of individual phrases and words regardless of the sentence and context they appear in [138, 58, 9]. In the second category, subjectivity of a phrase or word is analyzed within its context [119, 143, 107, 113]. Somasundaran et al. [123] developed genre specific lexicons using interesting function word combinations for detecting opinions in meetings. A very good study of the applications of subjectivity analysis from review mining to email classification is given in [138]. Despite similarities, our work is different from subjectivity analysis because the later only discriminates between opinions and facts. A discussion sentence may display an opinion about some topic yet no attitude. The language constituents considered in opinion detection may be different from those used to detect attitude. Moreover, extracting attitudes from on-line discussions is different from targeting subjective expressions [65, 71]. The later usually has a limited set of targets that compete for the subjective expressions (for example in a movie review, targets could be: director, actors, plot, and so forth). We cannot use similar methods because we are working on an open domain where anything could be a target. A very detailed survey that covers techniques and approaches in sentiment analysis and opinion mining could be found in [112].

Our work is also related to some of the work on mining on-line forums and discussions. Lin et al. [87] proposed a sparse coding-based method that simultaneously models semantics and structure of threaded discussions. Shen et al. [120] proposed three clustering methods for exploiting the temporal information in the streams, as well as an algorithm based on linguistic features to analyze the discourse structure information. Huang et al. [61] used an SVM classifier to extract (thread-title, reply) pairs as chat knowledge from on-line discussion forums to support the construction of a chatbot for a certain domain.

Other work has focused on the structure of questions and question-answer pairs in on-line forums and discussions [28, 24].

3.3 Problem Definition

Assume we have a set of sentences exchanged between participants in an on-line discussion. Our objective is to identify sentences that display an attitude from the text writer to the text recipient from those that do not. An attitude is the mental position of one participant with regard to another participant. An attitude may not be directly observable, but rather inferred from what participants say to one another. The attitude could be either positive or negative. Strategies for showing a positive attitude may include agreement, and praise, while strategies for showing a negative attitude may include disagreement, insults, and negative slang. After identifying sentences that display an attitude, we also predict the sign (positive or negative) of that attitude.

3.4 Representation and Preprocessing

In this section, we describe how sentences are represented and preprocessed. As we are only interested in attitudes between participants, we limit our study to sentences that use second person pronouns. Second person pronouns are usually used in a conversational genre to indicate that the text writer is addressing the text recipient. After identifying those sentences, we do some preprocessing to extract the most relevant fragments. We examine these fragments to identify the polarity of every word in the sentence. Every word could be assigned a semantic orientation. The semantic orientation could be either positive, negative, or neutral. The existence of polarized words in any sentence is an important indicator of whether it carries an attitude or not. We also extract several patterns at different levels of generalization representing any given sentence.

3.4.1 Word Polarity Identification

Identifying the polarity of words is an important step for our method. Our word identification module is based on the work we presented in Chapter II. We construct a graph where each node represents a word/part-of-speech pair. Two nodes are linked if the words are related. We use WordNet [101] to link related words based on synonyms, hypernyms, and similar to relations. For words that do not appear in Wordnet, we used Wiktionary, a collaboratively constructed dictionary. We also add some links based on co-occurrence statistics between words as from a large corpus. The resulting graph is a graph $G(W, E)$ where W is a set of word/part-of-speech pairs, and E is the set of edges connecting related words.

We define a random walk model on the graph and use the mean hitting time from any given word to a positive set and a negative set of seeds to predict whether a word is positive, negative, or neutral. We also use the method described in [141] to determine the contextual polarity of the identified words. The set of features used to predict contextual polarity includes word, sentence, polarity, structure features.

3.4.2 Identifying Relevant Parts of Sentences

The writing style in on-line discussion forums is very informal. Some of the sentences are very long, and punctuation marks are not always properly used. To solve this problem, we decided to use the grammatical structure of a sentence to identify the most relevant part of sentences that would be the subject of further analysis. Figure 3.1 shows a parse tree representing the grammatical structure of a particular sentence. If we closely examine the sentence, we will notice that we are only interested in a part of the sentence that includes the second person pronoun “you”. We extract this part, by starting at the word of interest , in this case “you”, and going up in the hierarchy till we hit the first sentence clause.

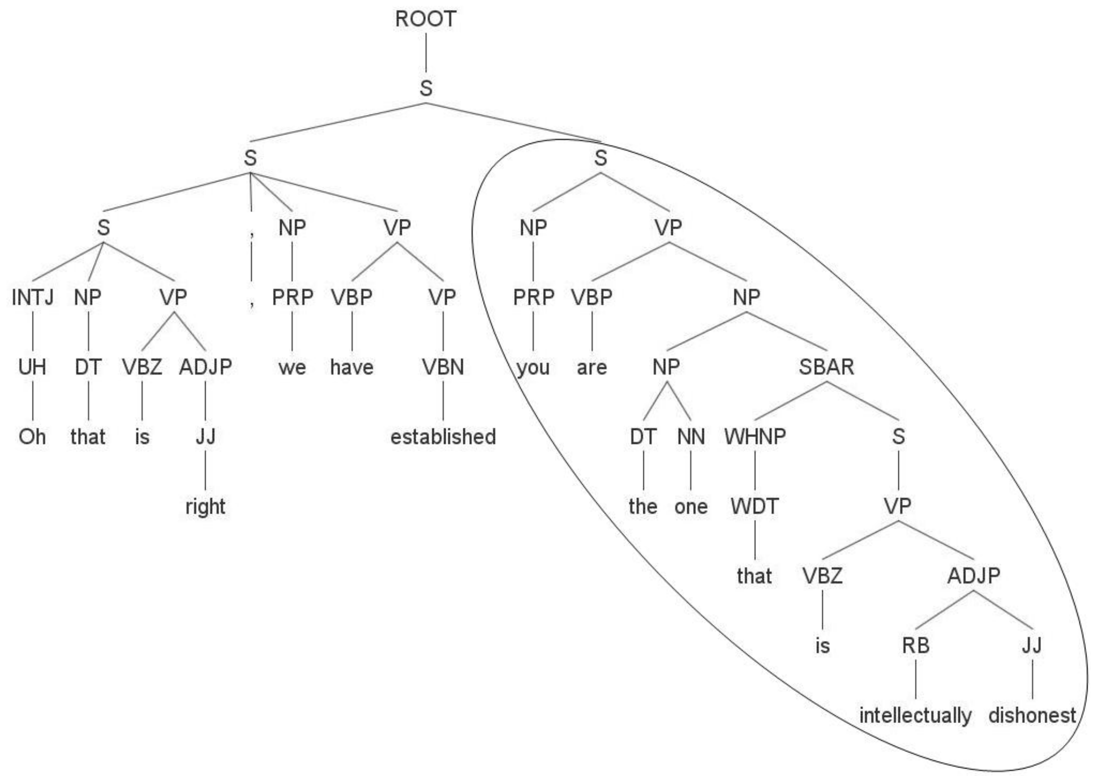


Figure 3.1: An example showing how to identify the relevant part of a sentence.

Once, we reach a sentence clause, we extract the corresponding text if it is grammatical, otherwise we go up one more level to the closest sentence clause. We used the Stanford parser to generate the grammatical structure of sentences [72].

3.4.3 Sentences as Patterns

The fragments we extracted earlier are more relevant to our task and are more suitable for further analysis. However, these fragments are completely lexicalized and consequently the performance of any analysis based on them will be limited by data sparseness. We can alleviate this by using more general representations of words. Those general representations can be used a long with words to generate a set of patterns that represent each fragment. Each pattern consists of a sequence of tokens. Such patterns could use lexical items, part-of-speech (POS) tags, word polarity tags, and dependency relations.

Table 3.1: Tags used for building patterns for representing sentences to identify attitude.

Type	Example
Text	That makes your claims so ignorant
POS	That/DT makes/VBZ your/PRP$ claims/NNS so/RB ignorant/JJ
Polarity	That/O makes/O your/O claims/O so/O ignorant/NEG
Dependency	your $\overset{\text{poss}}{\rightarrow}$ claims $\overset{\text{nsubj}}{\rightarrow}$ ignorant

We use three different patterns to represent each fragment:

- Lexical patterns: All polarized words are replaced with the corresponding polarity tag, and all other words are left as is.
- Part-of-speech patterns: All words are replaced with their POS tags. Second person pronouns are left as is. Polarized words are replaced with their polarity tags and their POS tags.
- Dependency grammar patterns: the shortest path connecting every second person pronoun to the closest polarized word in a dependency tree is extracted. The second person pronoun, the polarized word tag, and the types of the dependency relations along the path connecting them are used as a pattern. 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 [18]. Every polarized word is assigned to the closest second person pronoun in the dependency tree. This is only useful for sentences that have polarized words.

Table 3.1 shows the different kinds of representations for a particular sentence. We use text, part-of-speech tags, polarity tags, and dependency relations. The corresponding patterns for this sentence are shown in Table 3.2.

Table 3.2: Sample patterns for representing sentences to identify attitude.

Type	Example
Lexical pattern	That makes your claims so NEG
POS pattern	DT VBZ your_PRP$ NNS RB NEG_JJ
Dependency pattern	your poss nsubj NEG

3.5 Identifying Sentences with Attitude

In this section, we describe two models that predict whether a particular sentence shows an attitude or not. The first is based on building two Markov models trained on two sets of sentences, one with attitude and one without. The second is based on training a classifier using the existence of unigrams and bigrams from the lexical, part-of-speech, and dependency representations as features.

3.5.1 Markov Chains based Model

Given a set of patterns representing a set of sentences, we can build a graph $G = (V, E, w)$ where V is the set of all possible tokens that may appear in the patterns. $E = V \times V$ is the set of possible transitions between any two tokens. $w : E \rightarrow [0..1]$ is a weighting function that assigns to every pair of states (i, j) a weight $w(i, j)$ representing the probability that we have a transition from state i to state j.

This graph corresponds to a Markovian model. The set of states are the vocabulary, and the transition probabilities between states are estimated using Maximum Likelihood estimation as follows:

$$P_{ij} = \frac{N_{ij}}{N_i}$$

where N_{ij} is the number of times we saw i followed by j, and N_i is the total number of times we saw i in the training data. This is similar to building a language model over the language of the patterns. We use smoothing to account for unseen transitions.

We build two such models for every kind of patterns. The first model is built using all sentences that appeared in the training dataset and was labeled as having an attitude, and the second model is built using all sentences in the training dataset that do not have an attitude. If we have n kinds of patterns, we will build one such pair for every kind of patterns. Hence, we will end up with $2n$ models.

We split our training data into two splits; the first containing all sentences that have an attitude and the second containing all sentences that do not have an attitude. Given the methodology described in the previous section, we build n pairs of Markov models. Given any sentence, we extract the corresponding patterns and estimate the log likelihood that this sequence of tokens was generated from every model.

Given a model M, and sequence of tokens $T = (T_1, T_2, \ldots TS_n)$, the probability of this token sequence being generated from M is:

$$P_M(T) = \prod_{i=2}^{n} P(T_i|T_1, \ldots, T_{i-1}) = \prod_{i=2}^{n} W(T_{i-1}, T_i)$$

where n is the number of tokens in the pattern, and W is the probability transition function.

The log likelihood is then defined as:

$$LL_M(T) = \sum_{i=2}^{n} \log W(T_{i-1}, T_i)$$

For every pair of models, we may use the ratio between the two likelihoods as a feature:

$$f = \frac{LL_{M_{att}}(T)}{LL_{M_{noatt}}(T)}$$

where T is the token sequence, $LL_{M_{att}}(T)$ is the log likelihood of the sequence given the attitude model, and $LL_{M_{noatt}}(T)$ is the log likelihood of the pattern given the no-attitude model.

Given the n kinds of patterns, we can calculate three different features. A standard machine learning classifier is then trained using those features to predict whether a given sentence has an attitude or not.

3.5.2 Learning from n-grams

We regard the second person pronoun and the polarized expression as two entities and try to learn a classifier that predicts whether the two entities are related or not. The text connecting the two entities offers a very condensed representation of the information needed to assess whether they are related or not. For example the two sentences *"you are completely unqualified"* and *"you know what, he is unqualified ..."* show two different ways the words *"you"*, and *"unqualified"* could appear in a sentence. In the first case the polarized word "unqualified" refers to the word "you". In the second case, the two words are not related.

The sequence of words connecting the two entities is a very good predictor for whether they are related or not. However, these paths are completely lexicalized and consequently their performance will be limited by data sparseness. To alleviate this problem, we use higher levels of generalization to represent the path connecting the two tokens. These representations are the part-of-speech tags, and the shortest path in a dependency graph connecting the two tokens as shown in Tables 3.1, and 3.2.

The set of features we use are the set of unigrams, and bigrams representing the words, part-of-speech tags, and dependency relations connecting the two tokens. We use Support Vector Machines (SVM) as a learning system because it is good with handling high dimensional feature spaces.

3.6 Identifying the Sign of an Attitude

To determine the orientation of an attitude sentence, we tried two different methods. The first method assumes that the orientation of an attitude sentence is directly related to the polarity of the words it contains. If the sentence has only positive and neutral words, it is classified as positive. If the sentence has only negative and neutral words, it is classified as negative. If the sentence has both positive and negative words, we calculate the summation of the polarity scores of all positive words and that of all negative words. The polarity score of a word is an indicator of how strong of a polarized word it is. If the former is greater, we classify the sentence as positive,otherwise we classify the sentence as negative.

The problem with this method is that it assumes that all polarized words in a sentence with an attitude target the text recipient. Unfortunately, that is not always correct. For example, the sentence ”You are completely unqualified to judge this great person” has a positive word ”great” and a negative word ”unqualified”. The first method will not be able to predict whether the sentence is positive or negative. To solve this problem, we use another method that is based on the paths that connect polarized words to second person pronouns in a dependency parse tree. For every positive word w , we identify the shortest path connecting it to every second person pronoun in the sentence then we compute the average length of the shortest path connecting every positive word to the closest second person pronoun. We repeat for negative words and compare the two values. The sentence is classified as positive if the average length of the shortest path connecting positive words to the closest second person pronoun is smaller than the corresponding value for negative words. Otherwise, we classify the sentence as negative.

3.7 Data

Our data was randomly collected from a set of discussion groups. We collected a large number of threads from the first quarter of 2009 from a set of Usenet discussion groups. All threads were in English, and had 5 posts or more. We parsed the downloaded threads to identify the posts and senders. We kept posts that have quoted text and discarded all other posts. The reason behind that is that participants usually quote other participants text when they reply to them. This restriction allows us to identify the target of every post, and raises the probability that the post will display an attitude from its writer to its target. We plan to use more sophisticated methods for reconstructing the reply structure like the one in [87]. From those posts, we randomly selected approximately 10,000 sentences that use second person pronouns. We explained earlier how second person pronouns are used in discussion genres to indicate that the text writer is targeting the text recipient. Given a random sentence selected from some random discussion thread, the probability that the sentence does not have an attitude is significantly higher than the probability that it will have an attitude. Hence, restricting our dataset to posts with quoted text and sentences with second person pronouns is very important to make sure that we will have a considerable amount of attitudinal sentences. The data was tokenized, sentence-split, part-of-speech tagged with the OpenNLP toolkit. It was parsed with the Stanford dependency parser [72].

3.7.1 Annotation Scheme

The goal of the annotation scheme is to distinguish sentences that display an attitude from those that do not. Sentences could display either a negative or a positive attitude. Disagreement, insults, and negative slang are indicators of negative attitude. Agreement and praise are indicators of positive attitude. Our annotators were instructed to read every sentence and assign two labels to it. The first specifies whether the sentence displays an attitude or not. The existence of an attitude was judged on a three point scale: attitude, unsure, and no-attitude. The second is the sign of the attitude. If an attitude exists, annotators were asked to specify whether the attitude is positive or negative. To evaluate inter-annotator agreement, we use the agr operator presented in [140]. This metric measures the precision and recall of one annotator using the annotations of another annotator as a gold standard. The process is repeated for all pairs of annotators, and then the

	A	B	C	D
A	-	82.7	80.6	82.1
B	81.0	-	81.9	82.9
C	77.8	78.2	-	83.8
D	78.3	77.7	78.6	-

Table 3.3: Inter-annotator agreement in the attitude annotation task.

harmonic mean of all values is reported. Formally:

$$agr(A|B) = \frac{|A \cap B|}{|A|} \tag{3.1}$$

where A, and B are the annotation sets produced by the two reviewers. Table 3.3 shows the value of the agr operator for all pairs of annotators. The harmonic mean of the agr operator is 80$\%$. The agr operator was used over the Kappa Statistic because the distribution of the data was fairly skewed.

3.8 Experiments

3.8.1 Experimental Setup

We performed experiments on the data described in the previous section. The number of sentences with an attitude was around 20% of the entire dataset. The class imbalance caused by the small number of attitude sentences may hurt the performance of the learning algorithm [115]. A common way of addressing this problem is to artificially rebalance the training data. To do this we down-sample the majority class by randomly selecting, without replacement, a number of sentences without an attitude that equals the number of sentences with an attitude. That resulted in a balanced subset, approximately 4000 sentences, that we used in our experiments.

We used Support Vector Machines (SVM) as a classifier. We optimized SVM separately for every experiment. We used 10-fold cross validation for all tests. We evaluate our results in terms of precision, recall, accuracy, and F1. Statistical significance was tested using a

2-tailed paired t-test. All reported results are statistically significant at the 0.05 level. We compare the Markov model based method to several other baselines that will be described in the next subsection. We also perform experiments to measure the performance if we mix features from the baselines and the proposed method. Finally, we assess whether using n-grams as features and learning their weights using SVM improves performance or not.

3.8.2 Baselines

The first baseline is based on the hypothesis that the existence of polarized words is a strong indicator that the sentence has an attitude. As a result, we use the number of polarized words in the sentence, the percentage of polarized words to all other words, and whether the sentences has polarized words with mixed or same sign as features to train an SVM classifier to detect attitude.

The second baseline is based on the proximity between the polarized words and the second person pronouns. We assume that every polarized word is associated with the closest second person pronoun. Let w be a polarized word and $p(w)$ be the closest second person pronoun, and $surf\_dist(w, p(w))$ be the surface distance between w and $p(w)$. This baseline uses the minimum, maximum, and average of $surf\_dist(w, p(w))$ for all polarized words as features to train an SVM classifier to identify sentences with attitude.

The next baseline uses the dependency tree distance instead of the surface distance. We assume that every polarized word is associated to the second person pronoun that is connected to it using the smallest shortest path. The $dep\_dist(w, p(w))$ is calculated similar to the previous baselines but using the dependency tree distance. The minimum, maximum, and average of this distance for all polarized words are used as features to train an SVM classifier.

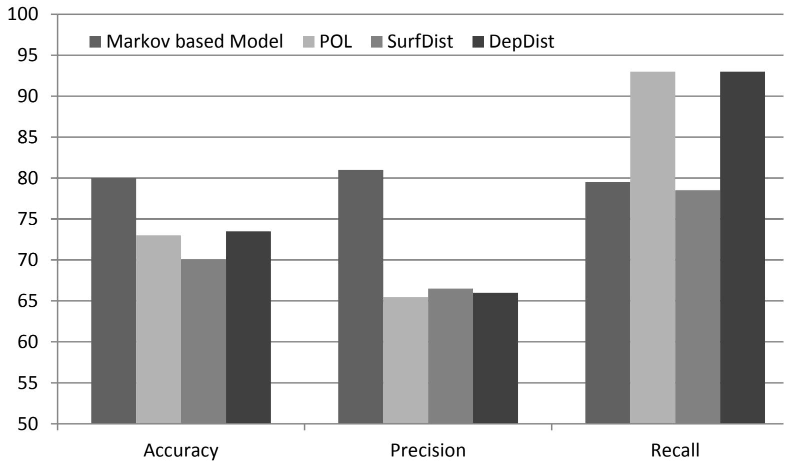


Figure 3.2: Accuracy, precision, and recall for the Markov model based method and the baselines.

3.8.3 Results and Discussion

Figure 3.2 compares the accuracy, precision, and recall of the proposed Markov model based method (ML), the polarity based classifier (POL), the surface distance based classifier (Surf_Dist), and the dependency distance based classifier (Dep_Dist). The values are selected to optimize F1. The figure shows that the surface distance based classifier behaves poorly with low accuracy, precision, and recall. The two other baselines behave poorly as well in terms of precision and accuracy, but they do very well in terms of recall. We looked at some of the examples to understand why those two baselines achieve very high recall. It turns out that they tend to predict most sentences that have polarized words as sentences with attitude. This results in many false positives and low true negative rate. Achieving high recall at the expense of losing precision is trivial. On the other hand, we notice that the proposed method results in very close values of precision and recall at the optimum F1 point.

To better compare the performance of the proposed method and the baseline, we study

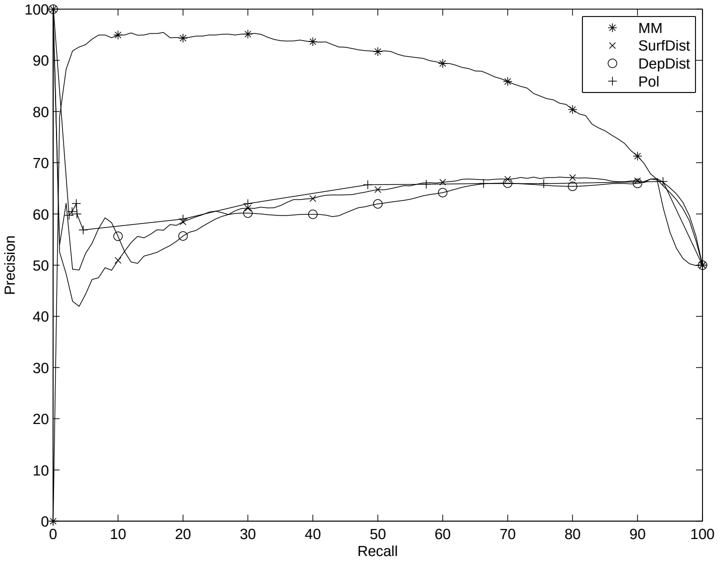


Figure 3.3: Precision recall graph comparing the Markov model based method and the baselines.

the precision-recall curves for all methods in Figure 3.3. We notice that the proposed Markov model based method outperforms all baselines at all operating points. We also notice that the proposed method provides a nice trade-off between precision and recall. This allows us some flexibility in choosing the operating point. For example, in some applications we might be interested in very high precision even if we lose recall, while in other applications we might sacrifice precision in order to get high recall. On the other hand, we notice that the baselines always have low precision regardless of recall.

Table 3.4 shows the accuracy, precision, recall, and F1 for the proposed Markov model based method and all baselines. It also shows the performance when we add features from the baselines to the proposed method, or merge some of the baselines. We see that we did not get any improvement when we added the baseline features to the proposed method. We believe that the proposed method subsumes the information captured by the baselines.

Our proposed method uses three different features that correspond to the three types of patterns we use to represent every sentence. To understand the contributions of every feature, we measure the performance of every feature by itself and also all possible com-

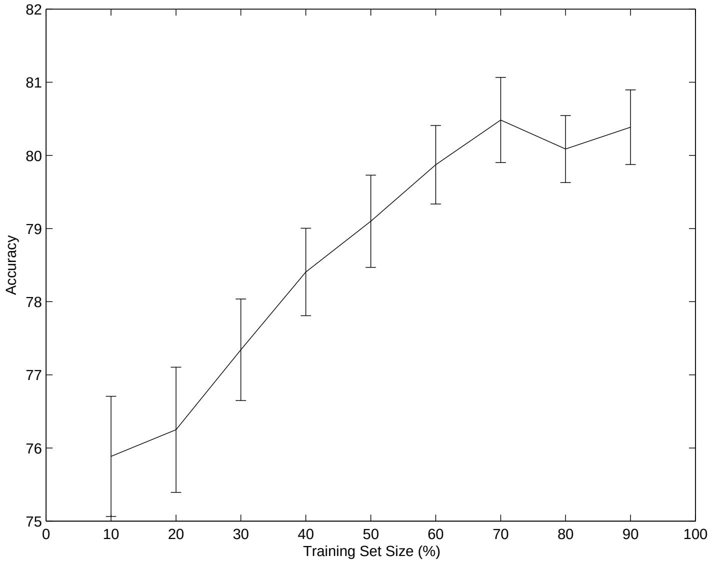


Figure 3.4: Accuracy learning curve for the proposed method.

binations of pairs of features. We compare that to the performance we get when using all features in Table 3.5. We see that the part-of-speech patterns performs better than the text patterns. This makes sense because the former suffers from data sparseness. Dependency patterns perform best in terms of recall, while part-of-speech patterns outperform all others in terms of precision, and accuracy. All pairs of features outperform any single feature that belongs to the corresponding pair in terms of F1. We also notice that using the three features results in better performance when compared to all other combinations. This shows that every kind of pattern captures slightly different information when compared to the others. It also shows that merging the three features improves performance.

We also compared the performance of the two learning algorithms we proposed. Table 3.6 compares the precision, recall, F1, and accuracy for the Markov chains based model and the learning from n-grams model. Both models try to learn the characteristics of the path connecting the second person pronouns and the polarized expressions. The former learns transition probabilities using Maximum Likelihood Estimation with smoothing, while the later learns the weights directly using SVM. We notice that optimizing the

Method	Accuracy	Precision	Recall	F1
ML	80.3	81.0	79.4	80.2
POL	73.1	66.4	93.9	77.7
ML+POL	79.9	77.9	83.4	80.5
SurfDist	70.2	67.1	79.2	72.7
DepDist	73.1	66.4	93.8	77.8
SurfDist+DepDist	73.1	66.4	93.8	77.7
ML+SurfDist	73.9	67.2	93.6	78.2
ML+DepDist	72.8	66.1	93.8	77.6
ML+SurfDist+DepDist	74.0	67.2	93.4	78.2
SurfDist+DepDist+POL	73.1	66.3	93.8	77.7
ML+SurfDist+DepDist+POL	73.0	66.2	93.8	77.6

Table 3.4: Precision, recall, F1, and accuracy for the proposed Markov model based method, the baselines, and different combinations of proposed method and the baselines features

Method	Accuracy	Precision	Recall	F1
txt	75.5	74.1	78.6	76.2
pos	77.7	78.2	76.9	77.5
dep	74.7	70.4	85.1	77.0
txt+pos	77.8	77.0	79.4	78.1
txt+dep	79.4	79.6	79.2	79.4
pos+dep	80.4	79.1	82.5	80.7
txt+pos+dep	80.3	81.0	79.4	80.2

Table 3.5: Precision, recall, F1, and accuracy for different combinations of the proposed Markov model based method's features.

weights for n-grams using SVM results in a better performance.

One important question is how much data is required to the proposed model. We constructed a learning curve, shown in Figure 3.4, by fixing the test set size at one tenth of the data, and varying the training set size. We carried out ten-fold cross validation as with our previous experiments. We see that adding more data continues to increase the accuracy, and that accuracy is quite sensitive to the training data. This suggests that adding more data to this model could lead to even better results.

We also measured the accuracy of the two methods we proposed for predicting the sign of attitudes. The accuracy of the first model that only uses the count and scores of polarized words was 95%. The accuracy of the second method that used dependency distance was 97%.

Method	Accuracy	Precision	Recall	F1
ML	80.3	81.0	79.4	80.2
N-gram Learning	82.3	82.3	82.3	82.3

Table 3.6: Precision, recall, F1, and accuracy for the two proposed method (Markov models vs. learning from unigrams and bigrams)

3.8.4 Error Analysis

We had a closer look at the results to find out what are the reasons behind incorrect predictions. We found two main reasons. First, errors in predicting word polarity usually propagates and results in errors in attitude prediction. The reasons behind incorrect word polarity predictions are ambiguity in word senses and infrequent words that have very few connection in thesaurus. A possible solution to this type of errors is to improve the word polarity identification module by including word sense disambiguation and adding more links to the words graph using glosses or co-occurrence statistics. The second reason is that some sentences are sarcastic in nature. It is so difficult to identify such sentences. Identifying sarcasm should be addressed as a separate problem. A method that utilizes a holistic approach that takes context and previous interactions between discussion participants into consideration could be used to address it.

3.9 Conclusions

We have shown that training supervised models using features of text, part-of-speech tags, and dependency relations allows us to identify sentences with attitudes from sentences without attitude. This model is more accurate than several other baselines that use features based on the existence of polarized word, and proximity between polarized words and second person pronouns both in text and dependency trees. This method allows us to extract signed social networks from multi-party on-line discussions. This opens the door to research efforts that go beyond standard social network analysis that is based on positive links only. It also allows us to study the dynamics behind interactions in on-line discussions, the relation between text and social interactions, and how groups form and break in on-line discussions.

CHAPTER IV

Extracting Signed Networks from Text

4.1 Introduction

A great body of research work has focused on social network analysis. Social network analysis plays a huge role in understanding and improving social computing applications. Most of this research has almost exclusively focused on positive links between individuals (e.g. friends, fans, followers, etc.). However, if we carefully examine the relationships between individuals in on-line communities, we will find out that limiting links to positive interactions is a very simplistic assumption. Along with agreement and friendship, individuals communicating in on-line communities also show disagreement and antagonism toward other members. Discussion forums are one example that makes it clear that considering both positive and negative interactions is essential for understanding the rich relationships that develop between individuals in on-line communities.

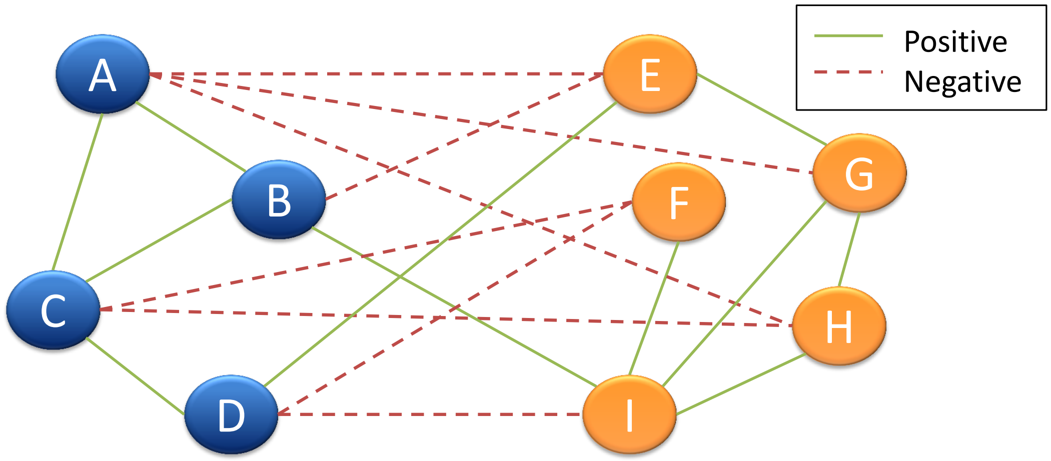
Even though, considering both negative and positive interactions provides much more insight toward understanding the social network, most of previous work only focused on positive interactions. We think that one of the main reasons behind this is the lack of a notion for explicitly labeling negative relations. For example, most social web applications allow people to mark others as friends, like them, follow them, etc. However, they do not allow people to explicitly label negative relations with others.

Previous work has built networks from discussions by linking people who reply to one another. Even though, the mere fact that X replied to Y's post does show an interaction, it does not tell us anything about the type of that interaction. In this case, the type of interaction is not readily available; however it may be mined from the text that underlies the social network. Hence, if we examine the text exchanged between individuals, we may be able to come up with conclusions about, not only the existence of an interaction, but also its type.

In this work, we apply Natural Language Processing techniques to text correspondences exchanged between individuals to identify the underlying social structure in on-line communities. Our work is uniquely characterized by the fact that we mine both positive (friendly), and negative (antagonistic) interactions.

Figure 4.1 shows a signed network representation for a subset of posts from a long (> 200 posts) discussion thread. The thread focused on the November 2010 Wikileaks cable release. Some participants praised the leak, while others criticized it. The figure shows a sample of two groups, one with and one against the release. We see that most negative edges are between groups, and most positive edges are within groups. The figure also shows pieces of evidence from text justifying signs associated with some of the edges. It is worth mentioning that networks generated from larger datasets (i.e. thousands of threads) have much more noise compared to this example. In some cases, it is clear that the participants can be easily split into two or even more groups. In other cases, it is not as clear.

The proposed method was applied to a very large dataset of on-line discussions. To evaluate our automated procedure, we asked human annotators to examine text correspondences exchanged between individuals and judge whether their interaction is positive or negative. We compared the edge signs that has been automatically identified to edges



Source	Target	Sign	Evidence from Text
A	E	-	I have to disagree with what you are saying.
G	A	-	You are missing the entire point, he is putting lives at risk.
D	I	-	and you manufacture lies for what reason?
E	G	+	you have explained your position very well.
C	H	+	I am neutral on this, but I agree with your assessment!

Figure 4.1: An example showing a signed social network along with evidence from text that justifies edge signs.

manually created by human annotators. We also connected our analysis to social psychology theories, namely the Structural Balance Theory [59], and studied how the extracted networks relate to such theories.

The rest of the chapter is structured as follows. In section 4.2, we review some of the related prior work on mining sentiment from text, mining on-line discussions, extracting social networks from text, and analyzing signed social networks. We define our problem and explain our approach in Section 4.3. Section 4.4 describes our dataset. Results and discussion are presented in Section 4.5. Some possible applications that would benefit from the proposed techniques that automatically extract signed networks from text are discussed in Section 4.6. We conclude in Section 4.7.

4.2 Related Work

In this section, we survey several lines of research that are related to our work.

4.2.1 Mining Sentiment from Text

Our general goal of mining attitude from one individual toward another makes our work related to a huge body of work on sentiment analysis. One such line of research is the well studied problem of identifying the polarity of individual words. We described a method in Chapter II, also published in [54], that uses a random walk model defined over a word relatedness graph to classify words as either positive or negative. We also surveyed work related to this problem in Chapter II.

Subjectivity analysis is yet another research line that is closely related to our general goal of mining attitude. The objective of subjectivity analysis is to identify text that presents opinion as opposed to objective text that presents factual information [138]. Prior work on subjectivity analysis mainly consists of two main categories: The first category is concerned with identifying the subjectivity of individual phrases and words regardless of the sentence and context they appear in [138, 58, 9]. In the second category, subjectivity of a phrase or word is analyzed within its context [119, 143, 107, 113]. A very detailed survey that covers techniques and approaches in sentiment analysis and opinion mining could be found in [112]. Our work is different from subjectivity analysis because we are not only interested in discriminating between opinions and facts. Rather, we are interested in identifying the polarity of interactions between individuals. Apparently, all text that carry an attitude towards another participant in opinionated. However, not all opinionated text shows an attitude. Hence, the language constituents considered in opinion detection may be different from those used to detect attitude.

In Chapter III, also published in [53], we present a method for identifying sentences that display an attitude from the text writer toward the text recipient. We define attitude as the mental position of one participant with regard to another participant. This work is limited to studying only sentences. The work we describe in this chapter builds on this work by generalizing it to identifying the polarity of an interaction between two individuals based on several posts they exchange. This allows us to extract a signed social network representing interactions in discussions.

4.2.2 Mining Online Discussions

Our use of discussion threads as a source of data connects us to some previous work on mining on-line discussions. Lin et al. [87] proposed a sparse coding-based model that simultaneously models semantics and structure of threaded discussions. Huang et al. [61] learn SVM classifiers from data to extract (thread-title, reply) pairs. Their objective was to build a chatbot for a certain domain using knowledge from on-line discussion forums. Shen et al. [120] proposed three clustering methods for exploiting the temporal information in discussion streams, as well as an algorithm based on linguistic features to analyze discourse structure information. Other work has focused on the structure of questions and question-answer pairs in on-line forums and discussions [28, 24].

4.2.3 Extracting Social Networks from Text

Little work has been done on the front of extracting social relations between individuals from text. Elson et al. [37] present a method for extracting social networks from nineteenth-century British novels and serials. They link two characters based on whether they are in conversation or not. McCallum et al. [94] explored the use of structured data such as email headers for social network construction. Gruzad and Hyrthonthwaite [48] explored the use of post text in discussions to study interaction patterns in e-learning communities.

Our work is related to this line of research because both of them employ natural language processing techniques to reveal embedded social structures. Despite similarities, our work is uniquely characterized by the fact that we extract signed social networks from text. Signed social networks are different because every edge has an associated polarity that differentiates between friendly and antagonistic interactions.

4.2.4 Signed Social Networks

There is a great body of work that focuses on the analysis of social networks. However, most of this work has only focused on positive interactions (e.g. friends, fans, followers, etc.). A few recent papers have taken the signs of edges into account.

Brzozowski et al. [17] study the positive and negative relationships between users of Essembly. Essembly is an ideological social network that distinguishes between ideological allies and nemeses. Kunegis et al. [77] analyze user relationships in the Slashdot technology news site. Slashdot allows users of the website to tag other users as friends or foes, providing positive and negative endorsements. Leskovec et al. [84] study signed social networks generated from Slashdot, Epinions, and Wikipedia. They also connect their analysis to theories of signed networks from social psychology. A similar study used the same datasets for predicting positive and negative links given their context [83]. Other work addressed the problem of partitioning signed networks by taking both positive and negative edges into consideration [142, 31].

All this work has been limited to analyzing a handful of datasets for which an explicit notion of both positive and negative relations exists. Our work goes beyond this limitation by leveraging the power of natural language processing to automate the discovery of signed social networks using the text embedded in the network.

4.3 Approach

The general goal of this work is to mine attitude between individuals engaged in an online discussion. We use that to extract a signed social network representing the interactions between different participants. Our approach consists of several steps. In this section, we will explain how we identify sentiment at the word level (i.e. polarity), at the sentence level (i.e. attitude), and finally generalize over this to find positive/negative interactions between individuals based on their text correspondences.

4.3.1 Finding Sentences with an Attitude

The first step toward identifying attitude is to identify words with positive/negative semantic orientation. The semantic orientation or polarity of a word indicates the direction the word deviates from the norm [82]. We use a Random Walk based method, described in Chapter II, to identify the semantic orientation of words. We construct a graph where each node represents a word/part-of-speech pair. We connect nodes based on synonyms, hypernyms, and similar-to relations from WordNet [101]. For words that do not appear in WordNet, we use distributional similarity [81] as a proxy for word relatedness.

The polarity of a word is also affected by the context where the word appears. For example, a positive word that appears in a negated context should have a negative polarity. Other polarized words sometimes appear as neutral words in some contexts. Hence, we use the method described in [141] to identify the contextual polarity of words given their isolated polarity. A large set of features is used for that purpose including words, sentences, structure, and other features.

After identifying the semantic orientation of individual words, we use the work, described in Chapter III, to identify sentences that show positive or negative attitude. We pose the problem as a relation extraction problem where we seek to predict whether a sentiment expression is referring to the addressee or not.

We represent every sentence with several representations at different levels of generalization. For example, the sentence "*your ideas are very inspiring*" will be represented using lexical, polarity, part-of-speech, and dependency information as follows:

LEX: "*YOUR ideas are very POS*"

POS: "*YOUR NNS VBP RB JJ_POS*"

DEP: "*YOUR poss nsubj POS*"

We use a supervised classification paradigm where we train an SVM classifier using features based on the existence of unigrams and bigrams from the three representations described above.

Once we decide that a sentence has an attitude, we can determine the sign of that attitude with help of a dependency parse tree. We compute the average length of the shortest paths connecting every positive word to the closest second person pronoun. We repeat for negative words and compare the two values. The sentence is classified as positive if the average length of the shortest paths connecting positive words to the closest second person pronoun is smaller than the corresponding value for negative words. Otherwise, it is classified as negative.

4.3.2 Extracting the Signed Network

In this subsection, we describe the procedure we used to build the signed network given the components we described in the previous subsection. This procedure consists of two main steps. The first is building the network without signs, and the second is assigning signs to different edges. We will go ahead and describe how every step is performed in the next few paragraphs.

To build the network, we parse our data to identify different threads, posts and senders.

Every sender is represented with a node in the network. An edge connects two nodes if an interaction between the corresponding participants exists. We add a directed edge $A \rightarrow B$, if A replies to B's posts at least n times in m different threads. We set m, and n to 2 in most of our experiments. To determine whether a particular post p_1 is a reply to another post p_2, we use quoted text. The design of most discussion forums allows participants to quote text from other participants' posts to indicate that they are replying to them. We extract quoted text and match it to previous posts to decide whether p_1 is a reply to p_2 or not.

Once we build the network, we move to the more challenging task in which we associate a sign with every edge. Unfortunately the sign of an interaction cannot be trivially inferred from the polarity of sentences. For example, a single negative sentence wrote by A and directed to B does not mean that the interaction between A and B is negative. One way to solve this problem would be to compare the number of negative sentences to positive sentences in all posts between A and B and classify the interaction according to the plurality value. We will show later, in our experiment section, that such a simplistic method does not perform well in predicting the sign of an interaction.

As a result, we decided to pose the problem as a classical supervised learning problem. We came up with a set of features that we think are good predictors of the interaction sign, and we trained a classifier using those features on a labeled dataset. Our features include numbers and percentages of positive/negative sentences per post, posts per thread, and so on. Some of the features we used are outlined below:

- Average number of positive sentences per post.
- Average number of negative sentences per post.
- Percentage of posts with more positive sentences than negative sentences (i.e. posi-

tive posts).

- Percentage of posts with more negative sentences than positive sentences (i.e. negative posts).
- Percentage of positive posts per thread.
- Percentage of negative posts per thread.
- Average number of positive posts per thread.
- Average number of negative posts per thread.
- Discussion Topic.

4.4 Data

Our data consists of a large amount of discussion threads collected from on-line discussion forums. We collected around $41,000$ threads and 1.2M posts from the period between the end of 2008 and the end of 2010. All threads were in English and had 5 posts or more. They covered a wide range of topics including: politics, religion, science, etc. We parsed the data and identified posts and senders as well as quoted text. The data was tokenized, sentence-split, and part-of-speech tagged with the OpenNLP toolkit. It was parsed with the Stanford parser [72].

We also collected an explicit friendship network connecting participants. This network has an edge connecting two individuals if they explicitly marked each others as friends. We will use this network in baselines that we will describe later.

We randomly selected 1000 interactions (with more than 5000 posts), and asked human annotators to label them. Our annotators were instructed to read all the posts exchanged between two participants and decide whether they have a positive or a negative interaction. We used Amazon Mechanical Turk for annotations. Following previous work [20, 4], we

	Class	Pos.	Neg.	Weigh. Avg.
Logistic Reg.	TP Rate	0.884	0.657	0.812
	FP Rate	0.343	0.116	0.271
	Precision	0.848	0.724	0.809
	Recall	0.884	0.657	0.812
	F-Measure	0.866	0.689	0.81
	Accuracy	-	-	**0.812**
SVM	TP Rate	0.847	0.809	0.835
	FP Rate	0.191	0.153	0.179
	Precision	0.906	0.71	0.844
	Recall	0.847	0.809	0.835
	F-Measure	0.875	0.756	0.838
	Accuracy	-	-	**0.835**

Table 4.1: Several metrics assessing the performance of the interaction sign classifier.

took several precautions to maintain data integrity. We restricted annotators to those based in the US to maintain an acceptable level of English fluency. We also restricted annotators to those who have more than 95% approval rate for all previous work. Moreover, we asked three different annotators to label every interaction. The label was computed by taking the majority vote among the three annotators.

The kappa measure between the three groups of annotations was 0.62. To better assess the quality of the annotations, we asked a trained annotator to label 10% of the data. We measured the agreement between the expert annotator and the majority label from the mechanical Turk. The kappa measure was 0.69.

Other datasets have been used to train different components of the system. The list of seed words used with the word polarity component came from [57, 126]. The sentence attitude classifier was trained using the data described in [53]. None of this data overlaps with the dataset described earlier.

4.5 Results and Discussion

We performed experiments on the data described in the previous section. We train and test the interaction sign classifier described in 4.3.2 on the labeled interactions. We build one unsigned network from every topic in the data set. We use the trained interaction

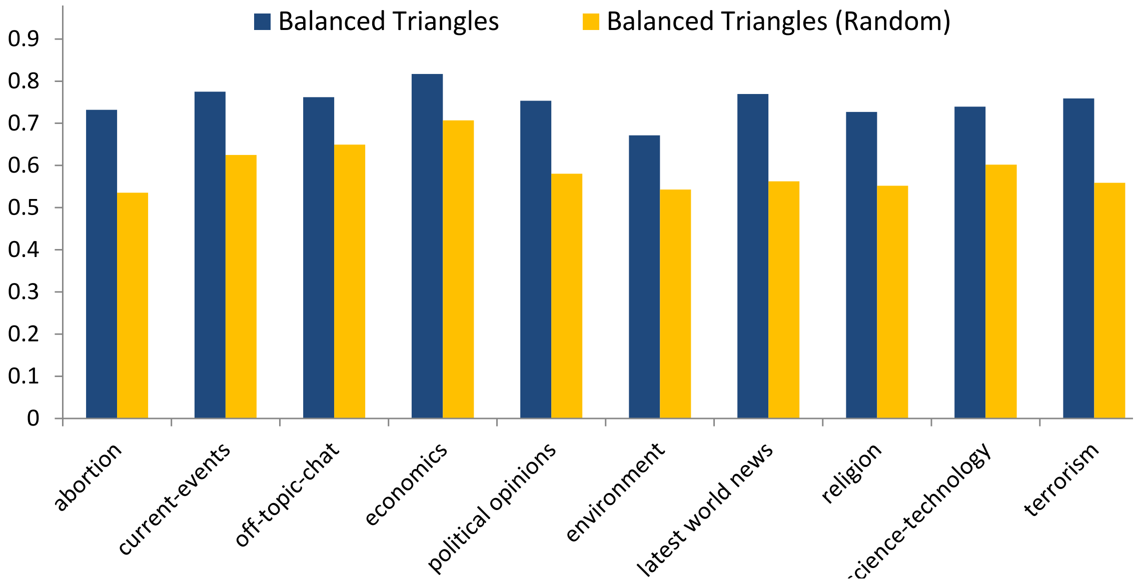


Figure 4.2: Percentage of balanced triangles in extracted network vs. random network.

sign classifier to associate a sign with every edge. This results in a signed social network for every topic. We decided to build a network for every topic as opposed to one single network because the relation between any two individuals may not be the same across topics. A and B may agree on some topic T_1, yet mostly disagree on another topic T_2. In the rest of this section, we will describe the experiments we did to assess the performance of the interaction sign classifier. We also connect our analysis to the structural balance theory and assess the agreement between our findings and this theory.

4.5.1 Interaction Sign Classifier

We used the features described in Section 4.3.2 to train a classifier that predicts the sign of an interaction between any two individuals. We used both Support Vector Machines (SVM) and logistic regression. We report several performance metrics for the two classifiers in Table 4.1. We notice that the SVM classifier performs better with an accuracy of 83.5% and an F-measure of 81%. All results were computed using 10 fold cross validation on the labeled data.

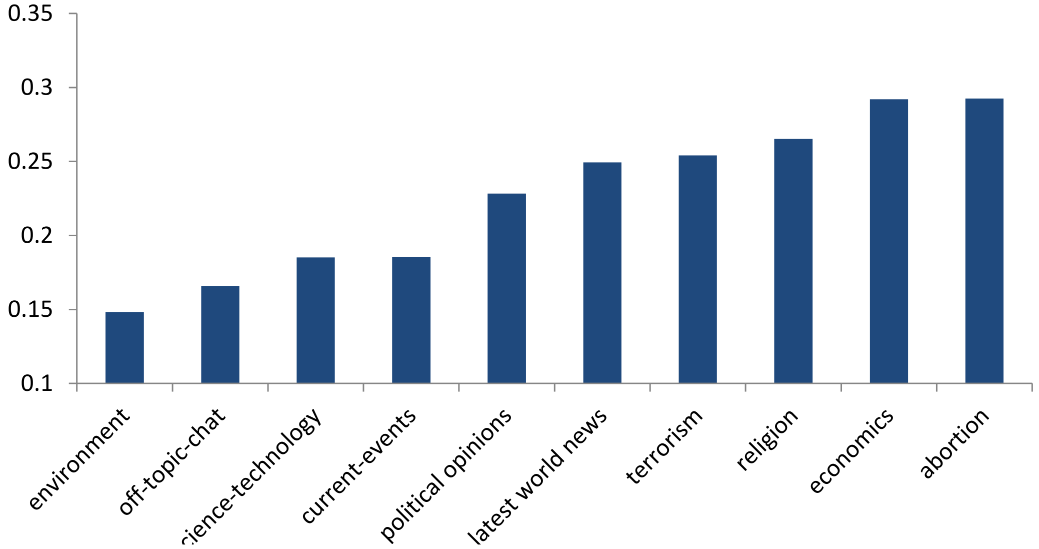


Figure 4.3: Percentage of negative edges across topics.

To better assess the performance of the proposed classifier, we compare it to three baselines. The first is a random baseline that predicts an interaction as positive with probability p that equals the proportion of positive instances to all instances in the training set. The other two classifiers are based on an explicit friendship network, described in Section 4.4. We compute the maximum number and percentage of negative posts exchanged between any two friends. We label the interaction as negative if the number or percentage of negative posts exceeds the numbers we computed using the friendship network. Otherwise, the interaction is labeled as positive.

The three baselines were tested using the entire labeled dataset. The SVM classifier was tested using 10 fold cross validation. The accuracy of the random classifier, the two based on a cut off number and percentage from the friendship network, and the SVM classifier are 65%, 69%, 71%, and 83.5% respectively. We notice that the random classifier is worst, and the classifier based on percentage cutoff outperforms the one based on number cutoff. The SVM classifier significantly outperforms all other classifiers.

	Extracted Networks				Random Networks			
Topic	(+ + +)	(+ + −)	(+ − −)	(− − −)	(+ + +)	(+ + −)	(+ − −)	(− − −)
abortion	51.67	26.31	18.92	0.48	35.39	43.92	18.16	2.52
current-events	67.36	22.26	8.76	0.23	54.08	36.90	8.39	0.64
off-topic-chat	65.28	23.54	9.45	0.25	58.07	34.59	6.88	0.46
economics	72.68	18.30	7.77	0.00	66.50	29.09	4.22	0.20
political opinions	60.60	24.24	12.81	0.43	45.97	40.79	12.06	1.19
environment	47.46	32.54	17.26	0.30	37.38	43.61	16.89	2.12
latest world news	58.29	22.41	16.33	0.62	42.26	42.20	13.98	1.56
religion	47.17	25.89	22.56	1.42	39.68	42.94	15.51	1.87
science-technology	57.53	26.03	14.33	0.00	50.14	38.93	10.05	0 .87
terrorism	64.96	23.36	9.46	0.73	41.54	42.42	14.36	1.68

Table 4.2: Percentage of different types of triangles in the extracted networks vs. the random networks.

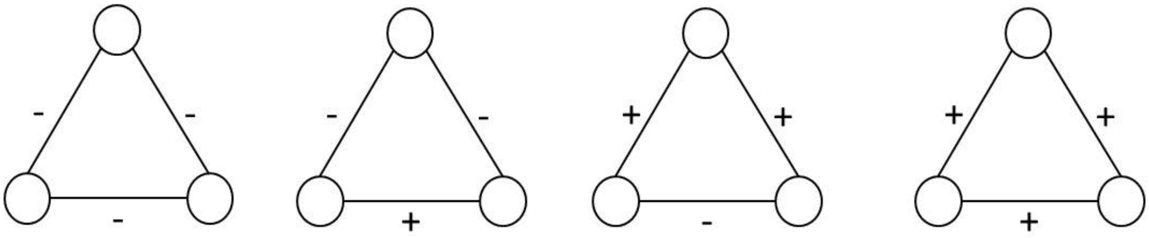


Figure 4.4: Triangles with an odd number of pluses are balanced and triangles with an even number of pluses are unbalanced.

4.5.2 Structural Balance Theory

The structural balance theory is a psychological theory that tries to explain the dynamics of signed social interactions. It has its origins in the work of Heider [59]. It was then formalized in a graph theoretic form in [21]. The theory is based on the principles that "the friend of my friend is my friend", "the enemy of my friend is my enemy", "the friend of my enemy is my enemy", and variations on these.

There are several possible ways in which triangles representing the relation of three people can be signed. The structural balance theory states that triangles that have an odd number of positive signs are balanced, while triangles that have an even number of positive signs are not. Figure 4.4 shows an example of the possible ways a triangle can be signed. According to the structural balance theory, the second and the fourth triangles are balanced, while the first and the third are not.

Even though the structural balance theory posits some triangles as unbalanced, that does not eliminate the chance of their existence. Actually, for most observed signed structures for social groups, exact structural balance does not hold [30]. Davis [26] developed the theory further into the weak structural balance theory, where he suggests that only triangles with exactly two positive edges are implausible in real networks, and that all other kinds of triangles should be permissible.

In this section, we connect our analysis to the structural balance theory. We compare the predictions of edge signs made by our system to the structural balance theory by counting the frequencies of different types of triangles in the predicted network. Table 4.2 shows the frequency of every type of triangles for 11 different topics. To better understand these numbers, we compare them to the frequencies of triangles in a set of random networks. We shuffle signs for all edges on every network keeping the fractions of positive and negative edges constant. We repeat shuffling for 1000 times. Every time, we compute the frequencies of different types of triangles. Table 4.2 shows the average frequencies over the 1000 runs.

We find that the all-positive triangle $(+++)$ is overrepresented in the generated network compared to chance across all topics. We also see that the triangle with two positive edges $(+ + -)$, and the all-negative triangle $(- - -)$ are underrepresented compared to chance across all topics. The triangle with a single positive edge is slightly overrepresented in most but not all of the topics compared to chance. This shows that the predicted network mostly agree with the structural balance theory. The slightly non standard behavior of the triangle with one positive edge could be explained in light of the weak balance theory by Davis [26]. Davis notes that the balance theorem is based on the following aphorisms:

1. A friend of a friend will be a friend

2. An enemy of a friend will be an enemy

3. A friend of an enemy will be an enemy

4. An enemy of an enemy will be a friend

He shows that the first three propositions continue to hold in his weak structural balance theorem. The fourth, however, depends on the number of subsets of the network. It holds in the case of two subsets, but not where there are more than two [26].

We also compare the number of balanced and unbalanced triangles, as defined by the structural balance theory, in the predicted networks and the corresponding shuffled networks. Figure 4.2 compares the percentage of balanced triangles in the predicted networks and the shuffled networks. We see that the balanced triangles are significantly overrepresented compared to chance.

4.6 Applications

In this section, we describe several applications that could benefit from the signed network representation of discussions that we automatically extract from text.

4.6.1 Disputed Topics

Discussion forums usually respond quickly to new topics and events. Some of those topics usually receive more attention and more dispute than others. We can identify such topics and in general measure the amount of dispute every topic receives using the extracted signed network. We computed the percentage of negative edges to all edges for every topic. We believe that this would act as a measure for how disputed a particular topic is. We see, from Figure 4.3, that “environment”, “science”, and “technology” topics are among the least disputed topics, whereas “terrorism”, “abortion” and “economics” are among the most disputed topics.

These findings are another way of validating our predictions. They also suggest another application for this work that focuses on measuring the amount of dispute different topics receive. This can be done for more specific topics, rather than high level topics as shown here, to identify hot topics that receive a lot of dispute.

4.6.2 Identifying Subgroups

In any discussion, different subgroups may emerge. Members of every subgroup usually have a common focus (positive or negative) toward the topic being discussed. Each member of a group is more likely to show a positive attitude (i.e. agreement, praise, etc.) to members of the same group. Members are also more likely to show a negative attitude (i.e. disagreement, insult, etc.) toward members of the opposing group. The signed network representation could prove to be very useful for identifying those subgroups.

The structural balance theory we described earlier states that a balanced network could be partitioned into two mutually antagonistic subgroups each having internal solidarity [21]. The weak structural balance theory developed by Davis [26] extended this theorem for cases where there can be more than two such mutually antagonistic subgroups. Doreian and Mrvar [30] used these theorems to construct a criterion function for a local optimization partitioning procedure for signed networks. The resulting partitions are optimized to get as much positive edges as possible within partitions and as much negative edges as possible between partitions. We use this algorithm to identify subgroups with opposing focus in discussions by partitioning a signed network automatically extracted from text as described earlier.

For experiments, we use a thread discussing the "Health Care Reform Bill". The thread started right after the bill was passed. It contained 2841 posts by 161 participants and remained active for several weeks. We asked a human annotator to examine the text written by every participant and decide whether he/she is with or against the bill. We ended up with two manually annotated subgroups with people with and against the bill. Figure 4.5 shows a signed network representation of the thread. Blue (dark) nodes represent participants with the bill, Yellow (light) nodes represent participants against the bill, red (solid) edges represent negative attitude, while green (dashed) edges represent positive attitude.

We compare our results to two baselines. The first uses graph clustering to partition a network based on the interaction of participants. We built a graph where each node represents a participant. Edges link participants if they exchange posts, and edge weight is based on the number of interactions. The second baseline is based on the premise that participants with similar text are more likely to belong to the same subgroup. We measure text similarity by computing the cosine similarity between tf-idf representations of text in a high dimensional vector space. We tried several methods for clustering those graphs: spectral clustering [89], Markov Clustering (MCL) [133], and a hierarchical agglomeration algorithm which works by greedily optimizing the modularity for graphs [23].

To evaluate the quality of the resulting partitions, we calculated the purity of every cluster. Purity [92] is a method in which each cluster is assigned to the class with the majority vote in 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:

$$purity(\Omega, C) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j| \tag{4.1}$$

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.

Table 4.3 shows the purity of the method based on signed networks and the baselines using different clustering algorithms. We notice that partitioning the signed network au-

Method	Purity
Signed Network Partitioning	77.8%
Unsigned Network - Spectral	55.4%
Unsigned Network - MCL	55.6%
Unsigned Network - Hierarchical	54.6%
Text - Spectral	55.2%
Text - MCL	53.9%
Text - Hierarchical	52.3%

Table 4.3: Cluster Purity for Singed Network partitioning, Unsigned Network Partitioning, and Text Clustering.

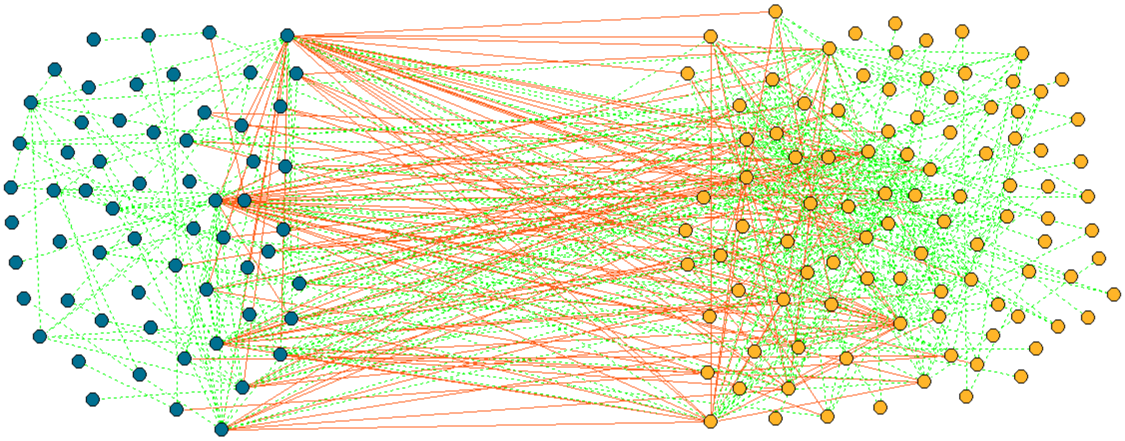


Figure 4.5: A signed network representing participants in a discussion about the "Health Care Reform Bill". Blue nodes represent participants with the bill, Yellow nodes represent participants against the bill, red edges represent negative attitude, while green edges represent positive attitude.

tomatically extracted from text results in significantly better partitions. We believe that the baselines performed poorly because the interactions and the text similarity are not a key factor 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. People in opposing subgroups tend to use very similar text when discussing the same topic and hence text clustering does not work as well.

4.7 Conclusions

In this chapter, we have shown that natural language processing techniques can be reliably used to extract signed social networks from text correspondences. We believe that this work provides an important step toward understanding the relation between language use and social interactions. We also believe that it will open the door to research efforts that go beyond standard social network analysis by studying the interplay of positive and negative connections. We also connected analysis social psychology theories of signed network and showed that our predictions mostly agree with the structural balance theory. Finally, we presented several applications that may benefit from the automatically extracted signed network.

Part II. Mining Salience from Text

CHAPTER V

Content based Networks

5.1 Introduction

There are several sources of data that record debates or discussions among a group of people. Those include parliamentary records, blogs, online forums, social networking applications, news groups, etc. This data represents a very important and unexploited source of information that contains several trends and ideas. In any debate or discussion, there are certain types of persons who influence other people and pass information or advice to them. Those persons are often regarded as experts in the field and they tend to affect the ideas and rhetoric of other participants. Identifying those influential persons is usually hard especially due to the lack of any structure that may describe how participants interact with each other.

Link popularity based algorithms, like PageRank [111] and Hypertext Induced Topic Selection (HITS) [73] have been successfully used for identifying authoritative Web pages. PageRank assigns a numerical weight to each Web page according to the links it receives and the weights of the pages that link to it. HITS determines two values for a page: its authority value, which estimates the value of the content of the page, and its hub value, which estimates the value of its links to other pages. Link popularity based algorithms work only for domains in which we have an explicit link structure between entities. This makes them inapplicable to domains where no explicit link structure is present (e.g. discussion groups and parliamentary discussions), or domains with weakly linked nature (e.g. blogs [85]).

In the absence of an explicit link structure, we may rely on linguistic analysis to find implicit connections between textual entities. Given a set of participants in a discussion who have a set of posts, we propose using text similarity between posts to build a content similarity based network that we can use later to select the most salient or representative set of participants.

In this chapter, we describe content similarity based networks and compare them to link based networks. Link based networks use explicit connections between entities. On the other hand, content similarity networks link entities based on the similarity of their textual representation. For example, two blog feeds may be connected using an explicit link if a hyperlink from one post in one of them to the other exists. They may also be linked in a content network if they discuss similar ideas or have been affected by each other. Content similarity networks use linguistic analysis to find implicit relations between textual representations of entities. Extracting such implicit connections enables us to apply link analysis techniques to new domains where explicit links are not available.

The main objective of this chapter is to describe content based networks and compare their topological and temporal properties to those of explicit link based networks. We use two different datasets, the ACL Anthology (AAN) dataset and the TREC Blog06 dataset. We study the different topological and temporal properties of the two networks. We show that the two networks bear a great deal of similarity in terms of their topological and temporal properties. We also study how correlated the two networks are and whether we can predict any of them using the other.

The rest of the chapter will proceed as follows. We first present an overview of the two different datasets we used in Section 5.2. We then describe how content similarity based networks are built in Section 5.3. We compare the topological and temporal features of the link based and the content similarity based networks in Section 5.4, and Section 5.5 respectively. Next, we study the correlation between the two types of networks for the two datasets in Section 5.6. Finally, we present conclusions in Section 5.7.

5.2 Dataset Description

The Association for Computational Linguistics is an international and professional society dedicated to the advancement in Natural Language Processing and Computational Linguistics Research. The ACL Anthology is a collection of papers from an ACL published journal - Computational Linguistics - as well as all proceedings from ACL sponsored conferences and workshops [66]. The ACL Anthology Network (AAN) dataset contains 9,241 authors, 11,834 papers, 44,138 paper to paper citations, and 261,944 author to author citations [66].

The ACL Anthology Network (AAN) was created by Joseph and Radev [66]. They processed the papers using OCR extraction tools and the references from each paper were parsed and extracted. The references were also manually matched to other ACL papers. This enabled them to build both the citation, and the author collaboration networks.

We built a textual similarity based network for AAN by representing each paper as a node and linking papers that are textually similar to each other. Edges are weighted according to how similar they are to each other. Edges are also directed such that newer papers point to older ones. We also create an author similarity network by linking authors to each other based on the textual similarity of their papers.

BLOG06 is a TREC test collection, created and distributed by the University of Glasgow. It contains a crawl of Feeds, and associated Permalink and homepage documents (from late 2005 and early 2006) [90]. The dataset contains 100,649 feeds covering a period of 11 weeks. Permalink documents and blog homepages were also collected. The dataset contains a total of 3,215,171 permalink documents and 324,880 home page documents [90]. To make the collection more realistic, a list of known spam blogs was also included. 17,969 spam blogs were added causing the spam component to form a reasonable component of the collection [90].

We removed feeds that do not have any permalinks and those that do not have a homepage. We also removed automatically generated tags that appear to be blogs. We parsed all permalinks and extracted all hyperlinks in each of them. We matched hyperlinks to other permalinks in the dataset. We ignored links pointing to posts from the same blog and links pointing to pages outside the dataset. We used this to build a posts network and a blogs network. In the posts network, each node represents a post and directed edges between posts indicates a hyperlink from one post to the other. In the blogs network, each node represents a blog feed. Edges indicate that the posts in the two blog feeds exchanged links to each other.

A collection of blogs can also be represented by a content similarity based network. Hence given a collection of blog posts and a similarity measure, we can build a network where each post is represented by a node and edges link blogs if they are textually similar. Like the papers network, edges point from newer to older posts and edges are weighted with the posts similarity measure. We also build a blogs network where each node represents a blog feed. Edges indicate that the posts in the two blog feeds are textually similar.

5.3 Content-based Networks

A collection of discussion posts can be represented as a network where participants using similar text are linked to each other. The proposed approach for assigning salience to participants in a discussion is based on the premise that important participants tend to be lexically similar to other important participants. Or in a finer level of granularity, important posts tend to be lexically similar to other important posts, and important posts tend to belong to important participants.

Hence given a collection of posts that are related to a specific topic and a similarity measure, we can build a network where each participant is represented by a node and edges link participants if their textual similarity exceeds some threshold. The edges of the network are weighted with the weight representing how similar the text of the two participants is to one another. We refer to this network as a Content Similarity based Network.

The most popular similarity function used to measure document similarity is the well-known cosine measure defined on the document vectors in the tf or tf-idf weighted term space. Some other possible similarity measures are edit distance, Kullback-Leibler (KL) divergence [80], language models [78], or generation probabilities [40].

To measure the similarity between two posts, we can use the bag-of-words model to represent each sentence as an N-dimensional vector of tf-idf scores, where N is the number of all possible words in the target language. The similarity between two posts is then computed using the cosine similarity between the two vectors.

A vector of term frequencies is used to represent each post. Those term frequencies are weighted according to the relative importance of the given term in the collection of posts. The weights are inverse document frequencies of each term. The inverse document frequency of a term w is given by [125]

$$\mathrm{idf}(w) = \log\left(\frac{N}{n_w}\right) \tag{5.1}$$

where n_w is the number of posts in the collection containing the term w, and N is the total number of posts in the collection. The *idf* values are calculated for terms using posts that are judged as relevant to the topic of interest.

The tf-idf cosine similarity measure is computed as the cosine of the angle between the tf-idf vectors. It is defined as follows:

$$\frac{\sum_{w \in u,v} \mathrm{tf}_u(w)\, \mathrm{tf}_v(w)\, \mathrm{idf}(w)^2}{\sqrt{\sum_{w \in u} (\mathrm{tf}_u(w)\, \mathrm{idf}(w))^2} \sqrt{\sum_{w \in v} (\mathrm{tf}_v(w)\, \mathrm{idf}(w))^2}}, \tag{5.2}$$

The graph we create using this method is initially a complete graph. To filter unnecessary weak connections, we use a threshold of 0.2. An edge is included between two entities only if their similarity exceeds this threshold. Previous work suggested that the best value for the cut off threshold is usually in the interval $[0.1 - 0.3]$ [39].

5.4 Topological Features

In this section, we study the topological properties of the content similarity networks representing the AAN and TREC BLOG06 datasets. We also compare the different features of the link based and content based networks for the two datasets.

5.4.1 Degree Distributions

In an undirected graph, the degree of a vertex v is the number of edges connected to v. If the graph is directed then each node, v, has an in-degree and an out-degree. The in-degree is the number of arcs to v, and the out-degree is the number of arcs from v. The degree distribution of a network is the probability distribution of these degrees over the whole network.

Figure 5.1 compares the in-degree distribution for the citation and similarity networks of the AAN dataset. The out-degree distributions for the same networks are shown in 5.2. Finally the degree distribution of the links and similarity networks for the TREC BLOG06 dataset in shown in Figure 5.3 and Figure 5.4.

From the figures, we notice that the degree distributions of the link based and similarity based networks are very similar. Previous studies stated that the in-degree distribution for the Web is scale free [5, 6, 15, 29], and the out-degree distribution is either not to follow a power law distribution, or to exhibit a steeper power law only in the tail [5, 6, 15, 29]. The figures show that both link based and similarity based networks exhibit similar degree distributions as the Web.

5.4.2 Small-World Effect

The small world phenomenon is the theory that every pair of nodes in a network can be reached through a short path. This means that the average shortest path between every pair of reachable nodes is short when compared to the network size [75].

Previous studies of the World Wide Web have shown that the Web graph has the small-world property. Albert et al. [5] gave a formula to compute the average shortest path in the Web graph if the number of nodes in the graph N is known: $\langle l \rangle = 0.35 + 2.06 log(N)$.

The average shortest path for the citation and similarity networks for AAN was measured as 3.34 and 3.21 respectively. This is even smaller than the values calculated by Albert et al. formula. We also notice that the average shortest path in the two types of networks is very similar. The average shortest path for the links and similarity networks for TREC BLOG06 was measured as 5.30 and 3.92 respectively. Like AAN, TREC BLOG06 networks have small values for the average shortest path. However, unlike AAN, those values are a little bit different from one another. The similarity network has a smaller average shortest path which makes sense given the fact that the link based network is rather sparse.

The average shortest path is not the only parameter that indicates whether it is easy to move from one node in the network to the other. The average shortest path is calculated by averaging the paths of reachable pairs. Hence, the percentage of reachable pairs must be taken into consideration too. The percentage of reachable pairs in the citation and similarity networks for the AAN data is 37.12 and 47.22 respectively. We notice that the number of reachable pairs in the similarity network is larger, and they are both larger than the number of reachable pairs in the Web graph as reported by Broder et al. [15]. The percentages of reachable pairs in the TREC BLOG06 links and similarity networks are 39.08% and 70.26% respectively. From these values we notice that the percentage of reachable pairs in the similarity network is as large as twice the same percentage for the links based networks.

5.4.3 Clustering

The clustering coefficient of a vertex in a graph quantifies how close the vertex and its neighbors are to being a clique (complete graph) [136]. The clustering coefficient is measured as the percentage of closed triads in a network.

The average clustering coefficient of the citation and similarity networks for the AAN data are 0.316 and 0.364 respectively. The two values are close again with the similarity network showing higher average clustering coefficient than the citation network. The clustering coefficients of the links and similarity networks for the TREC BLOG06 data are 0.075 and 0.161 respectively. We notice that the average clustering coefficient of the similarity based network is much higher than that of the link based network. This is different from the AAN networks where the average clustering coefficients of the two networks is rather similar. This is due to the sparseness of the blog link based network when compared to other networks.

5.4.4 Connectivity

A directed graph is called strongly connected if there is a path from each vertex in the graph to each other vertex. The strongly connected components (SCC) of a directed graph G are its maximal strongly connected subgraphs. Another component of any directed graph is the weakly connected component (WCC). The weakly connected compo-

Network	MAX SCC(%)	MAX WCC(%)
Web [15]	27.74%	91.76%
Web [6]	16.57%	100%
BLOG-Links	39.36%	96.30%
BLOG-Sim	68.91%	99.78%
AAN-Cit	58.51%	99.29%
AAN-Sim	74.61%	99.42%

Table 5.1: The sizes of the max SCC and the max WCC for the REC BLOG06 links network, TREC BLOG06 similarity network, AAN citation network, AAN similarity network and the Web.

nent (WCC) of a directed graph G are its maximal subgraphs such there is an undirected path between every pair of vertices in the subgraph. Table 5.1 compares the sizes of the SCC and WCC for the TREC BLOG06 links network, TREC BLOG06 similarity network, AAN citation network, AAN similarity network and the Web [15, 6].

5.5 Temporal Features

The AAN data includes papers from a very large time range (several years). Hence, we decided to use this dataset to compare the temporal evolution of different network features for both the citation and similarity networks.

We create several versions of the AAN networks at different points of time and compare the topological properties of the different versions. We created 5 different versions of the citation similarity networks based on papers before 1990, 1995, 200. 2004, and 2008.

Figures 5.5 and 5.6 show the temporal evolution of the degree distributions of the citation and similarity networks for the AAN datasets. The figures indicate that the shapes of the distributions are very consistent over the different points of time. This behavior is consistent over both the citation and the similarity networks.

We also examined how the clustering coefficients changes with time. Figure 5.7 shows the temporal evolution of the clustering coefficients for the AAN citation and similarity networks. The figure shows that the clustering coefficient is consistent over time and between both the citation and similarity networks

The temporal evolution of the size of the SCC and WCC in the AAN citation and similarity networks is shown in Figure 5.8. We notice from the figure that most of the nodes in both networks belong to the biggest WCC at all points of time. Unlike the percentage of nodes in the WCC, the percentage of nodes in the SCC increases with time. This shows that as the number of nodes increases, the connectivity in both networks also increases. This behavior is consistent in both citation and similarity networks.

5.6 Correlation between Link based and Similarity based Networks

In order to better understand the relation between link-based and similarity-based networks, we decided to measure the correlation between the existence of links and high similarity. To do this, we compare the cumulative distribution of similarity values for all pairs of nodes to that of nodes that have links. Figure 5.9 and Figure 5.10 compare the cumulative distribution of similarity values for pairs with links and all pairs for the AAN data and the TREC BLOG06 data respectively. We see from the figures that there is a clear tendency for similar nodes to exchange links or in other words nodes that exchange links tend to be similar and highly similar nodes tend to exchange links.

We also tried to predict each of the links in one type of networks based on the edges in the other type. For example, Figure 5.11 shows how many edges in the citation network can be predicted using the similarity network and vice versa for the AAN data. We may have several similarity networks based on the cutoff threshold we use to judge whether two nodes are similar or not. Hence, the figure shows the percentage of edges in the citation network that can be predicted by each of the similarity networks. It also shows the percentage of edges of each of the similarity networks that can be predicted using the citation network. We notice from the graph that we can predict up to 60% of citation edges using a similarity network with threshold of 0.1. This percentage decreases consistently as the threshold is increased. On the other hand, predicting edges in similarity networks with low threshold is not successful. It improves as the threshold increases but drops again for high thresholds.

We can also see similar behavior in Figure 5.12 which shows how much edges in the links network can be predicted using the similarity network and vice versa for the TREC BLOG06 data. The figure indicates that the prediction power of similarity networks decreases as the threshold increases. On the other hand, the percentage of edges predicted using the links network increases as the threshold increases.

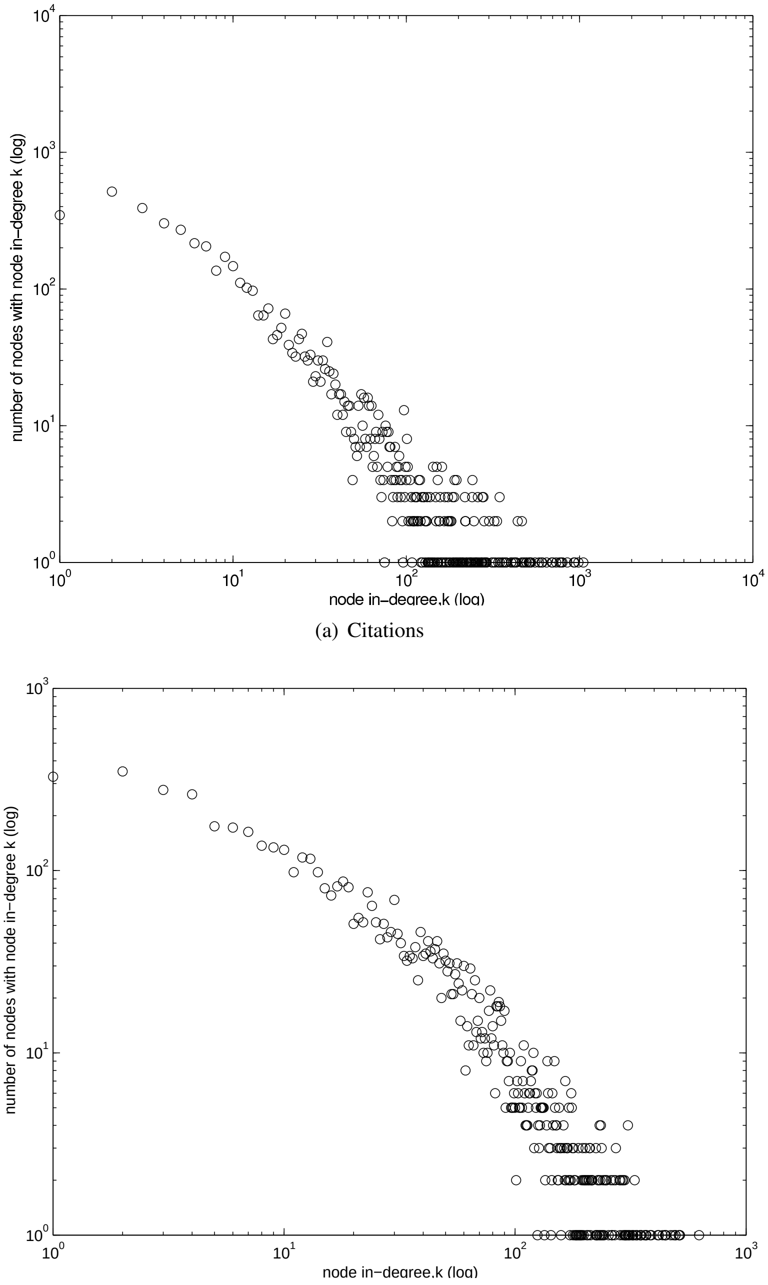
If we take into consideration that previous work suggested that the best value for the cut off threshold is usually in the interval $[0.1 - 0.3]$ [39], we can reach the conclusion that similarity based networks are doing much better in predicting link based edges than the other way around.

This set of experiments is suggesting that there is a strong correlation between links based networks and similarity based networks. It also suggests that both types of networks carry some information that is not available in the other network. However, it also suggests that the similarity based network is doing a much better job in predicting the link based networks than the other way around.

This suggests that we can use similarity based networks when explicit links between textual entities are not available or not enough. It also suggests that a hybrid approach that uses both networks would be much better than approaches that uses only one type and neglects the other.

5.7 Conclusions

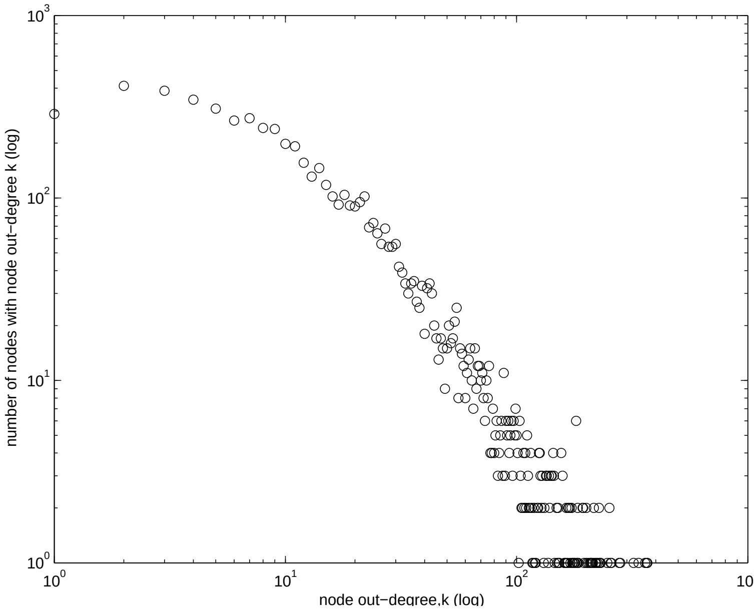
In this chapter, we described content similarity based networks and introduced the idea of using them to find implicit relations between entities that have some textual representation. We also studied the relation between link based networks and content similarity based networks. We used two datasets from two different domains. The first is the AAN dataset which contains a large number of papers from the ACL Anthology. The second is the TREC BLOG06 dataset which contains a large number of blog feeds and associated permalinks. We showed that there is a great consistency in the topological and temporal properties of the two types of networks over the two datasets. In the next chapters, we will describe methods that use content similarity based networks to identify static and dynamic salience of participants in a discussion.



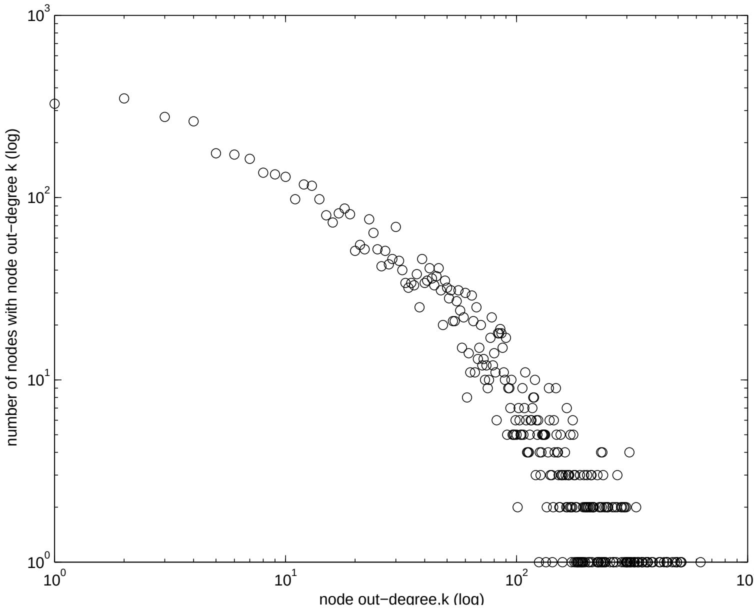
(a) Citations

(b) Similarity

Figure 5.1: The in-degree distributions for the AAN citation and similarity networks.

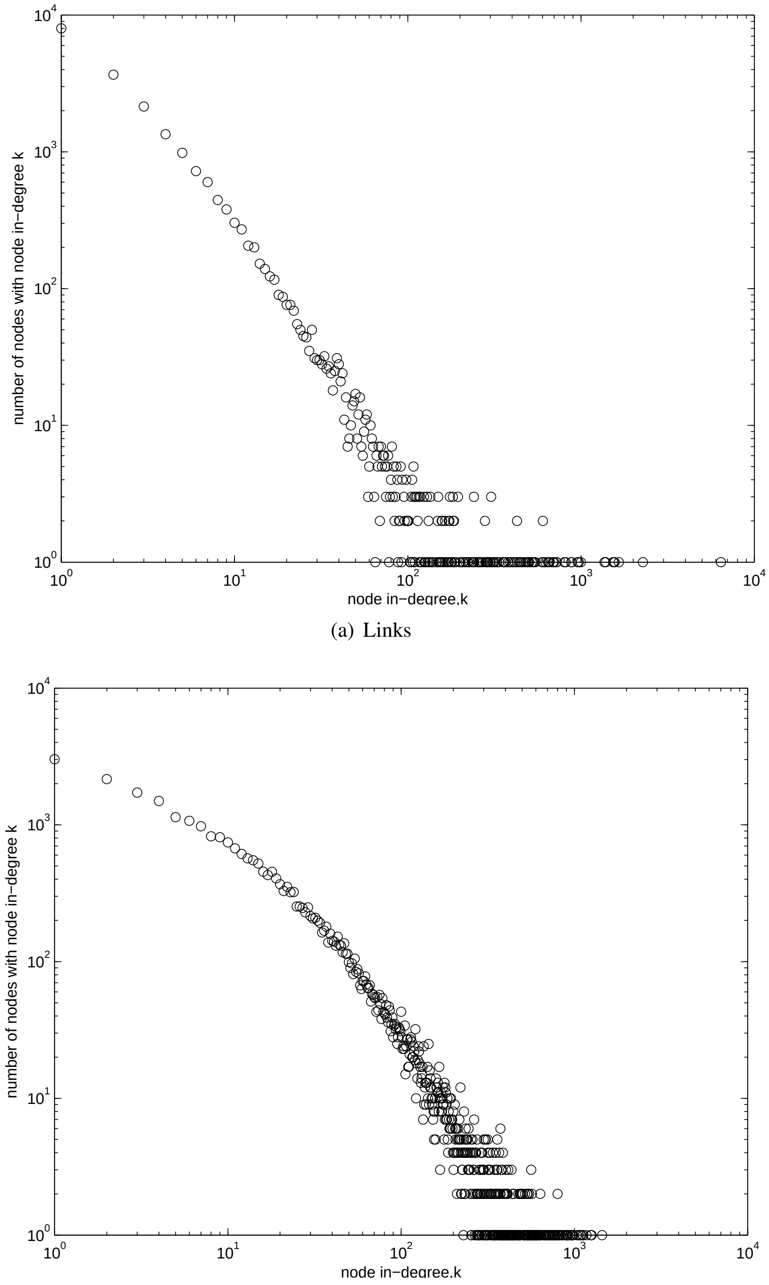


(a) Citations



(b) Similarity

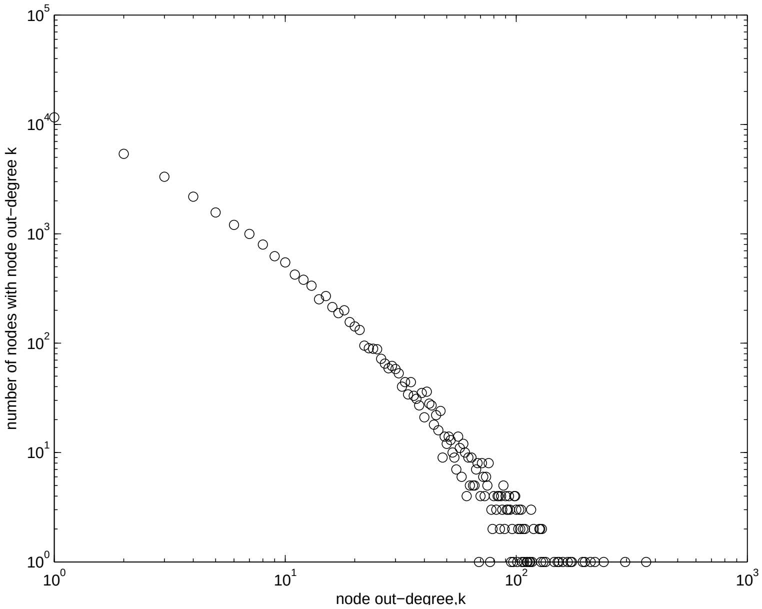
Figure 5.2: The out-degree distributions for the AAN citation and similarity networks.



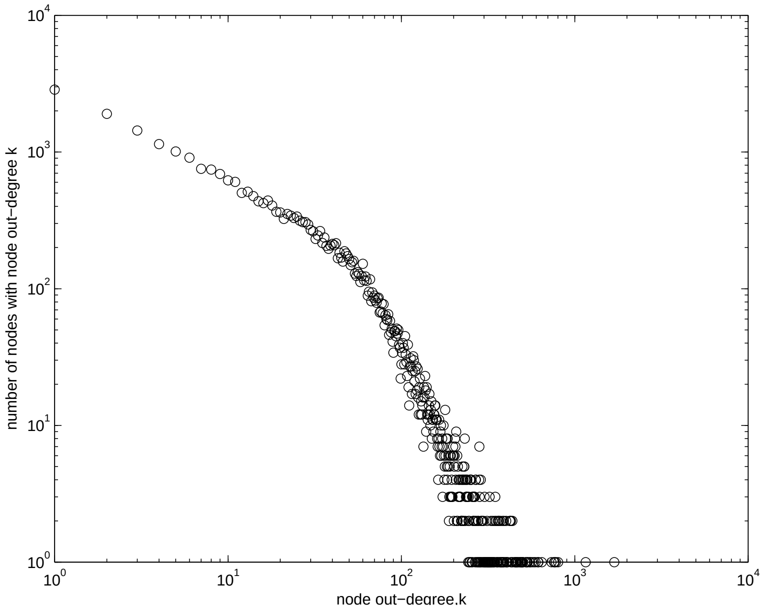
(a) Links

(b) Similarity

Figure 5.3: The in-degree distributions for the TREC BLOG06 links and similarity networks.

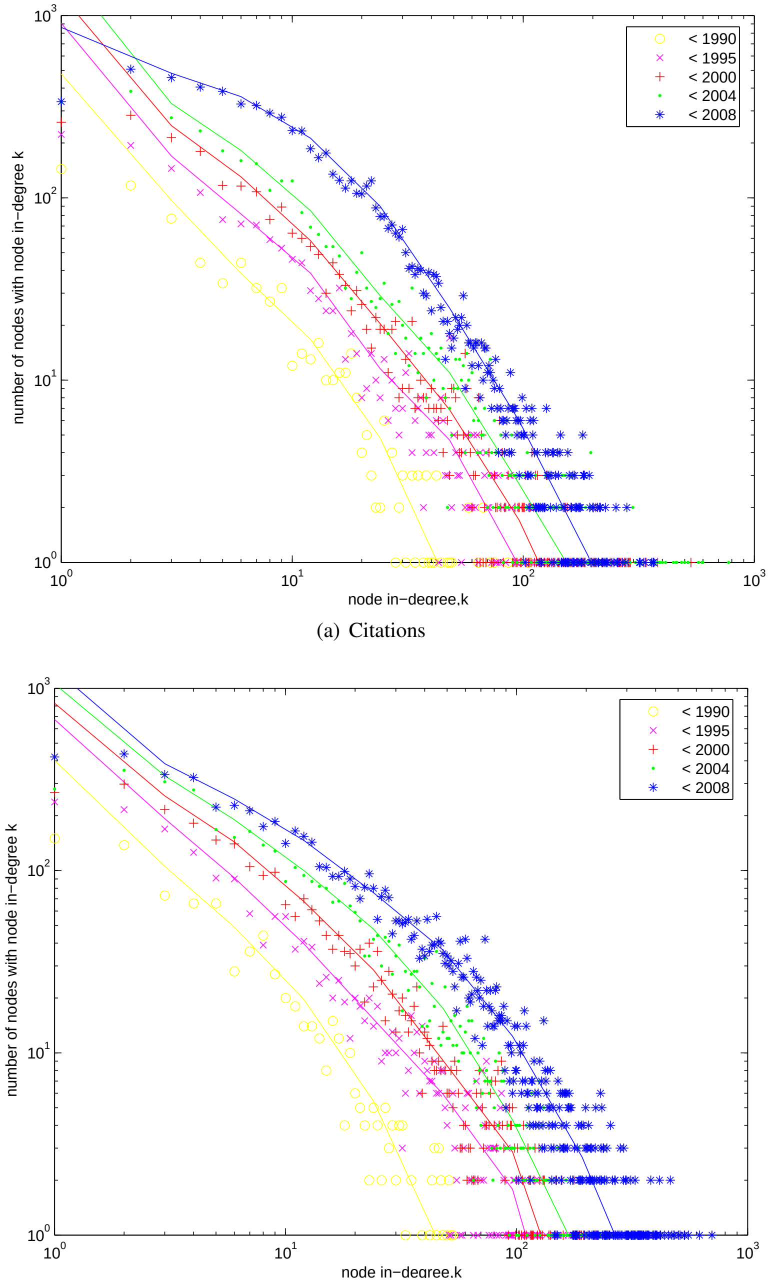


(a) Links



(b) Similarity

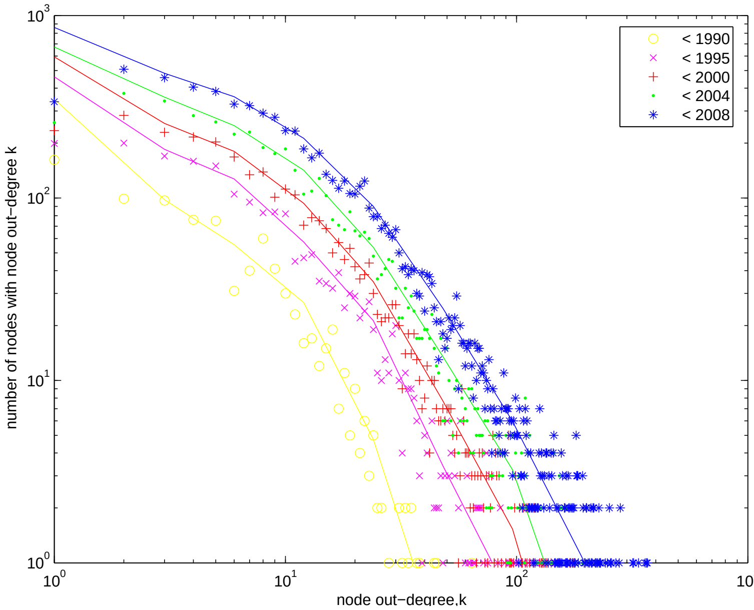
Figure 5.4: The out-degree distributions for the TREC BLOG06 links and similarity networks.



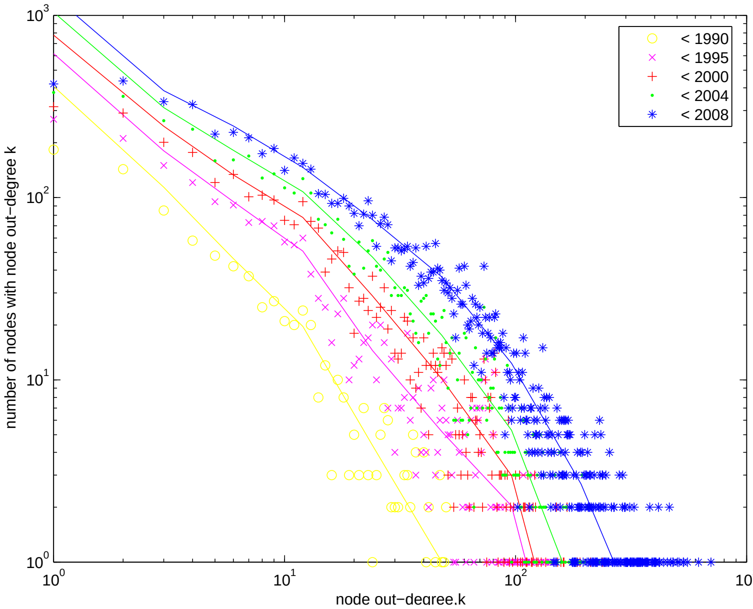
(a) Citations

(b) Similarity

Figure 5.5: Temporal evolution of the in-degree distributions for the AAN citation and similarity networks.



(a) Citations



(b) Similarity

Figure 5.6: Temporal evolution of the out-degree distributions for the AAN citation and similarity networks.

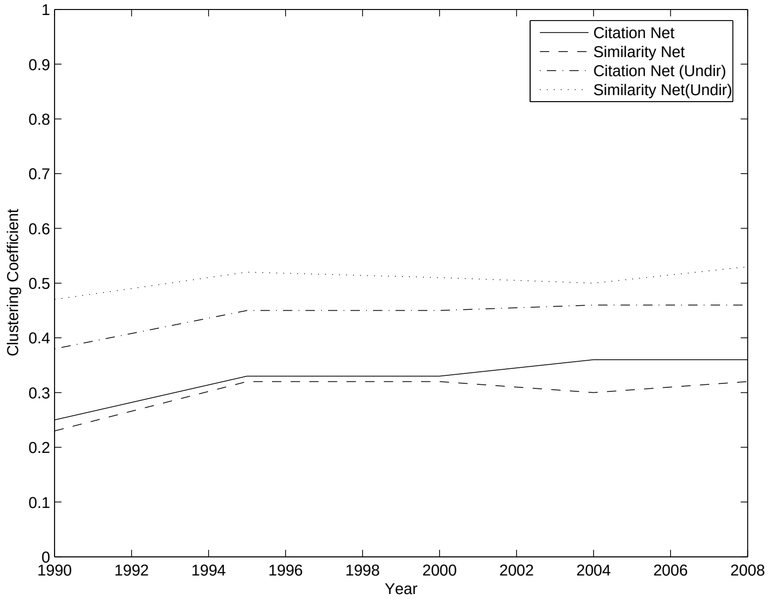


Figure 5.7: Temporal evolution of the clustering coefficients for the AAN citation and similarity networks.

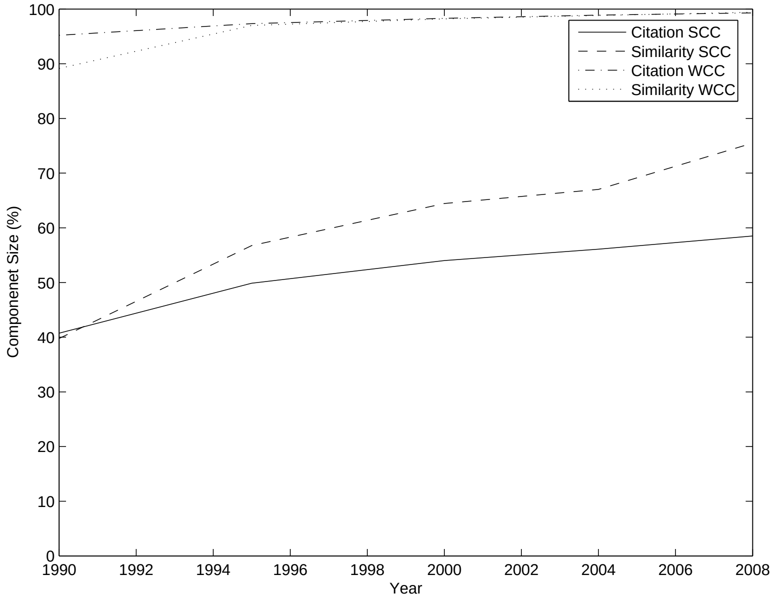


Figure 5.8: Temporal evolution of the size of the SCC and WCC for the AAN citation and similarity networks.

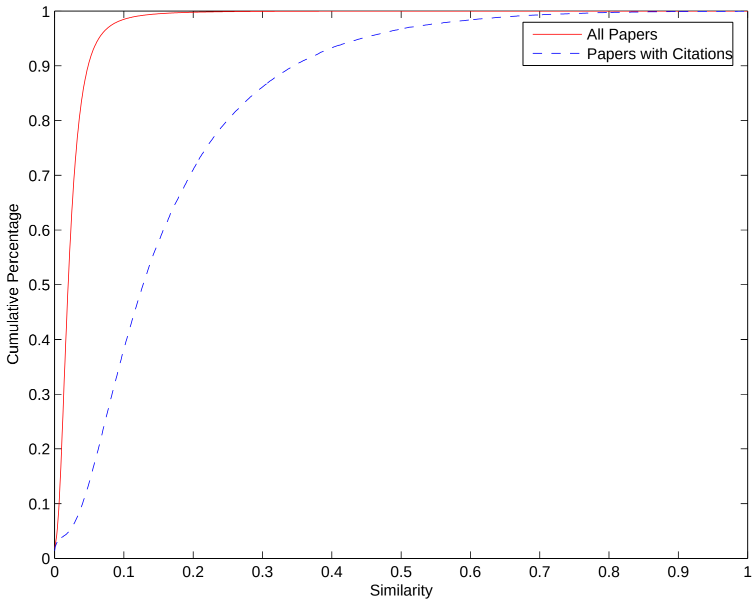


Figure 5.9: Correlation between citation and similarity - AAN data.

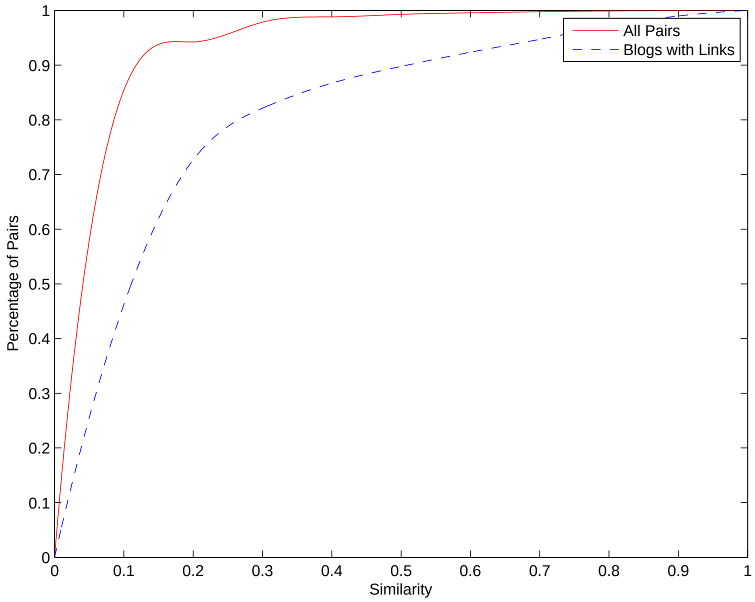


Figure 5.10: Correlation between citation and similarity - TREC BLOG06 data.

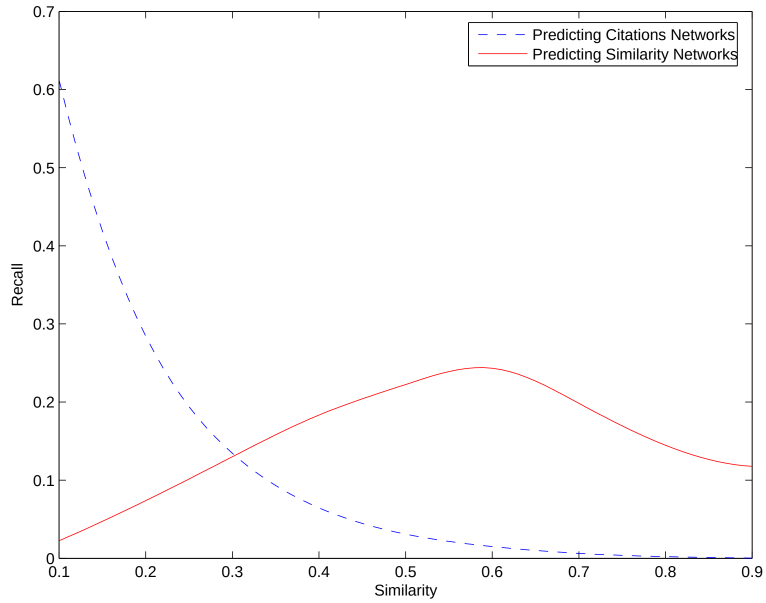


Figure 5.11: Predicting citation using similarity vs. predicting high similarity using citations - AAN data.

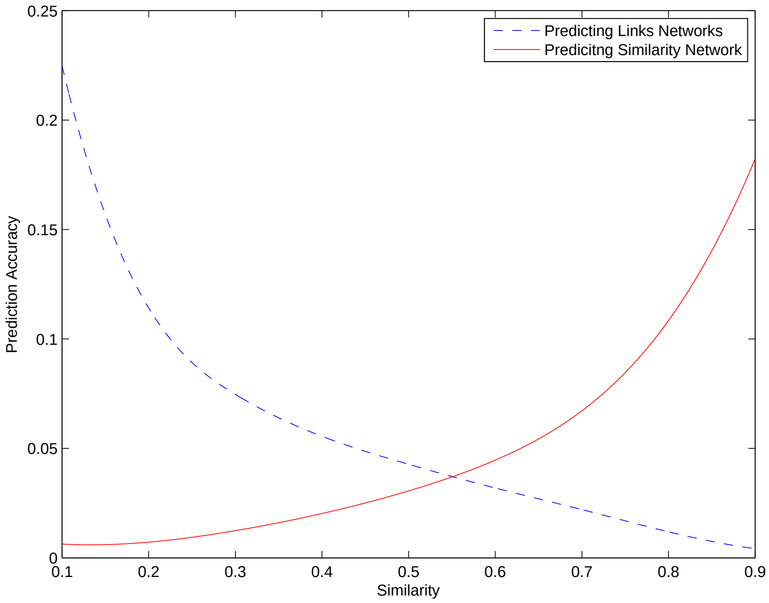


Figure 5.12: Predicting citation using similarity vs. predicting high similarity using citations - TREC BLOG06 data.

CHAPTER VI

Static Salience in Discussions

6.1 Introduction

In the previous chapter, we introduced the concept of content similarity based networks. We also compared their topological and temporal properties to explicit link based networks. In this chapter, we propose a graph based method based on random walks and lexical centrality for identifying salience of participants in a discussion.

Given a set of participants who have a set of posts, the proposed method uses text similarity between posts to select the most salient or representative set of participants. The method is based on the hypothesis that important or representative posts tend to be lexically similar to other important or representative posts. Hence, we may define the participant's score as a recursive function of the scores of its neighbors in a textual similarity network. This is equivalent to a random walk over the network of participants. The score of each participant is equivalent to the amount of time the random walk surfer spends on each node.

The proposed method takes diversity into consideration while assigning ranks to different participants. The proposed method discounts participants similar to highly ranked participants by decreasing their influence on the rest of the nodes in the graph. This allows other diverse node to receive higher ranks. The proposed method allows incorporating other measures that address the importance of participants regardless of their relation to other participants. This allows the method to bias the selection to participants with certain properties. For example, in the case of blog feeds, we can bias the selection to blogs with higher rate of posts, more fluent text, etc.

We use data from the blogosphere to show the potential of the proposed method. Blogs can be considered a means where people discuss particular topics by exchanging posts commenting on different aspects of the topic. The size of the collection of blogs on the World Wide Web has been lately exhibiting an exponential increase. Blogs are now one of the main means for spread of ideas and information throughout the Web. They discuss different trends, ideas, events, and so on. This gave rise to an increasing interest in analyzing the blogosphere by the Information Retrieval (IR) community, the Social Network Analysis community, as well as other communities.

A weblog (blog) is a website maintained by an individual who uses it as a self-publishing media by regularly publishing posts commenting on or describing some event or topic. Blogs made it easy for everybody to publish, read, comment, and share ideas. Blogs are different in style when compared to traditional Web pages. A blog is usually written by an individual person and organized in a set of posts. Blogs tend to be affected by each others. Some blogs start introducing new information and ideas that spread down to other blogs. One of the most interesting problems in the Blogosphere is how to provide the Internet users with a list of particularly important blogs with recurring interest in a specific topic.

There are several advantages of using the proposed method for identifying important blog feeds compared to other link popularity based algorithms. Link popularity based algorithms, like PageRank [111] and HITS [73], use the explicit hyperlink structure between pages to rank them. One of the reasons why link popularity based algorithms might not work well for blogs is the weakly linked nature of blog pages [85]. The small number of links per blog post, compared to traditional Web pages, results in a very sparse network especially when trying to build a network of blogs that are relevant to a particular topic.

Another reason is that blog posts have a relative short life span when compared to traditional Web pages. For example, a good blog post may eventually receive a lot of good links pointing to it. However, we will not able to assess how good this post is until it receives those links and that will need time. By that time, users may not be interested in it anymore. The proposed method alleviates those problems by using textual similarity between blog posts to measure the authority of blogs feeds.

The rest of this chapter will proceed as follows. We present an overview of related work in Section 6.2. In Section 6.3, we describe our graph based approach for blog ranking. Section 6.4 describes how diversity can be taken into consideration during the ranking process. Adding priors to capture the base quality of nodes is described in Section 6.5. Section 6.7 describes experiments and results. Finally, we present conclusions in Section 6.8.

6.2 Related Work

Using blogs as a case study for showing the potential of the proposed method makes our work related to a line of research that has focused on blog ranking and distillation. The problem of weblog (blog) ranking or selecting which blogs to read has been lately receiving increasing attention. The problem is different from traditional document ranking in ad-hoc information retrieval in several ways [8]. For example, a blog is not a single document, rather it is composed of a collection of documents. Moreover, blog queries always represent an interest in a particular topic, rather than a passing information need [8]. Hence, specific methods have been developed to target this problem.

Leskovec et al. [85] present a method detecting outbreaks in a network. To detect outbreaks, they select nodes in a network that can detect the spread of a virus or information as quickly as possible. They propose an algorithm that can detect outbreaks based on optimization of submodular function. They use their method to find the best locations for sensor placement in water distribution networks to quickly detect contaminants. They also extend their method to detecting which blogs one should read to catch all important stories.

TREC 2007 Blog track [91] presented a new blog distillation track which is quite related to the problem we are addressing. In their formulation, the problem is to find blog feeds with a principal, recurring interest in X, where X is some information need expressed as a query [91]. The input to such a system is a query and the output is a ranked list of blog feeds. The CMU system [36] achieved the best performance in the 2007 track. Their approach depends on indexing both feeds and individual posts. They also use a Wikipedia-based query expansion approach to improve the results.

Lin et al. [88] present a semi-supervised method for classifying political blogs in a blog network into liberal or conservative and ranking them within each predicted class. They use PageRank [14] to determine the importance or authority of blogs. However, they modify the algorithm such that authority scores propagate only within blogs belonging to the same community.

Arguello et al. [8, 36] present a system for blog ranking and recommendation. Their system compares different blog retrieval models that view either the blogs or the posts as retrieval units. They also use Wikipedia for query expansion to further improve queries.

Java et al. [62] study the feeds subscribed by a set of publicly listed Bloglines users. They use the subscription information to come up with feed topics and feeds that matter for particular topics. Using the Bloglines subscription data, they identify feeds that are popular to a given topic. Topics are approximated by the folders names and merging related folders together.

Song et al. [124] proposes a method for identifying opinion leaders in the Blogosphere. They define opinion leaders as those who bring in new ideas and opinions. They rank blogs according to how novel the information they contribute to the network.

Marlow et al. [93] use blogroll links and permalinks to predict the authority and influence of blogs. The study shows that hyperlinks between blogs can be used to track influence; however it does not well when the problem is restricted to finding important blogs within a particular topic.

Several methods have been proposed for identifying the most central nodes in a network. Some of the most popular methods, for measuring centrality of nodes in a network, are degree centrality, closeness, and betweenness [109].

Another powerful method for measuring centrality in a network is eigenvector centrality. This method has been successfully applied to several types of networks. For example it has been used to measure centrality in hyperlinked Web pages networks [14, 73], lexical networks [39, 99, 78, 79], and semantic networks [100].

6.3 Static Salience Scores

In this section, we describe how to assign relative weights or ranks to a set of participants in a debate or a discussion based on the content of their posts.

The main hypothesis behind this method is that important or representative posts tend to be lexically similar to other important or representative posts. Hence, we can use text similarity to link posts or participants to each other. Using textual similarity to link posts allows us to solve several problems that link popularity based algorithms cannot solve. For example, link popularity based algorithms cannot be applied to problems that do not have any link structures like a discussion between a set of speakers. They might not work well for other domains like blogs because of the weakly linked nature of blog pages [85].

Consider two posts p, and q such that p and q are lexically similar to each other. That will result in a link between p and q. This link is suggesting that p and q share a common topic of interest. And it is also suggesting that they may have been affected by each others and that the textual similarity is a way of conferring authority between them. Note that this is different from hyperlink based authority where p may simply ignore to add a link to q or add a non informative link to q based on some link-exchange agreement. Another advantage is that the assessment of q's quality is independent of the textual content of q. Hence, this assessment is completely out of q's control which would make the technique more immune to spamming.

Given a content-based network connecting participants in a discussion, we can define the importance score of a participant recursively in terms of the scores of other similar participants. This can also be implemented in a lower level of granularity where nodes of the graph represent posts, rather than a set of posts for some participant. In this case, each post is assigned an importance score. The importance score of a participant can then be calculated by taking the average of the scores of all his posts. In the former case where nodes represent participants, the importance score is directly assigned to the participant.

When building a network of posts, we only consider two posts similar if they belong to two different participants. This makes sure that posts within the same feed are not connected. Hence, a participant cannot gain credit by having several posts similar to each others.

The recursive definition of the score of any participant b in the participants network is given by:

$$p(b) = \sum_{t \in adj[b]} \frac{p(t)}{deg(t)} \tag{6.1}$$

where $deg(t)$ is the degree of node t, and $adj[b]$ is the set of all nodes adjacent to b in the network. This can be rewritten in matrix notation as:

$$\mathbf{p} = \mathbf{p}\mathbf{B} \tag{6.2}$$

where $\mathbf{p} = (p(b_1), p(b_2), \ldots, p(b_N))$ and the matrix $\mathbf{B}$ is the row normalized similarity matrix of the graph

$$\mathbf{B}(i,j) = \frac{\mathbf{S}(i,j)}{\sum_k \mathbf{S}(i,k)} \tag{6.3}$$

where $\mathbf{S}(i,j) = sim(b_i, b_j)$. Equation (6.2) shows that the vector of salience scores $\mathbf{p}$ is the left eigenvector of $\mathbf{B}$ with eigenvalue 1.

The matrix $\mathbf{B}$ can be thought of as a stochastic matrix that acts as the transition matrix of a Markov chain. An element $\mathbf{X(i,j)}$ of a stochastic matrix specifies the transition probability from state i to state j in the corresponding Markov chain. And the whole process can be seen as a Markovian random walk on the network. To help the random walker escape from periodic or disconnected components, we follow [14] that suggested reserving a small escape probability at each node that represents a chance of jumping to any node in the graph, making the Markov chain irreducible and aperiodic, which guarantees the existence of the eigenvector.

Brin and Page [14] suggest reserving a small escape probability at each node that represents a chance of jumping to any node in the graph, making the Markov chain irreducible and aperiodic, which guarantees the existence of the eigenvector.

Equation (6.2) can then be rewritten, assuming a uniform escape probability, as:

$$\mathbf{p} = \mathbf{p}[d\mathbf{U} + (1-d)\mathbf{B}] \tag{6.4}$$

where N is the total number of nodes, $\mathbf{U}$ is a square matrix with $\mathbf{U}(i,j) = 1/N$ for all i, j, and d is the escape probability chosen in the interval $[0.1, 0.2]$ [14].

6.4 Diversity Ranking

Suppose we already identified a node x as the most important node in the graph, and now, we would like to identify the second important node such that it is important and at the same time as diverse as possible with respect to the first selected node. The importance of a node is usually calculated as:

$$p(b) = \sum_{t \in adj[b]} \frac{p(t)}{deg(t)} \tag{6.5}$$

where $p(u)$ is the importance of node u, $adj[u]$ is the set of nodes that are adjacent to u, and $deg(v)$ is the degree of the node v.

The problem with this formula is that nodes very similar to x will benefit from their connection to x and hence receive high ranks. However those nodes are probably quite redundant and do not have any new information.

To solve this problem, we will modify the above formula as follows:

$$p(b) = d(b) \sum_{t \in adj[b]} \frac{p(t)}{deg(t)} \tag{6.6}$$

$$d(b) = 1 - sim(b, x) \tag{6.7}$$

Where $d(b)$ is a discounting factor to penalize nodes that are similar to the already selected node, and $sim(b, x)$ is the similarity between nodes b and x.

How about the case of more than one important node? For example, what should we do when we are trying to select the third or the fourth or even the n$^{\text{th}}$ node? In this case, x is set of nodes rather than a single node. The formula for calculating $d(b)$ can be modified as follows:

$$d(b) = 1 - max_{\forall x_i} sim(b, x_i) \tag{6.8}$$

In this way, each node is penalized with respect to the closest node to it that was already selected.

6.5 Adding Priors

Node importance calculated from the lexical network is a good measure for determining importance of nodes. However, it might not be the only attribute that affects node quality. Other attributes that are more related to the node itself, rather than to its position in the network, might also be involved. If we consider the blogosphere, we may imagine other quality attributes like text fluency (e.g., n-gram model probabilities), formatting, frequency of updates, use of particular vocabulary, average length of posts, number of posts, etc.

To incorporate these attributes into our graph based approach, we propose a modification to the proposed method to allow it to take initial node quality priors into consideration.

Assume we have a certain node quality measure that uses one or more of the features we mentioned above. Let's define a vector Q of priors where $Q = \{q_1, \ldots, q_n\}$, and n is the number of nodes in the network. Before going on we further normalize Q such that the sum of all of its entries is 1.

$$q_{i_{norm}} = \frac{q_i}{\sum_j q_j} \tag{6.9}$$

Let us also define a trade-off factor β, where $0 \leq \beta \geq 1$, that controls the weight we assign to the priors versus the weight we assign to the network based scores.

Now we can redefine the formula as follows:

$$p(b) = (1 - \beta) \sum_{t \in adj[b]} \frac{p(t)}{deg(t)} + \beta * q_{norm} \tag{6.10}$$

Several attributes can be used toward calculating the initial quality prior. Each of those attributes may be used to bias the solution toward a specific property. As mentioned earlier, we can use priors to favor nodes with higher number of posts, more fluent text, etc.

6.6 Combining Link based and Similarity based Networks

This chapter suggests that content similarity based networks can be used as an alternative for finding implicit links using measures of textual similarity. This is very useful in weakly linked domains, and in domains where explicit link structures are completely absent. In the former case, existing explicit links can be used along with implicit links based on content. There are several methods by which we can combine the two networks, we propose one such method here. We define a continuous spectrum of networks that results from combining the two networks as follows:

$$N = \beta * N_1 + (1 - \beta) * N_2 \tag{6.11}$$

The weight of each edge in the new network is simply a weighted average of the normalized weights of that edge in the two networks. If an edge does not exist, it is assumed that its weight is zero. We can either use a cut off threshold to produce a binary version of the network, or use it as is and take edge weights into consideration when doing any sort of analysis.

6.7 Experiments

6.7.1 Data

We used two large test collections through our experiments, the BLOG06 dataset created by the University of Glasgow [90], and the UCLA Blogocenter dataset [122].

BLOG06 is a TREC test collection, created and distributed by the University of Glasgow. It contains a crawl of Feeds, and associated Permalink and homepage documents (from late 2005 and early 2006) [90]. The dataset contains 100,649 feeds covering an 11

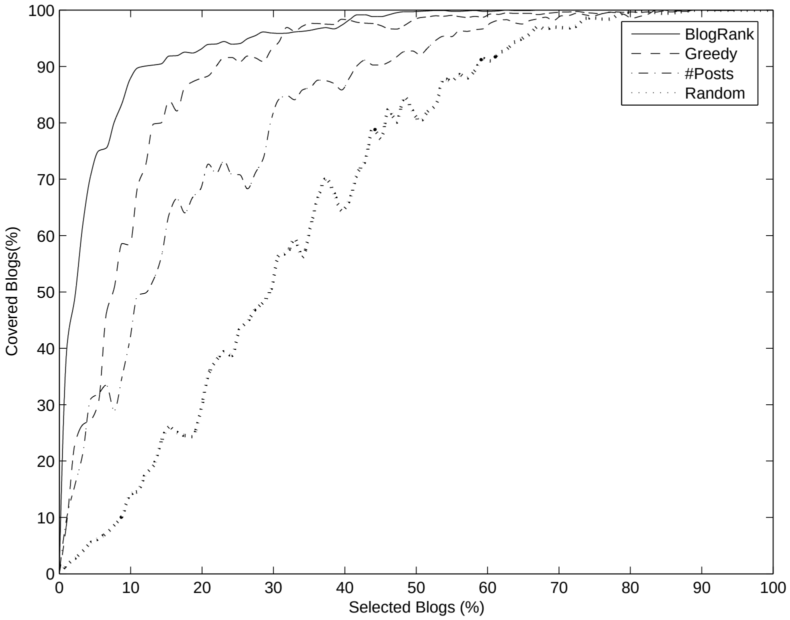


Figure 6.1: Percentage of covered blogs vs. percentage of selected blogs for BlogRank order, greedy algorithm order, number of posts order, and random order - Topic: Global Warming.

weeks period. Permalink documents and blog homepages were also collected. The dataset contains a total of 3,215,171 permalink documents and 324,880 home page documents [90]. To make the collection more realistic, a list of known spam blogs was also included. 17,969 spam blogs were added causing the spam component to form a reasonable component of the collection [90].

The other dataset is a massive dataset built by the The Blogocenter group at UCLA. They have been retrieving RSS feeds from the Bloglines, Blogspot, Microsoft Live Spaces, and syndic8.com aggregators for the past several years. The dataset contains over 192 million blog posts [122].

6.7.2 Evaluation Metrics

We borrow an idea from the studies of the spread of influence in social networks to evaluate our method's results. Diffusion models for the spread of an idea in a social network consider each node as either active or inactive. Inactive nodes tend to become active

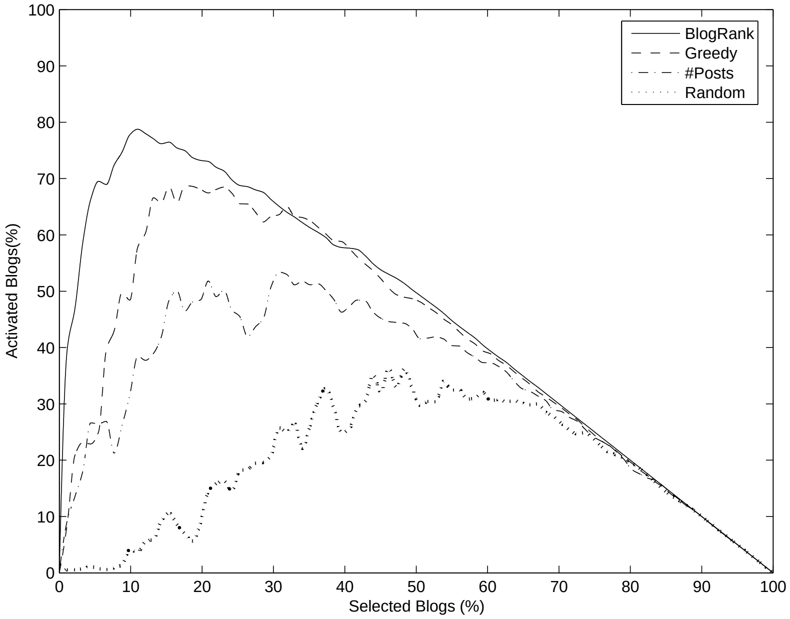


Figure 6.2: Percentage of activated blogs vs. percentage of selected blogs for BlogRank order, greedy algorithm order, number of posts order, and random order - Topic: Global Warming.

as more of its neighbors become active. One of the earliest models, that captures this process, was proposed in [47]. Several other models were then presented to capture the same process. At the core of all those models lies the Linear Threshold Model [70]. This model assumes that the influence of each node by any of its neighbors is a function of the weight of the edge connecting the node to that neighbor. The diffusion process starts by a set of active nodes and a threshold θ_v for each node v selected uniformly at random. At each step, any node v, for which the sum of the weights of the edges connecting it to its active neighbors exceeds its threshold, is activated.

$$\sum_{w \in adj[v]} b_{v,w} \geq \theta_v \tag{6.12}$$

The thresholds θ_v are randomly selected to reflect the lack of knowledge of their values [70].

In this section, we show how the proposed method can be used to identify important blog feeds. The top nodes in a ranked list of important blog feeds, interested in a particular topic, could summarize or cover most of the information content in a larger set of blogs. Hence, we can evaluate the ranked list by estimating the coverage of the selected nodes using diffusion models. Each node (blog) in the network will be considered as either active or inactive. An active node is a node that has been covered by the small selected set M. An inactive node is a node that has not yet been covered. We also assume that the node's tendency to become covered increases as more of its neighbors become covered. Or in other words, a node's information content is more likely to become covered as more and more of the information contents of its neighbors are covered.

We can use this model to evaluate the selection of the smaller set M with respect to the bigger set N. The quality of a selected set of feeds is evaluated by the number of nodes that become active in the blogs network when the selected feeds are initially designated as active. The output of the proposed method is a ranked set of blogs such that a blog with higher rank is a better candidate for inclusion in the selected set than a blog with a lower rank. Given a ranked list of blogs R, we evaluate all subsets M_i where $M_i = \{R_1..R_i\}$ for all i in $[1 \text{ to } |R|]$.

Due to the randomness inherent in selecting the nodes activation threshold, we repeat the evaluation n times and report the average of the results of the n runs. We used $n = 100$ in most of the experiments reported below.

6.7.3 Comparison to Other Systems

We compare our method to several other methods of blog selection. The first method is a simple method that selects blogs for inclusion in the selected set uniformly at random.

The second method is one of a family of heuristic selection techniques that try to find the most popular blogs by using some fixed "goodness" criteria. The criterion we use here is the number of posts in a blog. Whenever we want to select a subset of blogs M with

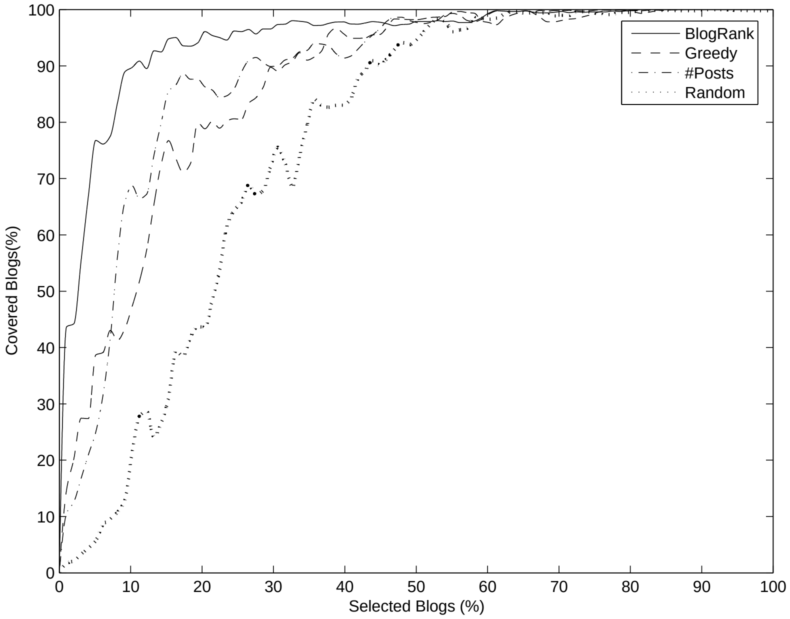


Figure 6.3: Percentage of covered blogs vs. percentage of selected blogs for BlogRank order, greedy algorithm order, number of posts order, and random order - Topic: iPhone.

$|M|$ blogs, we select the $|M|$ blogs with the highest number of posts.

The third method is based on a greedy algorithm that uses hill climbing search. The method greedily adds the node that maximizes the marginal gain to the selected set. Initially we start with an empty set $M_0 = \phi$. At each step i, we add the node that maximizes the marginal gain $b = \arg\max_{\text{non-selected blogs}} C(M_{i-1} \cup b) - C(M_{i-1})$ where $C(M_i)$ is the quality of the subset M_i measured by the number of nodes it covers in the blog network. So the method tries to maximize the gain in the evaluation metric described in the previous section.

6.7.4 Results

We compare the performance of the proposed method to a random selection method, a heuristic selection method and a greedy method. Figure 6.1 shows the percentage of covered blogs vs. percentage of selected blogs for the proposed method (which we will refer to as BlogRank [1]), the greedy method, the heuristic (number of posts) method, and

[1] The method could be applied to several other domains in addition to blogs as explained earlier.

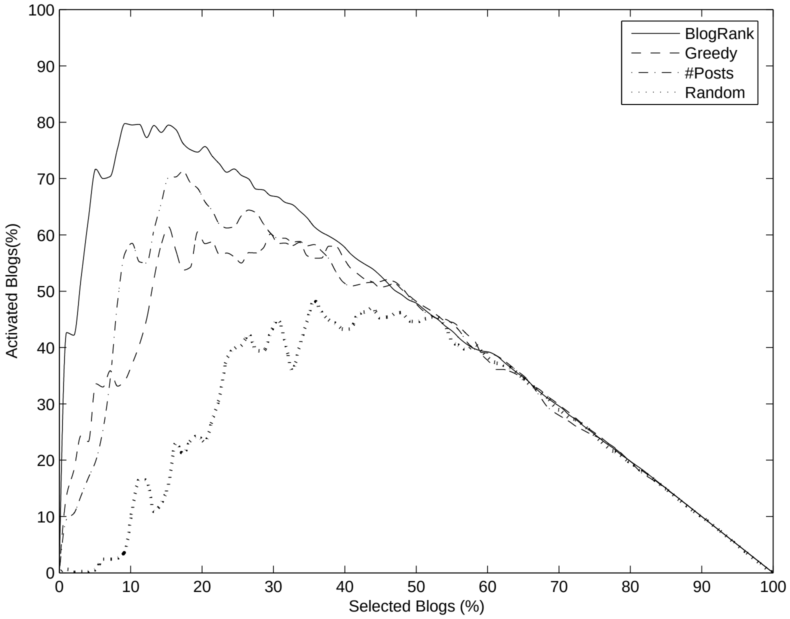


Figure 6.4: Percentage of activated blogs vs. percentage of selected blogs for BlogRank order, greedy algorithm order, number of posts order, and random order - Topic: iPhone.

a random method for the “Global Warming” topic. Figure 6.2 compares the performance of the same methods on the same topic with respect to the percentage of activated blogs vs. percentage of selected blogs. The difference between Figure 6.1 and Figure 6.2 is that Figure 6.2 does not count the selected blogs when calculating the number of covered blogs (i.e. it only shows blogs that were not selected yet were covered by other selected nodes). Figure 6.3 and Figure 6.4 show similar results for the “iPhone” topic.

We notice that random blog selection performs the worst. We also notice that heuristic blog selection based on the number of posts does not perform very well either. Selection based on the greedy based method outperforms random and heuristic selection. The figures also show that the proposed method outperforms all other methods with a considerable amount of improvement.

These results show that random selection and heuristic based selection for blogs that summarize or cover the information content on a particular topic do not work well. Greedy methods are better than random and heuristic methods; however they are very computa-

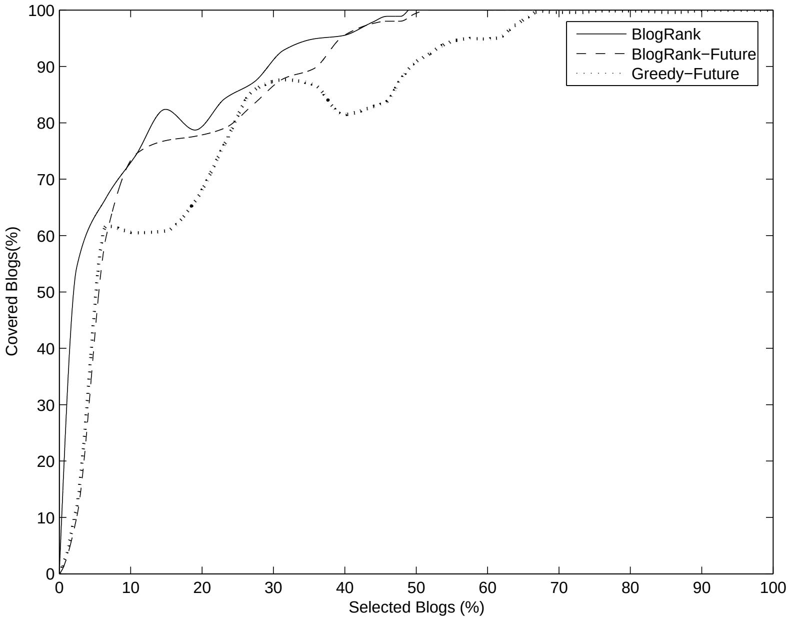


Figure 6.5: Percentage of activated blogs vs. percentage of selected blogs for BlogRank order (learned from all data and evaluated on future data), BlogRank-Future order (learned from history data and evaluated on future data), and Greedy-Future order (learned from history data and evaluated on future data) - Topic: Global Warming.

tionally costly ($O(|V|^4)$, where $|V|$ is the number of nodes). The proposed method outperforms all other methods and at the same time it is much faster than the greedy method.

6.7.5 Generalization to Future Data

The quality of a blog in summarizing or covering the information content in the Blogosphere with respect to some particular topic may change with time. Hence, we need to evaluate how good our method is in predicting the quality of a particular blog. We split our data into two smaller datasets. The first part represents “History” and covers the first half of the time period. The second part represents “Future” and covers the second half. We now have two networks for each topic (history and future). We use our method to rank blogs using only the history network and evaluate it on the future network. We also use our method to rank blogs using all the data set, but evaluate it on the future network. Finally we use the greedy method to rank blogs using only the history network and evaluate it on

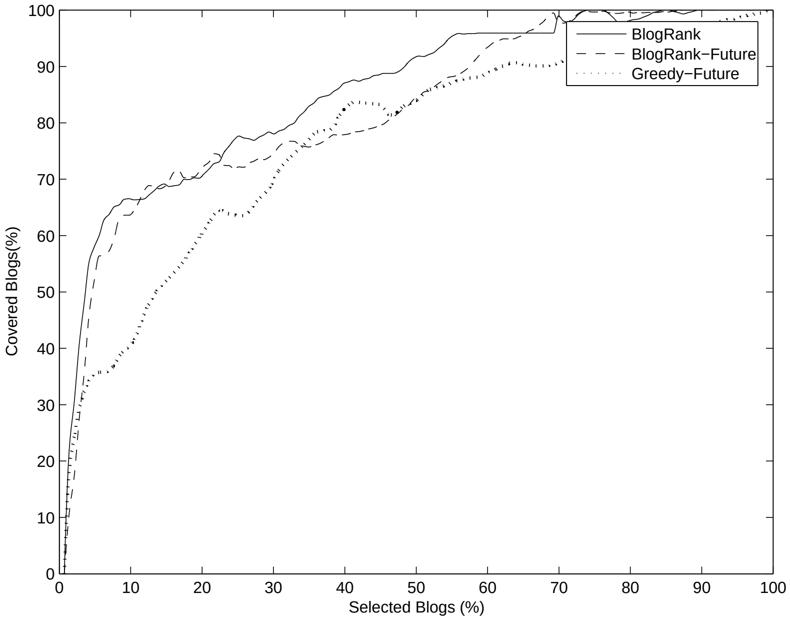


Figure 6.6: Percentage of activated blogs vs. percentage of selected blogs for BlogRank order (learned from all data and evaluated on future data), BlogRank-Future order (learned from history data and evaluated on future data), and Greedy-Future order (learned from history data and evaluated on future data) - Topic: iPhone.

the future network.

Figure 6.5 compares the performance of the proposed method when tested on known data, the proposed method when tested on unknown future data, and the greedy method when tested on unknown future data for the “Global Warming” topic. Figure 6.6 shows the same comparison for the “iPhone” topic. We notice that the gaps between the curves of the proposed method evaluated on known and unknown data is small. This suggests that the proposed method generalizes well for the future data. On the other hand, we see that the greedy method seems to overfit when evaluated on the future data.

6.7.6 Other Experiments

All the experiments reported in this section use the cosine similarity metric to measure text similarity. This is an arbitrary choice and any other text similarity measure may be used. For example, we tried using the KL divergence to measure text similarity. We

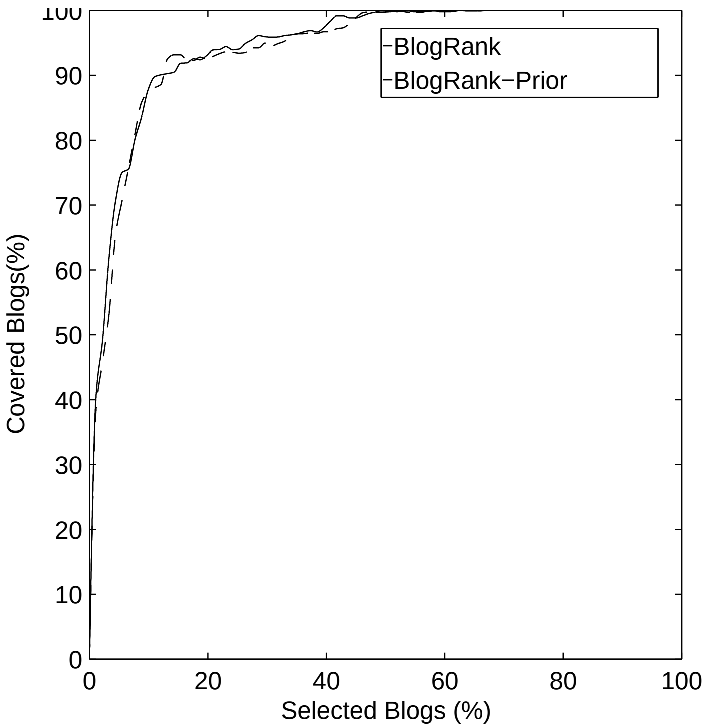


Figure 6.7: Percentage of covered blogs vs. percentage of selected blogs for BlogRank order and BlogRank with priors order - Topic: Global Warming.

compared the performance of the method when using cosine similarity and KL based similarity and the difference in performance was negligible. One advantage that cosine similarity has over KL based similarity is that it is a symmetric measure which reduces the number of similarity measures by half.

Another experiment that we performed was using priors to bias the blog selection toward some properties. For example, we tried using the number of posts as a prior. This led to a solution with almost the same quality as shown in Figure 6.7. Figure 6.8 compares the average number of posts for the two solutions with respect to the percentage of selected blogs. When we examine this figure, we notice that the new solution has larger average number of posts, especially for selected set with smaller sizes. This difference decreases till they become equal when all blogs are selected.

This shows that we can bias our solution to a specific property using the initial priors

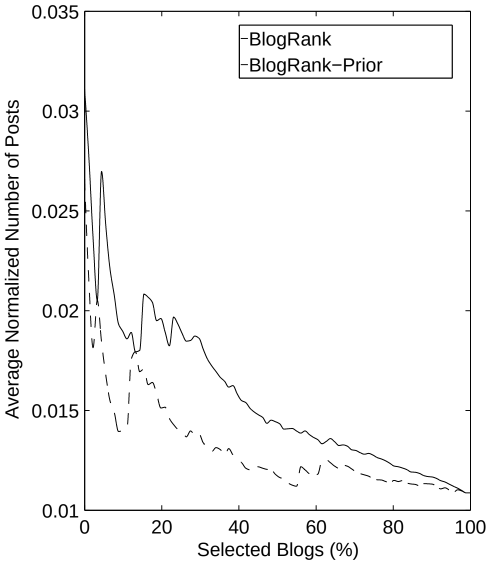


Figure 6.8: Average normalized number of posts vs. percentage of selected blogs for BlogRank order and BlogRank with priors order - Topic: Global Warming.

without losing much of the solution quality. For example, we can bias our solution to blogs with larger number of posts, smaller number of posts, longer posts, fluent text (as measured by language model scores), etc.

6.7.7 Combining Links and Content

Given a collection of blog posts, we built both a link based network and a content similarity based network. In the link based network, an edge is added if there is a link from one post to the other. In the similarity based network, two posts are linked if they are textually similar. Edges point from the newer to the older posts based on their dates. Given a posts network, we can construct a blogs network by adding an edge between blogs A, and B, if an edge exists between two posts such that the first belongs to A and the second belongs to B.

Given the two network, we compute the importance score as described earlier in this chapter. Figure 6.9 shows the precision at different ranks for the similarity based network, link based network and a combination of them (at $\beta = 0.5$). The figure shows that using

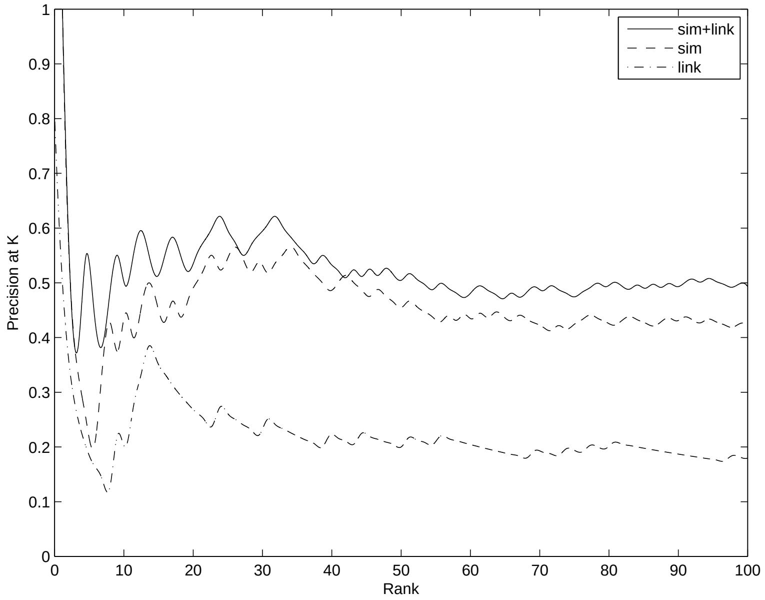


Figure 6.9: Precision at different ranks for a similarity based network, a link based network, and a combined network.

the content based network yields better results that the link based one. This is due in part to the sparse nature of the line based network when compared to the content similarity based network. On the other hand, edges in the link based network are important because they explicitly indicate some sort of authority. Hence, we notice that combining the link based and the content similarity based networks gives us better results when compared to the cases where only one network is used.

6.8 Conclusions

This work presented a stochastic graph based method for selecting the most salient participants in a discussion. The approach is based on lexical similarity and random walks. The proposed method takes diversity into consideration when measuring importance by penalizing nodes quite similar to already selected nodes. It may also benefit from additional initial priors to bias the selection toward nodes with a specific property such as text fluency. We evaluated our methods using data from the Blogosphere. We showed that the proposed method achieves promising results and outperforms other random, heuristic, and greedy selection methods, and that it performs well when tested on unseen future data.

CHAPTER VII

Dynamic Salience in Discussions

7.1 Introduction

The previous chapter proposed a method for ranking participants in a discussion based on the text they write. In this chapter, we extend this method to the case where discussions cover a long period of time. In this case, it is very important that our method is capable of capturing the dynamic nature of this data. One example of such data is speeches or debates. There are several sources of data that record speeches or entries in debates or discussions among a group of speakers or participants. Those include parliamentary records, blogs, and news groups. This data represents a very important and unexploited source of information that contains several trends and ideas. In any debate or discussion, there are certain types of persons who influence other people and pass information or advice to them. Those persons are often regarded as experts in the field or simply influential people and they tend to affect the ideas and rhetoric of other participants. This effect can be tracked down by tracking the similarity between different speeches. We can then imagine a debate with many people arguing about many different things as a network of speeches interacting with each other. We can then try to identify the most salient or important participants by identifying the most central speeches in this network and associating them with their speakers. When we have a large dataset of debates and conversations that expand over a long period of time, the salience of participants becomes a dynamic property that changes over time. To capture the dynamic nature of this process, the graph of speeches must evolve over time such that we have a different graph at each instance of time that reflects the interaction of speeches at this instant.

We apply our method to the *US Congressional Record*. The *US Congressional Record* documents everything said and done in the US Congress House and Senate. The speeches in this data set are made by a large number of people over a long period of time. Using political speeches as test data for the proposed method adds an extra layer of meaning onto the measure of speaker salience. Speaker salience of the Congress members can reflect the importance or influence in the US legislative process. The way salience scores evolve over time can answer several interesting questions like how the influence of speakers varies with majority status and change of party control. It can also allow us to study the dynamics of the relative distribution of attention to each topic area in different time periods.

The rest of this chapter will proceed as follows. Section 7.2 reviews some related work. In Section 7.3, we describe how the data can be clustered into different topic clusters. In Section 7.4, we describe our method for computing the salience of different participants in a discussion, we also describe how to the network of speakers varies over time. Section 7.5 describes the experimental setup. Finally, we present the conclusions in Section 7.6.

7.2 Related Work

Several methods have been proposed for identifying the most central nodes in a network. Degree centrality, closeness, and betweenness [109] are among the most known methods for measuring centrality of nodes in a network. Eigenvector centrality is another powerful method that that has been applied to several types of networks. For example, it has been used to measure centrality in hyperlinked web pages networks [14, 73], lexical networks [39, 99, 78, 79], and semantic networks [100].

The data we use for evaluating this method makes us close to a line of research interested in applying natural language processing techniques in the area of political science. Quinn et al. [116] introduced a multinomial mixture model to cluster political speeches into topics or related categories. In [114], a network analysis of the members and committees of the US House of Representatives is performed. The authors prove that there are connections linking some political positions to certain committees. This suggests that there are factors affecting committee membership and that they are not determined at random. In [129], the authors try to automatically classify speeches, from the US Congress debates, as supporting or opposing a given topic by taking advantage of the voting records of the speakers. Fader et al. [43] introduced MavenRank , which is a method based on lexical centrality that identifies the most influential members of the US Senate. It computes a single salience score for each speaker that is constant over time.

In this chapter, we introduce a new method for tracking the evolution of the salience of participants in a discussion over time. The work presented in this chapter has been published in [50]. The objective of this work is to dynamically rank speakers or participants in a discussion. The proposed method is dynamic in the sense that the computed importance varies over time.

7.3 Topic Clusters

Before applying the proposed method to a data set with speeches in multiple topics, we first need to divide the speech documents into topic clusters. We used the model described in [116] for this purpose. The model presented in this chapter assumes that the probabilities of a document belonging to a certain topic varies smoothly over time and the words within a given document have exactly the same probability of being drawn from a particular topic [116]. These two properties make the model different than standard mixture models [96] and the latent Dirichlet allocation model [13]. The model of [116] is most closely related to the model of [12], which presents a generalization of the model used by [116].

The output from the topic model is a $D \times K$ matrix $\mathbf{Z}$ where D is the number of speeches , K is the number of topics and the element z_{dk} represents the probability of the d^{th} speech being generated by topic k. We then assign each speech d to the k^{th} cluster where $k = \arg\max_j z_{dj}$. If the maximum value is not unique, one of the clusters having the maximum value is arbitrary selected.

7.4 Speaker Salience

In this section we review how to build a network of speeches and use it to identify speaker salience. The details of this part of our method are described in Chapter VI. We also describe how to generate different projections of the network at different times, and how to use those projections to get dynamic salience scores.

7.4.1 Computing Speaker Salience

As described in Chapter VI, a collection of speeches can be represented as a network where similar speeches are linked to each other. The proposed method is based on the premise that important speeches tend to be lexically similar to other important speeches, and important speeches tend to belong to important speakers. Hence given a collection of speeches and a similarity measure, we can build a network and define the centrality score of a speech recursively in terms of the scores of other similar speeches. Later, we can compute the salience of a speaker as the sum of the centrality measures of all his speeches.

To measure the similarity between two speeches, we use the bag-of-words model to represent each sentence as an N-dimensional vector of tf-idf scores, where N is the number of all possible words in the target language. A vector of term frequencies is used to represent each speech. Those term frequencies are weighted according to the relative importance of the given term in the cluster. The vectors representing speeches contain term frequencies (*tf*), and *topic-specific* inverse document frequency (*idf*). We calculated *idf* values specific to each topic, rather than to all speeches because the relative importance of words may vary from one topic to the other. The similarity between any two speeches is computed as the cosine of the angle between the corresponding tf-idf vectors.

The choice of tf-idf scores to measure speech similarity is an arbitrary choice. Some other possible similarity measures are edit distance, language models [78], or generation probabilities [40].

The recursive definition of the score of any speech s in the speeches network is given by

$$p(s) = \sum_{t \in adj[s]} \frac{p(t)}{deg(t)} \tag{7.1}$$

where $deg(t)$ is the degree of node t, and $adj[s]$ is the set of all speeches adjacent to s in the network. A small escape probability is used to represent a chance of jumping to any node in the graph, making the Markov chain irreducible and aperiodic.

7.4.2 Dynamic Salience Scores

We use the time stamps associated with the data to compute dynamic salience scores $P_T(u)$ that identify central speakers at some time T. To do this, we create a speech graph that evolves over time.

Let T be the current date and let u and v be two speech documents that occur on days t_u and t_v. Our goal is to discount the lexical similarity of u and v based on how far apart they are. One way to do this is by defining a new similarity measure $s(u, v; T)$ as:

$$s(u, v; T) = \text{tf-idf-cosine}(u, v) \cdot f(u, v; T) \tag{7.2}$$

where $f(u, v; T)$ is a function taking values in $[0, 1]$. If $f(u, v; T) = 1$ for all u, v, and T, then time is ignored when calculating similarity and $P_T(u) = P(u)$. On the other hand, suppose we let

$$f(u, v; T) = \begin{cases} 1 & \text{if } t_u = t_v = T, \\ 0 & \text{else.} \end{cases} \tag{7.3}$$

This removes all edges that link a speech, occurring at some time T, to all other speeches occurring at some time other than T and the ranking algorithm will be run on what is essentially the subgraph of documents restricted to time T (although the isolated speech documents will receive small non-zero scores because of the escape probability). These two cases act as the extreme boundaries of possible functions f: in the first case time difference has no effect on document similarity, while in the second case two documents must occur on the same day in order for them to be similar.

We use the following time weight functions in our experiments. In each case, we assume that the speeches represented by speech documents u and v have already occurred, that is, $t_u, t_v \leq T$. We will use the convention that $f(u, v; T) = 0$ if $t_u > T$ or $t_v > T$ for all time weight functions, which captures the idea that speeches that have not yet occurred have no influence on the graph at time T. Also define

$$\text{age}(u, v; T) = T - \min\{t_u, t_v\} \tag{7.4}$$

which gives the age of the oldest speech document from the pair u, v at time T.

- *Exponential:* Given a parameter $a > 0$, define

  $$f_{\text{exp},a}(u, v; T) = e^{-a\,\text{age}(u,v;T)}. \tag{7.5}$$

  This function will decrease the impact of similarity as time increases in an exponential fashion. a is a parameter that controls how fast this happens, where a larger value of a makes earlier speeches have a smaller impact on current scores and a smaller value of a means that earlier speeches will have a larger impact on current scores.

- *Linear:* Given $b > 0$, define

$$f_{\text{lin},d}(u, v; T) = \begin{cases} 1 - \frac{1}{b}\,\text{age}(u, v; T) \\ \quad \text{if age}(u, v; T) \leq b \\ 0 \quad \text{if age}(u, v; T) > b \end{cases} \tag{7.6}$$

  This function gives speech documents that occur at time T full weight and then decreases their weight linearly towards time $T + b$, where it becomes 0.

- *Boundary:* Given $d \geq 0$, define

$$f_{\text{bnd},d}(u, v; T) = \begin{cases} 1 \quad \text{if age}(u, v; T) \leq d \\ 0 \quad \text{if age}(u, v; T) > d \end{cases} \tag{7.7}$$

  This function gives speech documents occurring within d days of T the regular tf-idf similarity score, but sets the similarity of speech documents occurring outside of d days to 0. The case when $d = 0$ is one of the boundary cases explained above.

7.5 Experiments and Results

7.5.1 Data

We used the United States Congressional Speech corpus [103] in our experiment. This corpus is an XML formatted version of the electronic *United States Congressional Record* from the Library of Congress[1]. The *Congressional Record* is a verbatim transcript of the speeches made in the US House of Representatives and Senate and includes tens of thousands of speeches per year [103]. The data we used cover the period from January 2001 to January 2003.

[1] `http://thomas.loc.gov`

7.5.2 Experimental Setup

We used results from [116] to get topic clusters from the data, as described in Section 7.3. The total number of topics was 42. The average sized topic cluster had several hundred speech documents [116].

We applied the proposed method to topic clusters and ranked the speakers based on the centrality scores of their speeches. The graph nodes were speech documents. A speaker's score was determined by the average of the scores of the speeches given by that speaker. After comparing the different time weighting function as shown in Figure **??**, we decided to use the exponential time weight function for all the experiments discussed below. Exponential time weighting function decreases the impact of similarity as time increases in an exponential fashion. It also allows us to control the rate of decay using the parameter a.

7.5.3 Baseline

We compare the performance of our system to a simple baseline that calculates the salience of a speaker as a weighted count of the number of times he has spoken. The baseline gives high weight to recent speeches. The weight decreases as the speeches get older. The salience score of a speaker is calculated as follows:

$$BS(i) = \sum_{d} \gamma^{d_0 - d} * S_d^i \tag{7.8}$$

Where $BS(i)$ is the baseline score of speaker i, γ is the discounting factor, d_0 is the current date, and S_d^i is the number of speeches made by speaker i at date d. We used $\gamma = 0.9$ for all our experiments.

7.5.4 Results

One way to evaluate the dynamic salience scores is to look at changes when party control of the chamber switches. Similar to [27], we exploit the party switch made by Senator Jim Jeffords of Vermont and the resulting change in majority control of the Senate during the 107th Congress as a quasi-experimental design. In short, Jeffords announced his switch on May 24, 2001 from Republican to Independent status, effective June 6th, 2001. Jeffords stated that he would vote with the Democrats to organize the Senate, giving the Democrats a one-seat advantage and changing control of the Senate from the Republicans back to the Democrats. This change of majority status during the 107th Congress allows us to ignore many of the factors that could potentially influence dynamic salience scores at the start of a new congress.

On average, we expect committee chairs or a member of the majority party to be the most important speaker on each topic followed by ranking members or a member of the minority party. If our measure is capturing dynamics in the centrality of Senators, we expect Republicans to be more central before the Jeffords switch and Democrats becoming central soon afterwards, assuming the topic is being discussed on the Senate floor. We show that the proposed technique captures several interesting events in the data and also show that the baseline explained above fails to capture the same set of events.

Figure 7.1(a) shows the dynamic salience scores over time for Senator John McCain (R - Arizona) and Senator Carl Levin (D - Michigan) on topic 5 (Armed Forces 2) for the 107th Senate. McCain was the most salient speaker for this topic until June 2001. Soon after the change in majority status, a switch happened and Levin, the new chair of Senate Armed Services, replaced McCain as the most salient speaker. On the other hand, Figure 7.1(b) shows the baseline scores for the same topic and same speakers. We notice here that the baseline failed to capture the switch of salience near June 2001.

We can also observe similar behavior in Figure 7.2(a). This figure shows how Senate Majority Leader Trent Lott (R - Mississippi) was the most salient speaker on topic 35 (Procedural Legislation) until July 2001. Topic 35 does not map to a specific committee but rather is related to maneuvering bills through the legislative process on the floor, a job generally delegated to members in the Senate leadership. Just after his party gained majority status, Senator Harry Reid (D - Nevada) became the most salient speaker for this topic. This is consistent with Reid's switch from Assistant Minority Leader to Assistant Majority Leader. Again the baseline scores for the same topic and speakers in Figure 7.2(b) fails to capture the switch.

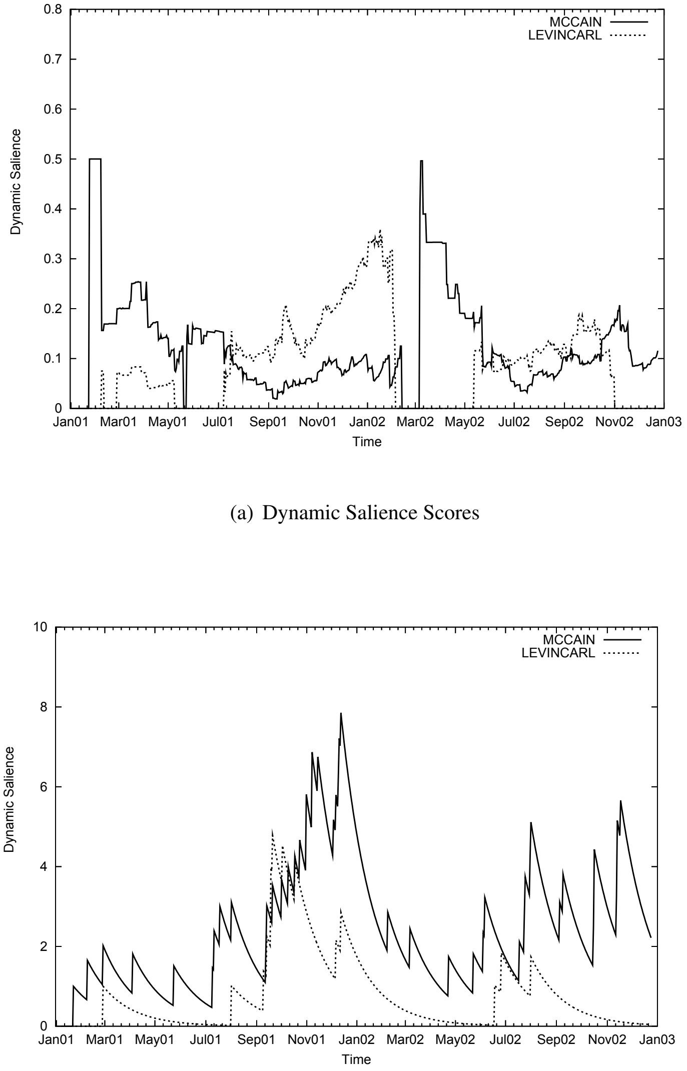
An even more interesting test would be to check whether the Democrats in general become more central than Republicans after the Jeffords switch. Figure 7.3(a) shows the normalized sum of the scores of all Democrats and all Republicans on topic 5 (Armed Forces 2) for the 107th Senate. The figure shows how the Republicans were most salient until soon after the Jeffords switch when the Democrats regained the majority and became more salient. We even discovered similar behavior when we studied how the average salience of Democrats and Republicans change across all topics. This is shown in Figure 7.4(a) where we can see that the Republicans were more salient on average for all topics until June 2001. Soon after the change in majority status, Democrats became more central. Figures 7.3(b) and 7.4(b) show the same results using the baseline system. We notice that the number of speeches made by the Democrats and the Republicans is very similar in most of the times. Even when one of the parties has more speeches than the other, it does not quite reflect the salience of the speakers or the parties in general.

An alternative approach to evaluate the dynamic scores is to exploit the cyclical nature of the legislative process as some bills are re-authorized on a fairly regular time schedule. For example, the farm bill comes due about every five years. As a new topic is coming up for debate, we expect the salience scores for relevant legislators to increase.

Figure 7.5 shows the dynamic scores of Senator Thomas Harkin (D - Iowa), and Senator Richard Lugar (R - Indiana) during the 107th senate on topic 24 (Agriculture). The two senators were identified, by the proposed method, as the most salient speakers for this topic, as expected, since they both served as chairmen of the Senate Committee on Agriculture, Nutrition, and Forestry when their party was in the majority during the 107th Senate. This committee was in charge of shepherding the Farm Bill through the Senate. The scores of both senators on the agriculture topic significantly increased starting late 2001 until June 2002. The debate began on the bill starting in September of 2001 and it was not passed until May 2002.

7.6 Conclusion

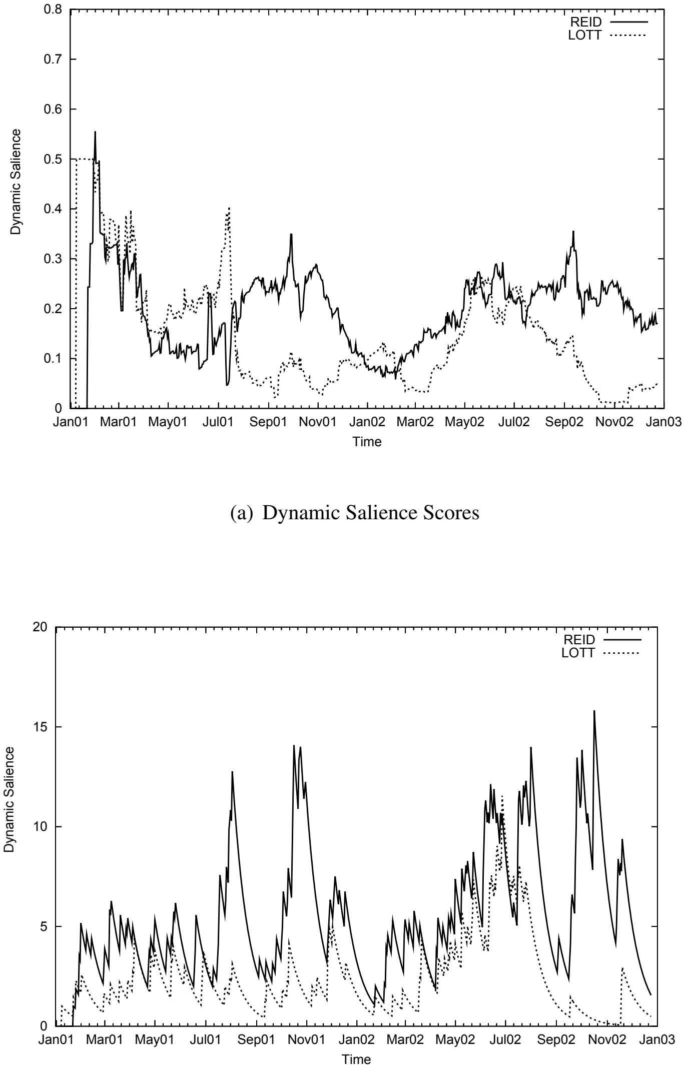
We presented a graph based method for analyzing the temporal evolution of the salience of participants in a discussion. We used this method to track the evolution of salience of speakers in the *US Congressional Record*. We showed that the way salience scores evolve over time can answer several interesting issues. We tracked how the influence of speakers varies with majority status and change of party control. We also showed how a baseline system that depends on the number of speeches fails to capture the interesting events captured by the proposed system. We studied the dynamics of the relative distribution of attention to each topic area in different time periods and captured the cyclical nature of the legislative process as some bills are re-authorized on a fairly regular time schedule.



(a) Dynamic Salience Scores

(b) Baseline

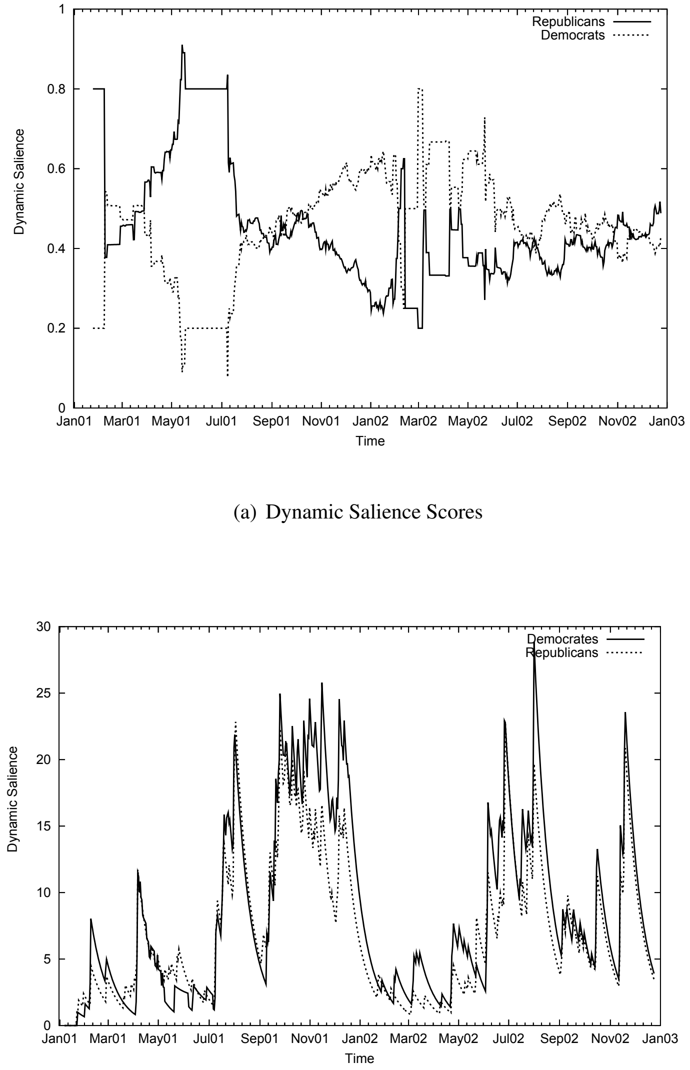
Figure 7.1: The switch of speaker salience near Jun 2001 for Topic 5(Armed Forces 2).



(a) Dynamic Salience Scores

(b) Baseline

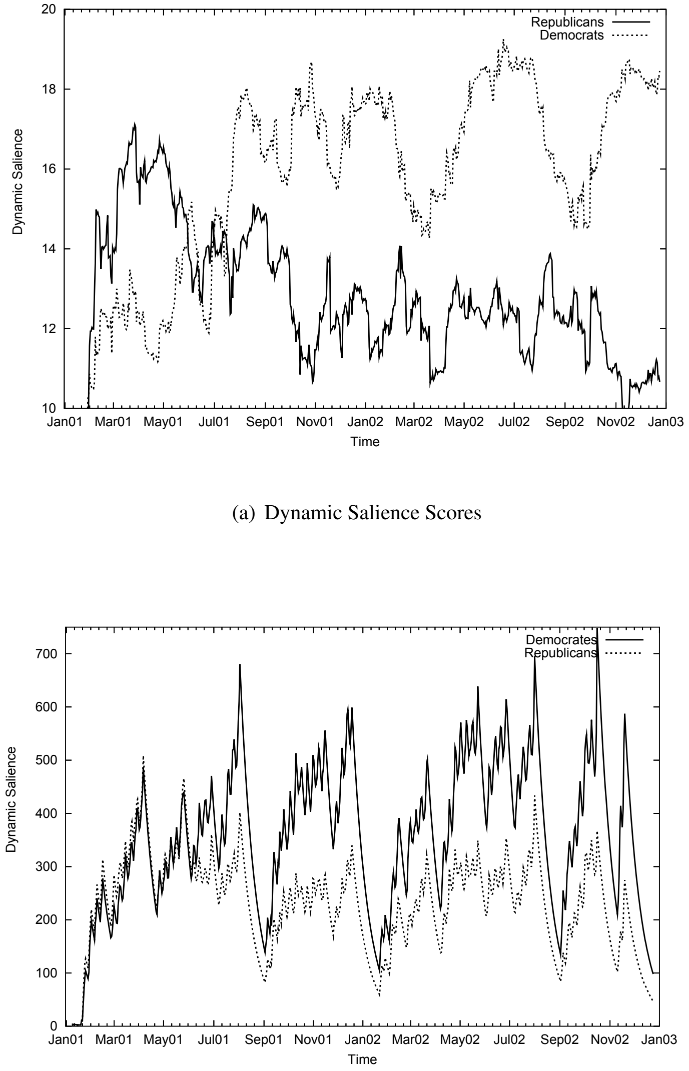
Figure 7.2: The switch of speaker salience near Jun 2001 for Topic 35(Procedural Legislation).



(a) Dynamic Salience Scores

(b) Baseline

Figure 7.3: The switch of speaker salience near Jun 2001 for Topic 5(Armed Forces 2), Republicans vs. Democrats.



(a) Dynamic Salience Scores

(b) Baseline

Figure 7.4: The switch of speaker salience near Jun 2001 for all topics, Republicans vs. Democrats.

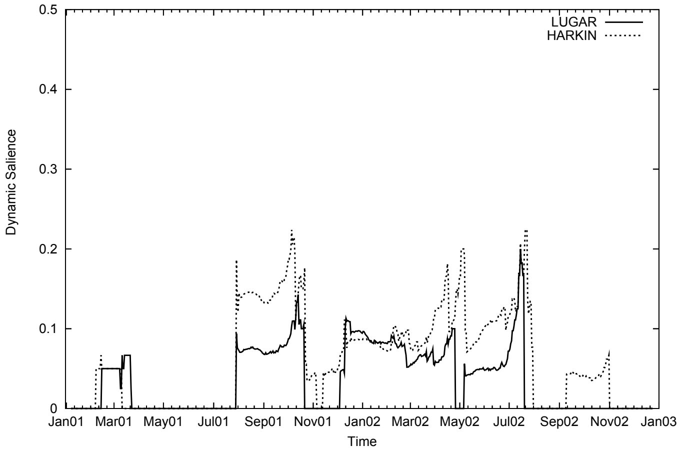


Figure 7.5: The effect of the Farm bill discussions on the relative distribution of attention to Topic 24 (Agriculture).

Part III. Conclusion and Future Directions

CHAPTER VIII

Conclusion

8.1 Summary of Contributions

One of the definitions of social media is: “a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, which allows the creation and exchange of user-generated content” [69]. Social media relies on the use of web-based technologies to turn communication into an interactive dialog. Social media can take many different forms, including:

- **Communication**: Blogs, Microblogging, Social networks, Discussion boards, and On-line advocacy.
- **Collaboration**: Wikis, Social tagging, and Social news.
- **Multimedia**: Photo sharing, Video sharing, Livecasting, Audio sharing, and Presentation sharing.
- **Reviews and opinions**: Product reviews, Business reviews, and Community question answering systems.
- **Entertainment**: Virtual worlds, and On-line game sharing.

Social media sites have been significantly growing in the past few years. For example, 75% of Internet surfers used “Social Media” by joining social networks, reading blogs, or contributing reviews to shopping sites [1]. Twitter processed more than one billion tweets in December 2009 and averages almost 40 million tweets per day[2]. In 2008, the blog search engine Technorati reported that they indexed 133 million blogs since 2002[3].

There is a huge amount of text exchanged between users in social media applications. Some forms of social media applications are mainly based on discussions like blogs, microblogs, discussion boards, and many others. Even those that are based on photo sharing, video sharing, product reviews, etc. have a lot of discussions involved. This huge amount of text carries an enormous amount of information documenting opinions, ideas, trends, relations, and so on.

The main question we tried to answer in this thesis is: "Can we use linguistic analysis techniques to understand the implicit relations that develop in on-line communities?" We had two main subgoals related to this question: (i) develop methods to automatically mine attitude from on-line discussions, and (ii) develop methods to automatically identify and track salience of participants in discussions. This chapter summarizes our main contributions and describes future directions for research.

This thesis consists of two main parts corresponding to the two goals we mentioned above. Chapters II, III and IV target our first goal by addressing the problems of identifying the semantic orientation of words, identifying sentences that show an attitude from the speaker to the addressee, and finally describing how a signed network representation can be automatically extracted from text to represent interactions between participants in a discussion. Chapters V, VI, and VII address our second goal where we proposed methods that assign salience to participants in discussions based on their text correspondences and to dynamically track how this salience evolves over time.

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

[2] http://digital.venturebeat.com/2010/02/10/54-of-us-internet-users-on-facebook-27-on-myspace/trackback/

[3] http://technorati.com/blogging/article/state-of-the-blogosphere-introduction/

Chapter II is the first chapter in the first part of this dissertation. In this chapter, we studied the problem of identifying semantic orientation of words. Predicting the semantic orientation of words is a very interesting task in Natural Language Processing and it has a wide variety of applications. The method we proposed is based on a random walk model that estimates the mean hitting time between any given word and two sets of positive and negative words to assign polarity to every word. The random walk is defined over a word relatedness graph that connects words that are semantically similar. The intuition behind this is that words with similar semantic meaning tend to have similar semantic orientation. We studied several ways for constructing this graph. We showed that resources like WordNet are very handy for coming up with such connections. The proposed method is based on the observation that a random walk starting at a given word is more likely to hit another word with the same semantic orientation before hitting a word with a different semantic orientation. Hence the mean hitting time from a given word to the positive (negative) set should be smaller if the given word is positive (negative). The proposed method is a semi-supervised method that uses a list of seeds along with the connections between words to classify new words as either positive or negative. In the mean time, we have shown that the proposed method can be applied in an unsupervised, or a minimally supervised, setting where only a handful of seed words are used to define the two semantic orientation classes rather than to train the model. We showed that the proposed method can predict semantic orientation with high accuracy. The proposed method was also shown to be fast, and simple to implement. We also extended the proposed method to the case of predicting the semantic orientation of foreign words. All previous work on this task has almost exclusively focused on English. We showed that applying off-the-shelf methods developed for English to other languages does not work well. The main reason behind that is the limited amount of resources available in foreign languages compared to English. We showed that the proposed method can predict the semantic orientation of foreign words with high accuracy and that it outperforms state of the art methods limited to using language specific resources. The proposed methods can be used for several other applications including: opinion mining, subjectivity analysis, product reviews mining, and many others.

The following chapter, Chapter III, studied the problem of identifying sentences that show an attitude from the text writer to the text recipient. This chapter takes a step further toward our overall goal of mining attitude from text exchanged in discussions. In the previous chapter, we showed how the semantic orientation of words can be reliably identified. The main goal of this chapter was to propose methods to identify sentences that show an attitude and predict whether that attitude is positive (friendly), or negative (antagonistic). Identifying sentences with attitude is harder that identifying opinionated sentences. While all sentences that carry an attitude are opinionated, not all opinionated sentences show an attitude toward other participants. We modeled the problem as a relation extraction problem where we sought to predict whether a relation exists between pronouns referring to the addressee and sentiment expressions. We showed that training supervised models using features of text, part-of-speech tags, and dependency relations allows us to identify sentences with attitudes from sentences without attitude. We compared the proposed method to several other baselines that use features based on the existence of polarized word, and proximity between polarized words and second person pronouns both in text and dependency trees. We showed that the proposed method is more accurate than all baselines. The methods proposed in this chapter are a key building block for the system that automatically extracts signed social networks from multi-party on-line discussions.

Chapter IV concludes the first part of this dissertation. In this chapter, we looked into the problem of mining attitude from discussions. We used on-line threaded discussions as our source of data. Threaded discussions include e-mails, e-mail lists, bulletin boards, newsgroups, and Internet forums. The goal of this chapter was to build a signed network representation of participant interaction where every edge has a sign that indicates whether the interaction is positive or negative. This is different from most of the research on social networks that has focused almost exclusively on positive links. We rely on natural language processing techniques to mine attitude between participants. We showed that linguistic analysis can be reliably used to extract signed social networks from text correspondences. We applied the proposed method to a large dataset of on-line discussions. We evaluated the performance of the proposed method using data annotated by human annotators. We asked annotators to examine text correspondences exchanged between individuals and judge whether their interaction is positive or negative. We also connected our analysis to social psychology theories and showed that our predictions agree to a large extent with the Structural Balance theory. We also discussed how signed networks, automatically extracted from text, can be used in several applications. We suggested using these methodologies for measuring the amount of dispute every topic receives, detecting subgroups that emerge in communities, mining public opinions toward different events, and summarizing different view points. We believe that this work provides an important step toward understanding the relation between language use and social interactions. We also believe that it will open the door to research efforts that go beyond standard social network analysis by studying the interplay of positive and negative connections.

In the second part of this dissertation, we studied the problem of identifying the most salient participants in a discussion by analyzing the text exchanged between them. This part started with Chapter V, where we presented the concept of content similarity based networks that link individuals based on the similarity of their textual representations. The idea behind content based networks is that we detect implicit relations between individuals by examining the text they write. This is very useful because several domains do not have any defined explicit link structures. We showed in this chapter that content based similarity can be used to find implicit links that can replace non-existing explicit links or augment networks with sparse explicit links. To gain more understanding of networks based on such implicit links, we compared them to explicit link networks built from the same datasets. We used two datasets from two different domains. The first is the AAN dataset which contains a large number of papers from the ACL Anthology. The second is the TREC BLOG06 dataset which contains a large number of blog feeds and associated permalinks. We used two datasets from two different domains to reduce the bias that may result from studying a single dataset. We selected two dataset with different characteristics, one weakly linked and the other more strongly linked. This allowed us to better compare the implicit and explicit based networks in different settings. We showed that there is a great consistency in the topological and temporal properties of the two types of networks over the two datasets. In the following chapters, we described methods that use content similarity based networks to identify static and dynamic salience of participants in a discussion.

Chapter VI uses the content based networks, we studied in Chapter V, to assign salience scores to participants in a discussion. The method presented in this chapter is a stochastic graph based method. It is based on defining a random walk model over a lexical similarity graph. The main intuition behind this method is that important participants tend to affect the ideas and rhetoric of other participants. Hence, the text written by important participants will tend to be lexically similar to the text written by other important participants. We showed that a method based on a random walk model that calculates the eigenvector centrality of text in a lexical graph can be used to assign salience scores to participants in discussions. We showed that there are other important aspects that the proposed method can take into consideration as well. For example, it is desirable in some applications to have a list of top salient entities, but in the mean time eliminate redundancy. We proposed a method that takes diversity into consideration when measuring importance by penalizing nodes very similar to already selected nodes. We also noticed that there are other attributes that affect the importance of any node that are more related to the node itself, rather than to its relation to other nodes in the network. One of our goals when designing this method was coming up with a general method that can accommodate several attributes while assigning salience to participants. To do that, we showed how the random walk model can be modified to allow for the incorporation of prior distributions reflecting other quality attributes. This allows us to bias the selection toward nodes with specific desirable properties such as text fluency. To show the potential of the proposed method, we applied it to the blogs domain. We used the proposed method to identify most salient blog feeds in particular topics. We showed that the proposed method achieves promising results and outperforms other random, heuristic, and greedy selection methods.

The methods we presented in Chapter VI are static. This means that they consider salience as a static property calculated over an aggregate of the relations between participants over the entire period of study. If the dataset spans a long period of time, this simplistic assumption does not hold any more. Rather, participant salience becomes a dynamic property that varies over time. In Chapter VII, we modified the method we proposed earlier in the previous chapter to cover the dynamic case, where the evolution of participant salience is tracked over time. To do that, we defined edge weight as a function of both static weight and time. This insured that weights change as time moves. This reduces the effect of edges as they get older. This also insured that some edges will disappear, when their weights drop below a certain threshold. Other edges will appear when new posts are considered. In the same time, new nodes will appear and other nodes will disappear if they become disconnected. This results in a dynamic graph that evolves over time. Applying methods from Chapter VI to this evolving graph will result in a salience score that is a function of time rather than single constant score. We used this method to track the evolution of salience of speakers in the *US Congressional Record*. We showed that the way salience scores evolve over time can answer several interesting issues. We studied how the majority status change correlates with the dynamic scores our method predicts. We showed that the predicted scores capture change in salience that results from change of majority status and party control. We also showed that a baseline system, that uses the discounted sum of the number of speeches, fails to capture the interesting events captured by the proposed method.

In this thesis, we studied how language choices correlate with social roles that develop in on-line communities. We found out that there is a large potential for applying Natural Language Processing techniques to text correspondences exchanged between individuals. This study allowed us to develop a deep understanding of the relation between the linguistic behavior of communicating groups of people and the social relations that develop between them. We were able to identify language constructs that people use to show positive attitude (i.e. agreement, praise), and negative attitude (i.e. disagreement, criticism, insult) toward other participants. We used this knowledge to build systems that can automatically extract a signed network representation of discussions. In those networks, every interaction has a sign that shows whether the interaction is friendly or antagonistic. We also studied how this reveals rifts and subgroup emergence in communities.

In addition, we studied how influence can be modeled by studying text. We showed that we can find implicit relations that model how authority is conferred between participants by analyzing text. We used this understanding of the relation between influence and text analysis to develop methods that model both static and dynamic salience in discussions. The studies and methods we presented in this thesis can find several interesting areas of applications, some of which are discussed earlier. For example, they can be used for identifying authoritative sources in social media, finding influential people in the community, detecting subgroups, mining attitude toward events and topics, measuring public opinion and dispute levels of different topics, summarizing different view points with respect to some topic or entity, and many other such applications.

8.2 Future Directions

Our long-term research goals are to develop more understanding of the relation between linguistic behavior and social relations in on-line communities. This will result in more accessibility to the huge amount of information embedded in social media. This deep linguistic analysis can actually reveal complex relations and behavioral patterns that current mining techniques cannot detect. We also aim to ultimately enhance social applications based on this understanding. In the remainder of this section, we will address more specific directions that we think this research can move in.

In this work, we mainly studied how text analysis may reveal explicit attitude and influence in discussions. There are several other language uses that exist in discussions. Extending this research to cover these uses may prove to be very useful. For example, sarcastic expressions are often used in discussions, sometimes with the intention of showing a negative attitude. Automatic detection of sarcasm in text is a very challenging problem and should be addressed as a separate problem. A method that utilizes holistic approaches that take context and previous interactions between discussion participants into consideration could be used to address it [19, 98]. Another interesting direction would be to mine the implicit attitude between participants based on their opinion with respect to common targets. For example, some person A may never explicitly show an attitude toward another person B in a community. However, identifying the opinion of both A, and B toward common targets may allow us to predict whether they mostly agree or mostly disagree. The work presented in Chapter III could be extended to cover this scenario.

Another possible direction of future work is to use iterative learning. In this approach, we will use linguistic features extracted from text to construct networks representing social groups. The resulting network structure could be used to learn new features that can be fed back to train a new system for social group identification. We have shown in Chapter IV how networks solely predicted using linguistic features agree with assertions from social psychology theories. We can use features inspired by such theories to build more accurate models of social relations in communities.

We mentioned earlier how research on signed networks has been limited due to the lack of on-line communities where users have an explicit means for labeling negative, as well as positive, relations. The methods we presented in this work allow us to automatically detect such relations. This may allow us to study those networks in more depth. For example, we can study networks temporal dynamics and how they evolve over time. We can also study whether unsigned networks generative models are applicable to signed networks. We may also propose new generative models for signed networks.

Finally, there are so many applications that could benefit from the methods proposed in this work. We would like to explore the applications front in more depth and study how the proposed methods could be applied to different domains. Such applications may include, but are not limited to: search and mining systems that specifically target social media, systems that measure public opinion toward events, topics, and products, intelligent systems that mine the activities and interactions of people in formal communities, and many others.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Lada Adamic and Natalie Glance. The political blogosphere and the 2004 U.S. election: Divided they blog. In *LinkKDD'05: Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005.

[2] Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman. Knowledge sharing and Yahoo answers: Everyone knows something. In *Proceedings of WWW'08*, pages 665–674. ACM, 2008.

[3] Eytan Adar and Lada A. Adamic. Tracking information epidemics in blogspace. In *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 207–214, 2005.

[4] Cem Akkaya, Alexander Conrad, Janyce Wiebe, and Rada Mihalcea. Amazon mechanical turk for subjectivity word sense disambiguation. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 195–203, 2010.

[5] R. Albert, A. Barabsi, and H. Jeong. Scale-free characteristics of random networks: The topology of the World Wide Web. *Physica A*, 281:69–77, 2000.

[6] R. Albert, H. Jeong, and A. Barabsi. The diameter of the world wide web. *Nature*, 401:130, 1999.

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

[8] Jaime Arguello, Jonathan L. Elsas, Jamie Callan, and Jaime G. Carbonell. Document representation and query expansion models for blog recommendation. In *International Conference on Weblogs and Social Media*, 2008.

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

[10] Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[11] W. Black, S. Elkateb, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease, and C. Fellbaum. Introducing the Arabic WordNet project. In *Third International WordNet Conference*, 2006.

[12] David Blei and John Lafferty. Dynamic topic models. In *Machine Learning: Proceedings of the Twenty-Third International Conference (ICML)*, 2006.

[13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[14] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. 30(1–7):107–117, 1998.

[15] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the Web: experiments and models. In *Proceedings of the Ninth International World-Wide Web Conference (WWW2000)*, 2000.

[16] Penelope Brown and Stephen C. Levinson. *Politeness — Some universals in language usage*. Cambridge University Press, Cambridge, 1987.

[17] Michael J. Brzozowski, Tad Hogg, and Gabor Szabo. Friends and foes: ideological social networking. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 817–820, New York, NY, USA, 2008.

[18] Razvan C. Bunescu and Raymond J. Mooney. A shortest path dependency kernel for relation extraction. In *HLT '05*, pages 724–731, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[19] Clint Burfoot and Timothy Baldwin. Automatic satire detection: are you having a laugh? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 161–164, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[20] Chris Callison-Burch. Fast, cheap, and creative: evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP '09, pages 286–295, 2009.

[21] Dorwin Cartwright and Frank Harary. Structure balance: A generalization of heiders theory. *Psych. Rev.*, 63, 1956.

[22] Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos. R-mat: A recursive model for graph mining. In *SDM*, 2004.

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

[24] Gao Cong, Long Wang, Chin-Yew Lin, Young-In Song, and Yueheng Sun. Finding question-answer pairs from online forums. In *SIGIR '08*, pages 467–474, 2008.

[25] Colin Cooper and Alan Frieze. A general model of Web graphs. *Random Struct. Algorithms*, 22:311–335, May 2003.

[26] J. A. Davis. Clustering and structural balance in graphs. *Human Relations*, 20:181–187, 1967.

[27] C. Den Hartog and N. Monroe. The value of majority status: The effect of Jeffords's switch on asset prices of republican and democratic firms. *Forthcoming Legislative Studies Quarterly*, 2004.

[28] Shilin Ding, Gao Cong, Chin-Yew Lin, and Xiaoyan Zhu. Using conditional random fields to extract contexts and answers of questions from online forums. In *ACL'08*, pages 710–718, 2008.

[29] D. Donato, L. Laura, S. Leonardi, , and S. Millozzi. Large scale properties of the webgraph. *European Physical Journal*, 38:239.243, 2004.

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

[31] Patrick Doreian and Andrej Mrvar. Partitioning signed social networks. *Social Networks*, 31(1):1 – 11, 2009.

[32] William G. Eggington. As Latin was, English is? *Discourse across Languages and Cultures*, 2002.

[33] Michael Elhadad, Jacques Robin, and Kathleen McKeown. Floating constraints in lexical choice. *Comput. Linguist.*, 23:195–239, June 1997.

[34] S. Elkateb, W. Black, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease, and C. Fellbaum. Building a WordNet for Arabic. In *Fifth International Conference on Language Resources and Evaluation*, 2006.

[35] S. Elkateb, W. Black, P. Vossen, D. Farwell, H. Rodrguez, A. Pease, and M. Alkhalifa. Arabic WordNet and the challenges of Arabic. In *Arabic NLP/MT Conference*, 2006.

[36] Jonathan Elsas, Jaime Arguello, Jamie Callan, and Jaime Carbonell. Retrieval and feedback models for blog distillation. In *The Sixteenth Text REtrieval Conference (TREC 2007)*, 2007.

[37] David Elson, Nicholas Dames, and Kathleen McKeown. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden, July 2010.

[38] P. Erdos and A. Renyi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.

[39] Güneş Erkan and Dragomir R. Radev. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*, 2004.

[40] Gunes Erkan. Language model-based document clustering using random walks. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 479–486, June 2006.

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

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

[43] Anthony Fader, Dragomir R. Radev, Michael H. Crespin, Burt L. Monroe, Kevin M. Quinn, and Michael Colaresi. Mavenrank: Identifying influential members of the us senate using lexical centrality. In *EMNLP 2007*, 2007.

[44] Lee Fedder. Syntactic choice in language generation. In *Proceedings of the ACL Workshop on Reversible Grammar in Natural Language Processing*, pages 45–52, 1991.

[45] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, June 2002.

[46] J. Goldenberg, B. Libai, and E. Muller. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, pages 211–223, August 2001.

[47] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 1978.

[48] Anatoliy Gruzd and Caroline Haythornthwaite. Automated discovery and analysis of social networks from threaded discussions. In *Proceedings of the International Network of Social Network Analysis (INSNA)*, St. Pete Beach, Florida, 2008.

[49] John J. Gumperz and Jenny Cook-Gumperz. Studying language, culture, and society: Sociolinguistics or linguistic anthropology? *Journal of Sociolinguistics*, 12(4):532–545, 2008.

[50] Ahmed Hassan, Anthony Fader, Michael Crespin, Kevin Quinn, Burt Monroe, Michael Colaresi, and Dragomir R. Radev. Tracking the dynamic evolution of participant salience in a discussion. In *International Conference on Computational Linguistics (COLING)*, 2008.

[51] Ahmed Hassan, Amjad Abu Jbara, and Dragomir R. Radev. Extracting signed networks from text. Submitted to EMNLP, 2011.

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

[53] 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.

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

[55] Ahmed Hassan and Dragomir R. Radev. Structural analysis of content based networks. Submitted to Information Processing and Mangement.

[56] Ahmed Hassan, Dragomir R. Radev, Junghoo Cho, and Amruta Joshi. Content based recommendation and summarization in the blogosphere. In *International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2009.

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

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

[59] Fritz Heider. Attitudes and cognitive organization. *Journal of Psychology*, 21:107–112, 1946.

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

[61] Jizhou Huang, Ming Zhou, and Dan Yang. Extracting chatbot knowledge from online discussion forums. In *IJCAI'07*, pages 423–428, 2007.

[62] A. Java, P. Kolari, T. Finin, A. Joshi, and T. Oates. Feeds that matter: A study of bloglines subscriptions. In *ICWSM*, 2007.

[63] S. Jha, D. Narayan, P. Pande, and P. Bhattacharyya. A WordNet for Hindi. In *International Workshop on Lexical Resources in Natural Language Processing*, 2001.

[64] Hongyan Jing and Kathleen McKeown. Combining multiple, large-scale resources in a reusable lexicon for natural language generation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, ACL '98, pages 607–613, 1998.

[65] Swapna Somasundaran Josef Ruppenhofer and Janyce Wiebe. Finding the sources and targets of subjective expressions. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, 2008.

[66] Mark T. Joseph and Dragomir R. Radev. Citation analysis, centrality, and the ACL anthology. Technical Report CSE-TR-535-07, University of Michigan. Department of Electrical Engineering and Computer Science, 2007.

[67] 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.

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

[69] 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.

[70] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD 2003*, 2003.

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

[72] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *ACL'03*, pages 423–430, 2003.

[73] J. Kleinberg. Authoritative sources in a hyperlinked environment. *9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[74] J. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The Web as a graph: Measurements, models and methods. In *Computing and Combinatorics, 5th Annual International Conference, COCOON'99*, volume 1627 of *Lecture Notes in Computer Science*, pages 1–17, July 1999.

[75] J. M. Kleinberg. Small-world phenomena and the dynamics of information. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[76] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the Web graph. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, 2000.

[77] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, pages 741–750, New York, NY, USA, 2009.

[78] Oren Kurland and Lillian Lee. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *Proceedings of SIGIR*, pages 306–313, 2005.

[79] Oren Kurland and Lillian Lee. Respect my authority! HITS without hyperlinks, utilizing cluster-based language models. In *Proceedings of SIGIR*, pages 83–90, 2006.

[80] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *The 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

[81] Lillian Lee. Measures of distributional similarity. In *37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, 1999.

[82] Adrienne Lehrer. Semantic fields and lezical structure. North Holland, Amsterdam and New York, 1974.

[83] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650, New York, NY, USA, 2010.

[84] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1361–1370, New York, NY, USA, 2010.

[85] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, Natalie Glance, and Nielsen BuzzMetrics. Cost-effective outbreak detection in networks. In *The 13th International Conference on Knowledge Discovery and Data Mining (KDD) 2007*, 2007.

[86] D. D. Lewis, Y. Yang, T. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[87] Chen Lin, Jiang-Ming Yang, Rui Cai, Xin-Jing Wang, and Wei Wang. Simultaneously modeling semantics and structure of threaded discussions: a sparse coding approach and its applications. In *SIGIR '09*, pages 131–138, 2009.

[88] Frank Lin and William Cohen. The multirank bootstrap algorithm: Semi-supervised political blog classification and ranking using semi-supervised link classification. In *International Conference on Weblogs and Social Media 2008*, 2008.

[89] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, December 2007.

[90] C. Macdonald and I. Ounis. The TREC Blogs06 collection: Creating and analysing a blog test collection. Technical Report TR-2006-224, University of Glasgow, 2006.

[91] C. Macdonald, I. Ounis, and I. Soboroff. Trec 2007 blog track. In *The Sixteenth Text REtrieval Conference (TREC 2007)*, 2007.

[92] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[93] C. Marlow. Audience, structure, and authority in weblog community. In *The 54th Annual Conference of the International Communication Association*, 2007.

[94] Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. Topic and role discovery in social networks with experiments on Enron and academic email. *J. Artif. Int. Res.*, 30:249–272, October 2007.

[95] Kathleen Mckeown, Lokesh Shrestha, and Owen Rambow. Using question-answer pairs in extractive summarization of email conversations. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '07, pages 542–550, Berlin, Heidelberg, 2007. Springer-Verlag.

[96] Geoffrey McLachlan and David Peel. *Finite Mixture Models*. New York: Wiley, 2000.

[97] F. Menczer. Correlated topologies in citation networks and the Web. *European Physical Journal B*, 38(2):211–221, 2004.

[98] Rada Mihalcea and Stephen Pulman. Characterizing humour: An exploration of features in humorous texts. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '07, pages 337–347, Berlin, Heidelberg, 2007. Springer-Verlag.

[99] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. In *EMNLP2004*, 2004.

[100] Rada Mihalcea, Paul Tarau, and Elizabeth Figa. Pagerank on semantic networks, with application to word sense disambiguation. pages 1126–1132, 2004.

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

[102] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, October 2002.

[103] Burt L. Monroe, Cheryl L. Monroe, Kevin M. Quinn, Dragomir Radev, Michael H. Crespin, Michael P. Colaresi, Anthony Fader, Jacob Balazer, and Steven P. Abney. United States Congressional speech corpus. Department of Political Science, The Pennsylvania State University, 2006.

[104] David A. Morand and Rosalie J. Ocker. Politeness theory and computer-mediated communication: A sociolinguistic approach to analyzing relational messages. *Hawaii International Conference on System Sciences*, 1:17b, 2003.

[105] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. In *KDD'02*, pages 341–349, 2002.

[106] Dipak Narayan, Debasri Chakrabarti, Prabhakar Pande, and P. Bhattacharyya. An experience in building the Indo WordNet - a WordNet for Hindi. In *First International Conference on Global WordNet*, 2002.

[107] 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.

[108] M. E. J. Newman. Scientific collaboration networks. I. Network construction and fundamental results. *Phys. Rev. E*, 64(1):016131, 2001.

[109] Mark E. J. Newman. A measure of betweenness centrality based on random walks. Technical Report cond-mat/0309045, Arxiv.org, 2003.

[110] J. Norris. Markov chains. Cambridge University Press, 1997.

[111] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. Technical Report 1999-66, Stanford Digital Library Technologies Project, Stanford University, November 11, 1999.

[112] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[113] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *HLT-EMNLP'05*, pages 339–346, 2005.

[114] Mason A. Porter, Peter J. Mucha, M. E. J. Newman, and Casey M. Warmbrand. A network analysis of committees in the U.S. house of representatives. *PNAS*, 102(20), May 2005.

[115] Foster Provost. Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI Workshop on Imbalanced Data Sets*, 2000.

[116] Kevin M. Quinn, Burt L. Monroe, Michael Colaresi, Michael H. Crespin, and Dragomir R. Radev. An automated method of topic-coding legislative speech over time with application to the 105th-108th U.S. senate. In *Midwest Political Science Association Meeting*, 2006.

[117] S. Redner. How popular is your paper? an empirical study of the citation distribution. *The European Physical Journal B*, 4:131, 1998.

[118] Paul Resnick and Richard Zeckhauser. Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics*, pages 127–157. Elsevier Science, 2002.

[119] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *EMNLP'03*, pages 105–112, 2003.

[120] Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. Thread detection in dynamic text message streams. In *SIGIR '06*, pages 35–42, 2006.

[121] Lokesh Shrestha and Kathleen McKeown. Detection of question-answer pairs in email conversations. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[122] Ka Cheung Sia, Junghoo Cho, Yun Chi, and Belle Tseng. Efficient computation of personal aggregate queries on blogs. In *The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2008.

[123] Swapna Somasundaran, Josef Ruppenhofer, and Janyce Wiebe. Detecting arguing and sentiment in meetings. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*, 2007.

[124] Xiaodan Song, Yun Chi, Koji Hino, and Belle Tseng. Identifying opinion leaders in the blogosphere. In *The sixteenth ACM conference on Conference on information and knowledge management*, 2007.

[125] K. Sparck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–20, 1972.

[126] Philip Stone, Dexter Dunphy, Marchall Smith, and Daniel Ogilvie. The general inquirer: A computer approach to content analysis. *The MIT Press*, 1966.

[127] Martin Szummer and Tommi Jaakkola. Partially labeled classification with Markov random walks. In *NIPS'02*, pages 945–952, 2002.

[128] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientations of words using spin model. In *ACL'05*, pages 133–140, 2005.

[129] Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *EMNLP 2006*, pages 327–335, 2006.

[130] Richard M. Tong. An operational system for detecting and tracking opinions in on-line discussion. Workshop note, SIGIR 2001 Workshop on Operational Text Classification, 2001.

[131] Peter Turney and Michael Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346, 2003.

[132] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL'02*, pages 417–424, 2002.

[133] Stijn van Dongen. Graph clustering by flow simulation. PhD thesis, University of Utrecht, 2000.

[134] P. Vossen. Eurowordnet: a multilingual database for information retrieval. In *DELOS workshop on Cross-language Information Retrieval*, 1997.

[135] Ronald Wardhaugh. *An Introduction to Sociolinguistics*, volume 1. Wiley-Blackwell, New York, 6th edition, 2009.

[136] D. J. Watts and Steven Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[137] Howard T. Welser, Eric Gleave, Danyel Fisher, and Marc Smith. Visualizing the Signatures of Social Roles in Online Discussion Groups. 2007.

[138] 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.

[139] 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.

[140] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 1(2):0, 2005.

[141] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP'05*, Vancouver, Canada, 2005.

[142] 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.

[143] 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.

[144] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML'03*, pages 912–919, 2003.